GenBI — Generative BI Platform

Complete codebase export

Session: GenBI session 2026-08-21 23:06
Git commit: 4703acd
Files: 204 · Lines: 40,032
Index: every filename below is a link; PDF outline mirrors the TOC.

Table of Contents

.claude/prompts/chart-gen-system.md 7
.claude/prompts/narrative-system.md 8
.claude/prompts/nl2sql-system.md 9
.claude/prompts/report-planner-system.md 10
.claude/rules/api-design.md 11
.claude/rules/llm-patterns.md 12
.claude/rules/sql-safety.md 13
.claude/skills/flint-chart/scripts/test-flint.sh 14
.claude/skills/flint-chart/SKILL.md 15
.dockerignore 16
.env.example 17
.gitattributes 18
.github/workflows/ci.yml 19
.gitignore 22
backend/alembic.ini 23
backend/app/__init__.py 24
backend/app/agents/__init__.py 25
backend/app/agents/base.py 26
backend/app/agents/chart/__init__.py 27
backend/app/agents/chart/flint_bridge.py 28
backend/app/agents/chart/flint_operator.py 32
backend/app/agents/chart/vis_flint_protocol.py 33
backend/app/agents/chart_gen_agent.py 35
backend/app/agents/narrative/__init__.py 38
backend/app/agents/narrative/narrative_agent.py 39
backend/app/agents/nl2sql/__init__.py 42
backend/app/agents/nl2sql/nl2sql_agent.py 43
backend/app/agents/registry.py 46
backend/app/agents/router_agent.py 47
backend/app/agents/validation/__init__.py 49
backend/app/agents/validation/chart_validator.py 50
backend/app/agents/validation/validation_agent.py 58
backend/app/api/__init__.py 62
backend/app/api/v1/__init__.py 63
backend/app/api/v1/auth.py 64
backend/app/api/v1/charts.py 66
backend/app/api/v1/chat.py 67
backend/app/api/v1/conversations.py 68
backend/app/api/v1/datasources.py 69
backend/app/api/v1/health.py 70
backend/app/api/v1/metrics.py 71
backend/app/api/v1/reports.py 74

backend/app/api/v1/router.py 76
backend/app/connectors/__init__.py 77
backend/app/connectors/base.py 78
backend/app/connectors/postgresql_connector.py 79
backend/app/connectors/registry.py 83
backend/app/core/__init__.py 84
backend/app/core/auth.py 85
backend/app/core/cache.py 86
backend/app/core/config.py 93
backend/app/core/embeddings.py 95
backend/app/core/llm_client.py 96
backend/app/core/logging.py 100
backend/app/core/masking.py 101
backend/app/core/observability.py 104
backend/app/core/security.py 108
backend/app/db/__init__.py 109
backend/app/db/graph_schema.py 110
backend/app/db/migrations/env.py 115
backend/app/db/migrations/script.py.mako 116
backend/app/db/migrations/versions/.gitkeep 117
backend/app/db/migrations/versions/0001_baseline_schema.py 118
backend/app/db/migrations/versions/0002_app_roles_rls.py 121
backend/app/db/migrations/versions/0003_analytics_tables.py 122
backend/app/db/migrations/versions/0004_messages.py 125
backend/app/db/migrations/versions/0005_reports.py 126
backend/app/db/models.py 127
backend/app/db/session.py 130
backend/app/main.py 131
backend/app/models/__init__.py 132
backend/app/models/chart.py 133
backend/app/models/chat.py 134
backend/app/semantic/__init__.py 135
backend/app/semantic/cube_client.py 136
backend/app/services/__init__.py 144
backend/app/services/audit.py 145
backend/app/services/chat_service.py 147
backend/app/services/conversations.py 156
backend/app/services/reports.py 158
backend/app/services/schema_retrieval.py 162
backend/Dockerfile 164
backend/pyproject.toml 165
backend/README.md 166
backend/tests/__init__.py 167
backend/tests/agents/__init__.py 168

backend/tests/agents/test_validation_agent.py 169
backend/tests/agents/test_validation_explain.py 171
backend/tests/api/test_auth.py 172
backend/tests/api/test_conversations_api.py 174
backend/tests/api/test_feedback_and_roles.py 176
backend/tests/api/test_metrics.py 178
backend/tests/api/test_reports_api.py 181
backend/tests/api/test_tenant_isolation.py 183
backend/tests/conftest.py 186
backend/tests/evals/__init__.py 188
backend/tests/evals/nl2sql_golden.json 189
backend/tests/evals/test_nl2sql_eval.py 192
backend/tests/fixtures/llm_mock.py 200
backend/tests/semantic/__init__.py 203
backend/tests/semantic/test_catalog.py 204
backend/tests/semantic/test_cube_client.py 206
backend/tests/semantic/test_cube_client_phase9.py 210
backend/tests/semantic/test_cube_query.py 212
backend/tests/services/test_audit.py 214
backend/tests/services/test_chat_pipeline.py 216
backend/tests/services/test_chat_service_context.py 219
backend/tests/services/test_conversation_wiring.py 221
backend/tests/services/test_conversations.py 224
backend/tests/services/test_login_rate_limit.py 226
backend/tests/services/test_reports.py 228
backend/tests/services/test_schema_retrieval.py 232
backend/tests/test_embeddings.py 234
backend/tests/test_security.py 235
backend/uv.lock 236
CLAUDE.md 315
docs/adr/001-dual-chart-stack.md 319
docs/adr/002-agent-architecture.md 320
docs/adr/003-vis-flint-protocol.md 321
docs/adr/004-flint-bridge-fallback.md 322
docs/adr/005-age-and-pgvector-image.md 323
docs/adr/006-enforced-rls-roles.md 324
docs/adr/007-cube-native-semantic-layer.md 325
docs/adr/008-per-tenant-cube-data-path.md 326
docs/agent-system.md 327
docs/api-reference.md 330
docs/api/openapi.yaml 335
docs/architecture-overview.md 337
docs/core-services.md 339
docs/data-layer.md 342

docs/frontend-guide.md 345

docs/index.md 349

docs/infrastructure.md 350

docs/semantic-layer.md 353

frontend/.dockerignore 355

frontend/Dockerfile 356

frontend/eslint.config.mjs 357

frontend/next-env.d.ts 358

frontend/next.config.js 359

frontend/package.json 360

frontend/pnpm-lock.yaml 361

frontend/src/app/chat/page.tsx 416

frontend/src/app/explore/page.tsx 417

frontend/src/app/globals.css 418

frontend/src/app/layout.tsx 419

frontend/src/app/login/page.tsx 420

frontend/src/app/page.tsx 421

frontend/src/app/reports/page.tsx 422

frontend/src/components/auth/auth-provider.tsx 423

frontend/src/components/auth/login-form.tsx 425

frontend/src/components/charts/chart-card.tsx 426

frontend/src/components/chat/chat-view.tsx 428

frontend/src/components/explore/explore-view.tsx 434

frontend/src/components/reports/reports-view.tsx 437

frontend/src/components/ui/avatar.tsx 440

frontend/src/components/ui/badge.tsx 441

frontend/src/components/ui/button.tsx 442

frontend/src/components/ui/card.tsx 443

frontend/src/components/ui/dialog.tsx 444

frontend/src/components/ui/dropdown-menu.tsx 446

frontend/src/components/ui/input.tsx 447

frontend/src/components/ui/separator.tsx 448

frontend/src/components/ui/skeleton.tsx 449

frontend/src/components/ui/tabs.tsx 450

frontend/src/components/ui/tooltip.tsx 451

frontend/src/lib/api-client.ts 452

frontend/src/lib/auth-storage.ts 455

frontend/src/lib/shadcn.ts 456

frontend/src/lib/validators.ts 457

frontend/src/types/chart.ts 459

frontend/src/types/index.ts 460

frontend/tailwind.config.ts 461

frontend/tsconfig.json 462

infra/docker-compose.dev.yml 463

infra/docker-compose.yml 465

infra/postgres/Dockerfile 466

infra/postgres/init.sql 467

infra/prometheus/prometheus.yml 471

LICENSE 472

Makefile 473

pyproject.toml 474

README.md 475

scripts/db_admin.py 479

scripts/embed_schema.py 480

scripts/gen-env.sh 484

scripts/seed_test_data.py 486

scripts/setup.sh 491

scripts/verify.sh 493

semantic/cube/.env.example 495

semantic/cube/cube.js 496

semantic/cube/model/activity.yml 497

semantic/cube/model/customers.yml 498

semantic/cube/model/deals.yml 499

semantic/cube/model/orders.yml 500

semantic/cube/model/products.yml 501

semantic/cube/model/regions.yml 502

semantic/cube/model/sales.yml 503

semantic/cube/model/sales_reps.yml 504

semantic/cube/model/transactions.yml 505

semantic/cube/model/web_users.yml 506

semantic/README.md 507

todo.md 508

uv.lock 521

VERIFICATION.md 601

.claude/prompts/chart-gen-system.md

41 lines · 1.2 KB

```
# Chart Generation Agent — System Prompt

You are a chart specification generator for the Flint Chart rendering system. Your task is to convert a dataset and user intent into a valid ChartAssemblyInput specification.

## Flint ChartAssemblyInput Schema

```json
{
  "chartType": "string (e.g. 'Bar Chart', 'Line Chart')",
  "encodings": {
    "x": {"field": "column_name"},
    "y": {"field": "column_name"},
    "color": {"field": "column_name"}
  },
  "baseSize": {"width": 600, "height": 400},
  "semantic_types": {
    "column_name": "Category|Quantity|Temporal"
  },
  "data": {
    "values": [...]
  }
}
```

Chart Type Selection Rules

1. If the user explicitly asks for a chart type, use it
2. If there is a temporal field on x, prefer Line Chart or Area Chart
3. If there is one category and one quantity with <= 8 rows, consider Pie/Donut Chart
4. Default to Bar Chart for categorical comparisons
5. Use Scatter Chart for two quantitative fields

Field Inference

- Columns with text/names -> Category (map to x or color)
- Columns with numbers -> Quantity (map to y or size)
- Columns with dates -> Temporal (map to x)

Output

Return ONLY the JSON spec. No explanatory text.
```

.claude/prompts/narrative-system.md

28 lines · 1.3 KB

```
# Narrative Generation Agent — System Prompt

You are a data storyteller for a generative BI platform. Your task is to write a concise, insightful narrative paragraph explaining the key findings in a dataset and its accompanying chart.

## Style

- Professional but conversational — like a skilled data analyst talking to a business stakeholder
- Lead with the most important finding
- Use specific numbers, not vague terms like "many" or "large"
- 3-5 sentences maximum
- No bullet points — flowing prose
- Include: what happened, why it matters, and one suggestion for action

## Context

You receive:
1. The user's original question
2. A summary of the data (head of result set)
3. The chart type and key encodings
4. Any notable statistics (totals, averages, outliers, trends)

## Example

**User question**: "How did our Q1 revenue compare to Q1 last year?"
**Data summary**: Revenue grew from $2.1M to $2.8M, with Enterprise segment growing 45% while SMB was flat.

**Output**:
Revenue reached $2.8 million in Q1, a 33% increase over the same period last year. The growth was heavily concentrated in the Enterprise segment, which expanded 45% to $1.9 million, while SMB revenue remained essentially flat at $0.9 million. This suggests your upmarket strategy is working, but the SMB stall warrants attention.
```

.claude/prompts/nl2sql-system.md

33 lines · 1.1 KB

```
# NL2SQL Agent — System Prompt

You are an expert SQL query writer for a generative BI platform. Your task is to convert a user's natural language question into a valid, safe, read-o
nly SQL query.

## Context Available

You have access to:
- **Schema context**: The most relevant tables and columns for this query, retrieved via semantic search
- **Metric definitions**: Business metrics from the Cube.dev semantic layer
- **Few-shot examples**: Similar queries that have been validated in the past

## Rules

### Always
1. Generate ONLY SELECT queries — never DROP, DELETE, TRUNCATE, UPDATE, INSERT, ALTER, CREATE
2. Use fully qualified table names: `schema_name.table_name`
3. Include tenant filter in every WHERE clause
4. Use parameterized values — never embed user input directly
5. Limit results: `LIMIT 1000` unless the user asks for more
6. Use CTEs (WITH clauses) for complex queries

### Output Format

Return a JSON object with:
```json
{
  "sql": "SELECT ...",
  "explanation": "This query...",
  "tables_used": ["schema.table1"],
  "assumptions": ["Assuming X means Y..."],
  "warnings": []
}
```
```

.claude/prompts/report-planner-system.md

36 lines · 1.6 KB

```
# Report Planner

You plan multi-section analytical reports from a natural language request. You do NOT write SQL or prose — you only choose metrics from the provided c
atalog and give each section a short title.

## Input

You receive:
- **Report request**: the user's natural language description of the report they want
- **Available Metrics**: the semantic layer catalog (metric names like `Sales.revenue_total`, their types, descriptions, and dimensions)

## Output

Respond with ONLY a JSON object (no markdown fences, no commentary):

```json
{
  "title": "Short report title (max 80 chars)",
  "sections": [
    {
      "metric": "Cube.MeasureName from the catalog — exact match required",
      "title": "Section heading (max 100 chars)",
      "dimension": "A dimension of that metric to slice by (omit for a single total)",
      "granularity": "month|day — only when the dimension is a time dimension"
    }
  ]
}
```

Rules

1. Pick between 2 and 4 sections that together cover the request. Fewer, well-chosen metrics beat more, redundant ones.
2. `metric` MUST be an exact name from the catalog. Never invent or pluralize metric names.
3. Prefer slicing by a categorical dimension (region, category, stage, country...) unless the request implies a trend — then use a time dimension with
 "month".
4. Section titles are human-readable ("Revenue by Region", not "sales_revenue_total").
5. If the request mentions specific metrics, use them first; fill remaining sections with complementary metrics from the catalog.
6. If the catalog has fewer metrics than sections requested, use what exists — never repeat a metric across sections.
```

.claude/rules/api-design.md

34 lines · 1.0 KB

```
# API Design Standards

## Route conventions

- ALL API routes are versioned under `/api/v1/`
- Route files are `snake_case.py` in `backend/app/api/v1/`

## Request/Response schema

- Every request body is a Pydantic v2 model with `extra="forbid"`
- Every response body is a Pydantic v2 model
- Error responses use: `{"detail": {"code": "ERROR_CODE", "message": "Human-readable message"}}`

## Async everywhere

- All FastAPI route handlers are `async def`
- All DB calls use async SQLAlchemy (`async_session` with `asyncpg`)

## Dependency injection

- Auth/session/db always via FastAPI `Depends()` — never instantiated in route body
- `user: dict = Depends(get_current_user)` for auth
- `db: AsyncSession = Depends(get_db)` for database

## Streaming

- SSE endpoints return `StreamingResponse` with media_type `text/event-stream`
- Each event is: `data: {json}\n\n`

## CORS

- Configured via `settings.CORS_ORIGINS`
- In development: `["http://localhost:3000"]`
- In production: strict origin list
```

.claude/rules/llm-patterns.md

35 lines · 1.2 KB

```
# LLM Prompt Engineering Conventions

## Model selection

- Use `settings.LLM_REASONING_MODEL` (claude-opus-4) for SQL generation, complex reasoning
- Use `settings.LLM_FAST_MODEL` (claude-haiku-4) for intent classification, chart specs, narratives
- NEVER hardcode model names in business logic

## Prompt template management

- Prompt templates are versioned files in `.claude/prompts/`
- Load templates with `load_prompt("prompt_name")` — never inline long prompts
- Templates use `{variable}` substitution with Pydantic validation

## Schema context compression

- Embed only relevant table/column metadata into LLM context
- Use pgvector cosine similarity to retrieve top-k relevant tables
- Never send the full database schema to the LLM
- Include column descriptions from dbt schema.yml in the context

## Chain of thought

- Reasoning models (opus-4) always use `thinking` blocks for SQL generation
- Fast models (haiku-4) use direct generation

## Error handling

- LLM calls always go through `llm_client.py` which handles timeout + retry, token budgets, and latency tracking

## Few-shot examples

- Stored in `backend/app/agents/examples/` as JSON files
- Load dynamically based on query similarity via pgvector
- Never hardcode examples in source code
```

.claude/rules/sql-safety.md

31 lines · 1.2 KB

```
# SQL Safety Rules

## NEVER generate destructive SQL

All database access through the query engine is READ-ONLY. The following operations are forbidden:

- `DROP TABLE`, `DROP DATABASE`, `DROP SCHEMA`, `DROP INDEX`
- `DELETE FROM`, `TRUNCATE TABLE`
- `ALTER TABLE DROP COLUMN`, `ALTER TABLE RENAME`
- `UPDATE` (unless through explicit service methods with authorization)
- `INSERT` (unless through explicit service methods with authorization)
- `CREATE USER`, `GRANT`, `REVOKE`
- `COPY` (export/import)
- Any multi-statement queries (separated by `;`)

## Always use parameterized queries

- NEVER concatenate user input into SQL strings
- Use SQLAlchemy parameterized queries: `select().where(table.c.name == bindparam("name"))`
- Raw SQL only through `text()` with explicit bind parameters

## Timeout enforcement

- All analytical queries are capped at 30s via `SET statement_timeout = '30s'`
- The ValidationAgent injects this before every query execution

## ValidationAgent must run before every SQL execution

- The ValidationAgent checks for destructive patterns
- It runs a dry-run EXPLAIN plan for cost estimation
- Queries that would scan > 1M rows require explicit user confirmation
```

.claude/skills/flint-chart/scripts/test-flint.sh

15 lines · 0.4 KB

```
#!/usr/bin/env bash
# Flint Chart integration test script
# Verifies that the Flint MCP server is reachable and functional.
set -euo pipefail

echo "=== Flint Chart Integration Test ==="

if npx -y flint-chart-mcp --version > /dev/null 2>&1; then
    echo "PASS: flint-chart-mcp is available via npx"
else
    echo "FAIL: flint-chart-mcp not found — check npm registry access"
    exit 1
fi

echo "=== All checks passed ==="
```

.claude/skills/flint-chart/SKILL.md

84 lines · 2.6 KB

```
# Flint Chart — Agent Skill

## When to use this skill

Use Flint Chart when the user requests any of the following:
- A chart or visualization ("make a chart", "visualize this", "plot the data")
- A graph, diagram, or data graphic
- A report that includes charts
- PNG or SVG export of a visualization
- A static chart for embedding in a document, email, or dashboard tile

Do NOT use Flint Chart for interactive dashboard widgets where hover/zoom/brush is needed — those use the GPT-Vis protocol with AntV G2.

## Flint ChartAssemblyInput Schema

Flint uses a single unified schema for approximately 40 chart types across three backends (Vega-Lite, ECharts, Chart.js).

### Required fields

```json
{
  "chartType": "Bar Chart",
  "data": {
    "values": [
      {"x_field": "value", "y_field": 123}
    ]
  }
}
```

Full schema

```json
{
  "chartType": "Bar Chart",
  "encodings": {
    "x": {"field": "column_name"},
    "y": {"field": "column_name"},
    "color": {"field": "column_name"},
    "size": {"field": "column_name"}
  },
  "baseSize": {"width": 600, "height": 400},
  "semantic_types": {
    "column_name": "Category",
    "other_column": "Quantity",
    "date_column": "Temporal"
  },
  "data": {
    "values": [...]
  }
}
```

Supported Chart Types

Categorical: Bar Chart, Stacked Bar Chart, Grouped Bar Chart, Pie Chart, Donut Chart
Trend/Temporal: Line Chart, Multi-series Line Chart, Area Chart, Stacked Area Chart
Distribution: Histogram, Box Plot, Scatter Chart
Relational: Heatmap, Bubble Chart

Semantic Types

- `Category` — discrete labels (names, regions, IDs)
- `Quantity` — continuous numeric values (revenue, counts, percentages)
- `Temporal` — dates, timestamps, years

Backend Selection

- **Vega-Lite** (recommended): Best for SVG output, typography, and complex statistical charts
- **ECharts**: More chart types (Sankey, radar, treemap), better for Asian character rendering
- **Chart.js**: Fastest rendering, PNG only, good for simple dashboards

Data Format

Prefer `data.values` for small datasets (< 100 rows). For larger datasets, write to a temp file in `FLINT_MCP_DATA_ROOTS` and reference via `data.url`
.

Integration with GenBI Pipeline

1. The NL2SQL agent generates a SQL query from the user's natural language
2. The SQL is executed against the data source, producing a result set
3. The ChartGenAgent analyzes the result set, infers semantic types, and selects a chart type
4. The ChartGenAgent produces a `ChartAssemblyInput` spec
5. The FlintChartBridge renders the spec via Flint MCP
6. The result (PNG/SVG) is embedded in the chat response
```

.dockerignore

17 lines · 0.3 KB

```
# Build context is the REPO ROOT (backend Dockerfile copies backend/ and
# .claude/prompts) — exclude everything else, especially frontend's
# node_modules (600MB+, and Windows file modes break the tar checksum).
**/node_modules
**/.next
**/out
**/.git
**/.venv
**/__pycache__
**/.mimosa
**/.zcode
frontend
infra
docs
scripts
semantic
*.md
```

.env.example

56 lines · 1.8 KB

```
# ==============================================================================
# GenBI Environment Configuration
# Copy this file to .env and fill in values
# NEVER commit .env files — this .env.example is the canonical template
# ==============================================================================

# --- LLM ---
ANTHROPIC_API_KEY=sk-ant-...
LLM_REASONING_MODEL=claude-opus-4
LLM_FAST_MODEL=claude-haiku-4

# --- Embeddings (schema retrieval; dims must match VECTOR(1536) columns) ---
OPENAI_API_KEY=sk-...
EMBEDDING_MODEL=text-embedding-3-small
EMBEDDING_DIMS=1536

# --- Database (asyncpg format) ---
# Runtime role: non-superuser, RLS-bound (see docs/adr/006-enforced-rls-roles.md).
# Owner credentials (Alembic, admin scripts) stay in DATABASE_URL_SYNC.
DATABASE_URL=postgresql+asyncpg://genbi_app:genbi_app@localhost:5432/genbi
# Login-only role (reads `users` across tenants). Leave blank to derive from
# DATABASE_URL with genbi_auth credentials.
DATABASE_URL_AUTH=
# Owner/superuser — migrations and admin scripts ONLY, never the app runtime.
DATABASE_URL_SYNC=postgresql://genbi:genbi@localhost:5432/genbi

# --- Redis ---
REDIS_URL=redis://localhost:6379/0

# --- Semantic Layer ---
CUBE_API_URL=http://localhost:4000/cubejs-api/v1
CUBE_API_SECRET=change-me-in-production

# --- Observability ---
LANGFUSE_SECRET_KEY=sk-lf-...
LANGFUSE_PUBLIC_KEY=pk-lf-...
LANGFUSE_HOST=https://cloud.langfuse.com
OTEL_EXPORTER_OTLP_ENDPOINT=http://localhost:4318

# --- Auth ---
JWT_SECRET_KEY=change-me-in-production-use-openssl-rand-64
JWT_ALGORITHM=HS256
JWT_EXPIRE_MINUTES=60

# --- Tenant ---
TENANT_ENCRYPTION_KEY=change-me-in-production-use-fernet-generate-key

# --- Flint Chart (MCP) ---
FLINT_MCP_BACKENDS=vegalite,echarts,chartjs
FLINT_MCP_DATA_ROOTS=/tmp/genbi-charts

# --- App ---
APP_ENV=development
DEBUG=true
LOG_LEVEL=DEBUG
CORS_ORIGINS=http://localhost:3000
```

.gitattributes

2 lines · 0.1 KB

```
# Auto detect text files and perform LF normalization
* text=auto
```

.github/workflows/ci.yml

253 lines · 7.7 KB

```yaml
name: CI/CD Pipeline

on:
  push:
    branches: [main, develop]
  pull_request:
    branches: [main]

env:
  PYTHON_VERSION: "3.12"
  NODE_VERSION: "20"

jobs:
  # ------------------------------------------------------------------
  # Backend CI
  # ------------------------------------------------------------------
  backend-lint-test:
    name: Backend — Lint & Test
    runs-on: ubuntu-latest
    defaults:
      run:
        working-directory: backend

    services:
      postgres:
        image: pgvector/pgvector:pg16
        env:
          POSTGRES_USER: genbi
          POSTGRES_PASSWORD: genbi
          POSTGRES_DB: genbi_test
        ports:
          - 5432:5432
        options: >-
          --health-cmd pg_isready
          --health-interval 10s
          --health-timeout 5s
          --health-retries 5

      redis:
        image: redis:7-alpine
        ports:
          - 6379:6379
        options: >-
          --health-cmd "redis-cli ping"
          --health-interval 10s
          --health-timeout 5s
          --health-retries 5

    steps:
      - uses: actions/checkout@v4

      - name: Install uv
        uses: astral-sh/setup-uv@v2
        with:
          python-version: ${{ env.PYTHON_VERSION }}

      - name: Install dependencies
        run: uv sync --frozen

      - name: Lint with ruff
        run: |
          uv run ruff check .
          uv run ruff format --check .

      - name: Run migrations on test DB
        env:
          DATABASE_URL: postgresql+asyncpg://genbi:genbi@localhost:5432/genbi_test
          DATABASE_URL_SYNC: postgresql://genbi:genbi@localhost:5432/genbi_test
        run: uv run alembic upgrade head

      - name: Run unit & integration tests
        # Test step connects as the RLS-bound roles (created by the migration
        # step above) so the auth + isolation suites exercise real enforcement,
        # exactly as the app runs in dev/prod. DATABASE_URL_SYNC stays on the
        # owner role for test fixtures that need admin cleanup.
        env:
          DATABASE_URL: postgresql+asyncpg://genbi_app:genbi_app@localhost:5432/genbi_test
          DATABASE_URL_AUTH: postgresql+asyncpg://genbi_auth:genbi_auth@localhost:5432/genbi_test
          DATABASE_URL_SYNC: postgresql://genbi:genbi@localhost:5432/genbi_test
          REDIS_URL: redis://localhost:6379/0
          ANTHROPIC_API_KEY: ${{ secrets.ANTHROPIC_API_KEY }}
          CUBE_API_URL: http://localhost:4000/cubejs-api/v1
          CUBE_API_SECRET: ${{ secrets.CUBE_API_SECRET }}
          JWT_SECRET_KEY: ci-test-key-not-for-production
          APP_ENV: test
        run: uv run pytest tests/ -v -m "not e2e" --tb=short --cov=app --cov-report=xml

      - name: Run golden NL2SQL eval
        env:
          DATABASE_URL: postgresql+asyncpg://genbi:genbi@localhost:5432/genbi_test
          ANTHROPIC_API_KEY: ${{ secrets.ANTHROPIC_API_KEY }}
          APP_ENV: test
        run: uv run pytest tests/evals/ -v --tb=short

      - name: Upload coverage to Codecov
        uses: codecov/codecov-action@v4
        with:
          files: backend/coverage.xml
          flags: backend

```

```yaml
  # ----------------------------------------------------------------
  # Frontend CI
  # ----------------------------------------------------------------
  frontend-lint-build:
    name: Frontend — Lint & Build
    runs-on: ubuntu-latest
    defaults:
      run:
        working-directory: frontend

    steps:
      - uses: actions/checkout@v4

      - name: Setup pnpm
        uses: pnpm/action-setup@v3
        with:
          version: 9

      - name: Setup Node
        uses: actions/setup-node@v4
        with:
          node-version: ${{ env.NODE_VERSION }}
          cache: pnpm
          cache-dependency-path: frontend/pnpm-lock.yaml

      - name: Install dependencies
        run: pnpm install --frozen-lockfile

      - name: TypeScript type check
        run: pnpm typecheck

      - name: Lint
        run: pnpm lint

      - name: Build
        run: pnpm build

  # ----------------------------------------------------------------
  # Custom Postgres image (pgvector + Apache AGE)
  # ----------------------------------------------------------------
  postgres-image:
    name: Postgres — Build AGE Image
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v4

      - name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v3

      - name: Login to GitHub Container Registry
        uses: docker/login-action@v3
        with:
          registry: ghcr.io
          username: ${{ github.actor }}
          password: ${{ secrets.GITHUB_TOKEN }}
        if: github.event_name == 'push' && (github.ref == 'refs/heads/main' || github.ref == 'refs/heads/develop')

      - name: Build & push pgvector+AGE image
        uses: docker/build-push-action@v5
        with:
          context: infra/postgres
          push: ${{ github.event_name == 'push' && (github.ref == 'refs/heads/main' || github.ref == 'refs/heads/develop') }}
          tags: |
            ghcr.io/${{ github.repository }}/genbi-postgres:latest
          cache-from: type=gha
          cache-to: type=gha,mode=max

  # ----------------------------------------------------------------
  # Docker build
  # ----------------------------------------------------------------
  docker-build:
    name: Docker — Build Images
    runs-on: ubuntu-latest
    needs: [backend-lint-test, frontend-lint-build]
    if: github.event_name == 'push' && (github.ref == 'refs/heads/main' || github.ref == 'refs/heads/develop')

    steps:
      - uses: actions/checkout@v4

      - name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v3

      - name: Login to GitHub Container Registry
        uses: docker/login-action@v3
        with:
          registry: ghcr.io
          username: ${{ github.actor }}
          password: ${{ secrets.GITHUB_TOKEN }}

      - name: Build & push backend image
        uses: docker/build-push-action@v5
        with:
          context: backend
          push: true
          tags: |
            ghcr.io/${{ github.repository }}/genbi-backend:${{ github.sha }}
            ghcr.io/${{ github.repository }}/genbi-backend:latest
          cache-from: type=gha
          cache-to: type=gha,mode=max

      - name: Build & push frontend image
        uses: docker/build-push-action@v5
        with:
          context: frontend
```

```
          push: true
          tags: |
            ghcr.io/${{ github.repository }}/genbi-frontend:${{ github.sha }}
            ghcr.io/${{ github.repository }}/genbi-frontend:latest
          cache-from: type=gha
          cache-to: type=gha,mode=max

  # ----------------------------------------------------------------
  # Golden Eval Gate (blocking for main)
  # ----------------------------------------------------------------
  eval-gate:
    name: Golden Eval Gate (>= 90%)
    runs-on: ubuntu-latest
    if: github.ref == 'refs/heads/main'
    needs: [backend-lint-test]
    defaults:
      run:
        working-directory: backend

    services:
      postgres:
        image: pgvector/pgvector:pg16
        env:
          POSTGRES_USER: genbi
          POSTGRES_PASSWORD: genbi
          POSTGRES_DB: genbi_test
        ports:
          - 5432:5432

    steps:
      - uses: actions/checkout@v4

      - name: Install uv
        uses: astral-sh/setup-uv@v2
        with:
          python-version: ${{ env.PYTHON_VERSION }}

      - name: Install dependencies
        run: uv sync --frozen

      - name: Run golden eval suite
        env:
          DATABASE_URL: postgresql+asyncpg://genbi:genbi@localhost:5432/genbi_test
          ANTHROPIC_API_KEY: ${{ secrets.ANTHROPIC_API_KEY }}
          APP_ENV: test
        run: |
          uv run pytest tests/evals/test_nl2sql_eval.py -v --tb=short
          echo "Golden eval suite passed (>= 90% threshold)"
```

.gitignore

56 lines · 0.5 KB

```
# Python
__pycache__/
*.py[cod]
*$py.class
*.egg-info/
dist/
build/
*.egg
.venv/
venv/
.env
*.env.local

# Node
node_modules/
.next/
out/
*.tsbuildinfo
npm-debug.log*
pnpm-debug.log*

# IDEs
.idea/
.vscode/
*.swp
*.swo
*~

# OS
.DS_Store
Thumbs.db

# Docker
docker-data/

# Test
.coverage
htmlcov/
.pytest_cache/
*.cover

# Langfuse
.langfuse/

# DBT
semantic/dbt/logs/
semantic/dbt/target/

# Certificates
*.pem
*.key
*.crt
!infra/certs/.gitkeep

# Graphify (knowledge graph output)
graphify-out/
```

backend/alembic.ini

50 lines · 0.9 KB

```
# Alembic configuration for GenBI.
#
# The SQLAlchemy URL is intentionally left empty — it is overridden at runtime
# by backend/app/db/migrations/env.py from settings.DATABASE_URL_SYNC.
#
# Run:
#   cd backend
#   uv run alembic upgrade head     # apply migrations
#   uv run alembic current          # show current revision

[alembic]
script_location = app/db/migrations
prepend_sys_path = .
sqlalchemy.url =

[post_write_hooks]

[loggers]
keys = root,sqlalchemy,alembic

[handlers]
keys = console

[formatters]
keys = generic

[logger_root]
level = WARN
handlers = console
qualname =

[logger_sqlalchemy]
level = WARN
handlers =
qualname = sqlalchemy.engine

[logger_alembic]
level = INFO
handlers =
qualname = alembic

[handler_console]
class = StreamHandler
args = (sys.stderr,)
level = NOTSET
formatter = generic

[formatter_generic]
format = %(levelname)-5.5s [%(name)s] %(message)s
datefmt = %H:%M:%S
```

backend/app/__init__.py

0 lines · 0.0 KB

```
1|
```

backend/app/agents/__init__.py

0 lines · 0.0 KB

```
1|
```

backend/app/agents/base.py

63 lines · 1.8 KB

```python
"""Agent base classes — all GenBI agents inherit from BaseAgent."""

from abc import ABC, abstractmethod
from dataclasses import dataclass, field
from datetime import UTC, datetime
from uuid import uuid4


@dataclass
class AgentConfig:
    """Configuration for an agent instance."""

    model_name: str
    temperature: float = 0.0
    max_tokens: int = 4096
    thinking: bool = False  # Use Claude extended thinking for reasoning tasks


@dataclass
class AgentResult:
    """Standard result envelope from any agent execution."""

    agent_name: str
    success: bool
    output: dict
    input_tokens: int = 0
    output_tokens: int = 0
    latency_ms: float = 0
    model_version: str = ""
    warnings: list[str] = field(default_factory=list)
    errors: list[str] = field(default_factory=list)

    @property
    def total_tokens(self) -> int:
        return self.input_tokens + self.output_tokens


class BaseAgent(ABC):
    """Abstract base for all GenBI agents.

    Every agent must:
    - Accept typed input via `execute(**kwargs)`
    - Return a standardized `AgentResult`
    - Log input/output tokens, latency, and model version for audit
    - Never raise raw exceptions — errors go in AgentResult.errors
    """

    name: str = "base_agent"
    description: str = "Base agent class"

    def __init__(self, config: AgentConfig):
        self.config = config
        self._run_id = str(uuid4())

    @abstractmethod
    async def execute(self, **kwargs) -> AgentResult:
        """Execute the agent's core logic. Must be overridden by subclasses."""
        ...

    def _timed_result(self, result: AgentResult, start_time: float) -> AgentResult:
        """Attach latency and run metadata to an AgentResult."""
        result.latency_ms = (datetime.now(UTC).timestamp() - start_time) * 1000
        return result
```

backend/app/agents/chart/__init__.py

1 lines · 0.1 KB

```
1| """Chart integration package — Flint MCP bridge, AWEL operator, and vis-flint protocol."""
```

backend/app/agents/chart/flint_bridge.py

329 lines · 11.6 KB

```python
"""Flint Chart integration module for GenBI.

This module bridges GenBI's agent pipeline to Microsoft Flint Chart MCP,
providing chart rendering via a unified ChartAssemblyInput spec that compiles
to Vega-Lite, ECharts, or Chart.js.

Architecture:
    GenBI Agent Pipeline
        → ChartGenAgent produces ChartAssemblyInput spec
            → FlintChartBridge (this module) calls Flint MCP
                → Flint MCP server compiles & renders locally
                    → Returns PNG/SVG inline

The bridge also supports:
- AWEL operator integration (FlintChartOperator for workflow canvas)
- vis-flint protocol for embedding charts in chat responses
- Backend selection per-render (Vega-Lite for SVG, ECharts for complex, Chart.js for speed)
"""

import subprocess
import tempfile
from pathlib import Path

from app.core.config import settings
from app.core.logging import logger


class FlintChartBridge:
    """Bridge to the Flint Chart MCP server.

    Flint runs as a Node.js MCP stdio server via `npx -y flint-chart-mcp`.
    This bridge communicates with it via the MCP protocol for chart rendering.

    For the initial integration, we use a simpler approach: call Flint's
    core rendering via subprocess, passing the spec as JSON on stdin.
    As Flint matures (heading toward v1.0), we'll switch to the MCP client
    for persistent connections.
    """

    def __init__(self, tenant_id: str = "default"):
        self.tenant_id = tenant_id
        self._data_root = Path(settings.FLINT_MCP_DATA_ROOTS.split(",")[0].strip())
        self._data_root.mkdir(parents=True, exist_ok=True)

    async def render(
        self,
        spec: dict,
        backend: str = "vegalite",
        output_format: str = "png",
        scale: int = 2,
    ) -> dict:
        """Render a ChartAssemblyInput spec to PNG or SVG.

        Args:
            spec: ChartAssemblyInput specification dict
            backend: One of 'vegalite', 'echarts', 'chartjs'
            output_format: 'png' or 'svg'
            scale: Output scale factor for PNG

        Returns:
            dict with keys: success, format, image_base64/svg, backend, warnings, errors
        """
        logger.info(
            "Rendering chart via Flint",
            backend=backend,
            format=output_format,
            tenant_id=self.tenant_id,
            chart_type=spec.get("chartType", "unknown"),
        )

        # Write data to temp file if inline values exceed threshold
        spec = self._prepare_data(spec)

        try:
            result = await self._call_flint_mcp(spec, backend, output_format, scale)
            result["backend"] = backend
            return result
        except Exception as e:
            logger.error(f"Flint rendering failed: {e}")
            return {
                "success": False,
                "format": output_format,
                "backend": backend,
                "warnings": [],
                "errors": [str(e)],
            }

    def _prepare_data(self, spec: dict) -> dict:
        """If inline data is large, write it to a temp file and use data.url instead.
        This avoids hitting Flint's inline row limits."""
        data = spec.get("data", {})
        values = data.get("values", [])

        if len(values) > 100:  # Threshold for file-based data
            import json

            temp_file = self._data_root / f"genbi_chart_{self.tenant_id}.json"
            temp_file.write_text(json.dumps(values, default=str))
            spec["data"] = {"url": str(temp_file)}
            logger.debug(f"Wrote {len(values)} rows to {temp_file}")
```

```python

        return spec

    async def _call_flint_mcp(
        self,
        spec: dict,
        backend: str,
        output_format: str,
        scale: int,
    ) -> dict:
        """Call Flint MCP to render a chart.

        This uses the MCP protocol via stdin/stdout. For the initial integration,
        we construct the equivalent tool call directly.

        TODO: Replace with proper MCP client session when Flint's MCP server
        supports persistent connections and HTTP/SSE transport.
        """
        # Validate and compile the spec
        validation = await self._validate_spec(spec)
        if validation.get("errors"):
            return {
                "success": False,
                "format": output_format,
                "warnings": validation.get("warnings", []),
                "errors": validation["errors"],
            }

        # For now, render via Vega-Lite directly using the spec
        # This is a local rendering path that bypasses Flint MCP
        # while Flint is in early v0.1.0 — switch to MCP when it stabilizes
        return await self._render_vegalite_fallback(spec, output_format, scale)

    async def _validate_spec(self, spec: dict) -> dict:
        """Validate a ChartAssemblyInput spec before rendering."""
        errors = []
        warnings = []

        required = ["chartType", "data"]
        for key in required:
            if key not in spec:
                errors.append(f"Missing required field: {key}")

        if "chartType" in spec and spec["chartType"] not in self.SUPPORTED_CHART_TYPES:
            warnings.append(
                f"Unknown chart type: {spec['chartType']} — will attempt best-effort rendering"
            )

        data = spec.get("data", {})
        if "values" in data and len(data["values"]) > 10000:
            errors.append(f"Row count {len(data['values'])} exceeds limit of 10,000")

        if "url" in data:
            url = data["url"]
            if url.startswith(("http://", "https://")):
                errors.append(
                    "Remote data URLs are not supported — use inline values or local files"
                )
            elif not any(url.startswith(str(r)) for r in self._data_root.parents):
                warnings.append(f"Data file {url} is outside approved data roots")

        return {"errors": errors, "warnings": warnings}

    async def _render_vegalite_fallback(self, spec: dict, output_format: str, scale: int) -> dict:
        """Fallback renderer using Vega-Lite directly (no Flint MCP required).
        This is the bootstrapping path — replace with actual MCP calls when
        Flint stabilizes."""
        import base64
        from importlib.util import find_spec

        if find_spec("altair") is None:
            return {
                "success": False,
                "format": output_format,
                "warnings": [],
                "errors": ["altair not installed — install with: uv add altair"],
            }

        data = spec.get("data", {}).get("values", [])
        if not data:
            return {
                "success": False,
                "format": output_format,
                "warnings": [],
                "errors": ["No data provided for chart rendering"],
            }

        encodings = spec.get("encodings", {})
        chart_type = spec.get("chartType", "Bar Chart")
        base_size = spec.get("baseSize", {"width": 600, "height": 400})

        import pandas as pd

        df = pd.DataFrame(data)

        chart = self._build_altair_chart(df, chart_type, encodings, base_size)

        if output_format == "svg":
            svg_str = chart.to_json(format="svg")
            return {
                "success": True,
                "format": "svg",
                "svg": svg_str,
                "warnings": [],
                "errors": [],
```

```
            }

        # PNG via Vega-Lite -> SVG -> resvg

        with tempfile.NamedTemporaryFile(suffix=".svg", delete=False) as f:
            f.write(chart.to_json(format="svg").encode())
            svg_path = f.name

        png_path = svg_path.replace(".svg", ".png")
        try:
            subprocess.run(
                ["resvg", svg_path, png_path, "-w", str(base_size["width"] * scale)],
                check=True,
                capture_output=True,
                timeout=30,
            )
            png_data = Path(png_path).read_bytes()
            image_base64 = base64.b64encode(png_data).decode()
            Path(svg_path).unlink(missing_ok=True)
            Path(png_path).unlink(missing_ok=True)
            return {
                "success": True,
                "format": "png",
                "image_base64": image_base64,
                "warnings": [],
                "errors": [],
            }
        except (subprocess.CalledProcessError, FileNotFoundError):
            # resvg not available — return SVG even for PNG requests
            svg_data = Path(svg_path).read_text()
            Path(svg_path).unlink(missing_ok=True)
            return {
                "success": True,
                "format": "svg",
                "svg": svg_data,
                "warnings": ["resvg not installed — returned SVG instead of PNG"],
                "errors": [],
            }

    def _build_altair_chart(self, df, chart_type: str, encodings: dict, base_size: dict):
        """Build an Altair/Vega-Lite chart from a simplified Flint-inspired spec."""
        import altair as alt

        x_field = encodings.get("x", {}).get("field", df.columns[0])
        y_field = encodings.get("y", {}).get("field", df.columns[-1])
        color_field = encodings.get("color", {}).get("field")

        width = base_size.get("width", 600)
        height = base_size.get("height", 400)

        chart_map = {
            "Bar Chart": lambda: (
                alt.Chart(df)
                .mark_bar()
                .encode(
                    x=alt.X(f"{x_field}:N", title=x_field),
                    y=alt.Y(f"{y_field}:Q", title=y_field),
                    color=alt.value("#4C78A8")
                    if not color_field
                    else alt.Color(f"{color_field}:N"),
                )
            ),
            "Line Chart": lambda: (
                alt.Chart(df)
                .mark_line(point=True)
                .encode(
                    x=alt.X(
                        f"{x_field}:T" if "date" in x_field.lower() else f"{x_field}:O",
                        title=x_field,
                    ),
                    y=alt.Y(f"{y_field}:Q", title=y_field),
                    color=alt.Color(f"{color_field}:N") if color_field else alt.Undefined,
                )
            ),
            "Scatter Chart": lambda: (
                alt.Chart(df)
                .mark_circle(size=60)
                .encode(
                    x=alt.X(f"{x_field}:Q", title=x_field),
                    y=alt.Y(f"{y_field}:Q", title=y_field),
                    color=alt.Color(f"{color_field}:N") if color_field else alt.Undefined,
                )
            ),
            "Area Chart": lambda: (
                alt.Chart(df)
                .mark_area(opacity=0.5)
                .encode(
                    x=alt.X(
                        f"{x_field}:T" if "date" in x_field.lower() else f"{x_field}:O",
                        title=x_field,
                    ),
                    y=alt.Y(f"{y_field}:Q", title=y_field),
                )
            ),
            "Pie Chart": lambda: (
                alt.Chart(df)
                .mark_arc()
                .encode(
                    theta=alt.Theta(f"{y_field}:Q"),
                    color=alt.Color(f"{x_field}:N"),
                )
            ),
        }

        builder = chart_map.get(chart_type, chart_map["Bar Chart"])
```

```
        chart = builder()
        return chart.properties(width=width, height=height)

    SUPPORTED_CHART_TYPES = {
        "Bar Chart",
        "Stacked Bar Chart",
        "Grouped Bar Chart",
        "Line Chart",
        "Multi-series Line Chart",
        "Scatter Chart",
        "Scatter Plot",
        "Area Chart",
        "Stacked Area Chart",
        "Pie Chart",
        "Donut Chart",
        "Histogram",
        "Box Plot",
        "Heatmap",
    }
```

backend/app/agents/chart/flint_operator.py

95 lines · 3.0 KB

```python
"""FlintChartOperator — AWEL-compatible operator for the Flow Canvas.

This operator wraps FlintChartBridge into a reusable DAG node that can be
dropped into DB-GPT-style visual workflows. It:

1. Accepts a DataFrame from a preceding SQL execution operator
2. Introspects column types and suggests chart type + encodings
3. Generates a ChartAssemblyInput spec via LLM (if needed)
4. Renders the chart via FlintChartBridge
5. Returns a ChartResult (PNG/SVG) to downstream operators

The operator is designed to follow DB-GPT's AWEL MapOperator pattern
so it can be registered on the Flow Canvas alongside existing operators.
"""

from dataclasses import dataclass, field


@dataclass
class ChartOperatorInput:
    """Input to the FlintChartOperator — typically comes from SQL execution."""

    data: list[dict] = field(default_factory=list)
    chart_type_hint: str | None = None
    title: str | None = None
    description: str | None = None


@dataclass
class ChartOperatorOutput:
    """Output from the FlintChartOperator — consumed by report assembly or chat response."""

    success: bool
    spec: dict = field(default_factory=dict)
    image_base64: str | None = None
    svg: str | None = None
    format: str = "png"
    backend: str = "vegalite"
    warnings: list[str] = field(default_factory=list)
    errors: list[str] = field(default_factory=list)


async def flint_chart_operator(
    input_data: ChartOperatorInput,
    tenant_id: str = "default",
    backend: str = "vegalite",
    output_format: str = "png",
) -> ChartOperatorOutput:
    """Execute the Flint chart operator.

    This function is the AWEL operator entry point. In a full DB-GPT deployment,
    it would be wrapped as a MapOperator subclass with ViewMetadata for the canvas.

    Usage in AWEL:
        with DAG("sales_report") as dag:
            sql_node = SQLOperator(...)
            chart_node = MapOperator.from_fn(flint_chart_operator)
            sql_node >> chart_node
    """
    from app.agents.base import AgentConfig
    from app.agents.chart.flint_bridge import FlintChartBridge
    from app.agents.chart_gen_agent import ChartGenAgent
    from app.core.config import settings

    # Generate spec
    agent = ChartGenAgent(
        config=AgentConfig(
            model_name=settings.LLM_FAST_MODEL,
            temperature=0,
        )
    )

    result = await agent._generate_chart_spec(
        data=input_data.data,
        preferred_chart_type=input_data.chart_type_hint,
    )

    # Render
    bridge = FlintChartBridge(tenant_id=tenant_id)
    render_result = await bridge.render(
        spec=result,
        backend=backend,
        output_format=output_format,
    )

    return ChartOperatorOutput(
        success=render_result.get("success", False),
        spec=result,
        image_base64=render_result.get("image_base64"),
        svg=render_result.get("svg"),
        format=output_format,
        backend=backend,
        warnings=render_result.get("warnings", []),
        errors=render_result.get("errors", []),
    )
```

backend/app/agents/chart/vis_flint_protocol.py

124 lines · 3.8 KB

```python
"""vis-flint protocol handler — extends the GPT-Vis protocol for Flint charts.

DB-GPT's GPT-Vis protocol uses markdown code blocks with language identifiers
like ```vis-dashboard``` and ```vis_chart``` that the frontend dispatches to
AntV G2 components.

This module extends that protocol with a new ```vis-flint``` block type that
routes to Flint MCP for rendering. The frontend can embed the resulting
PNG/SVG alongside interactive G2 charts.

Usage in agent response:
    ```vis-flint
    {
      "chartType": "Bar Chart",
      "encodings": {
        "x": {"field": "region"},
        "y": {"field": "revenue"}
      },
      "data": {
        "values": [...]
      }
    }
    ```

The backend chat pipeline detects these blocks, renders them via Flint,
and replaces them with inline base64 images or SVG markup.
"""

import re
from dataclasses import dataclass

@dataclass
class VisFlintBlock:
    """Parsed vis-flint code block from an agent response."""

    raw_json: str
    spec: dict
    start_pos: int
    end_pos: int

def parse_vis_flint_blocks(text: str) -> list[VisFlintBlock]:
    """Extract all ```vis-flint``` code blocks from agent output text.

    Returns a list of parsed blocks with their positions in the original text,
    so the caller can replace them with rendered chart content.
    """
    import json

    pattern = re.compile(r"```vis-flint\n(.*?)```", re.DOTALL)
    blocks = []

    for match in pattern.finditer(text):
        raw = match.group(1).strip()
        try:
            spec = json.loads(raw)
        except json.JSONDecodeError:
            spec = {"error": "Invalid JSON in vis-flint block"}
        blocks.append(
            VisFlintBlock(
                raw_json=raw,
                spec=spec,
                start_pos=match.start(),
                end_pos=match.end(),
)
)

    return blocks

async def render_vis_flint_blocks(text: str, tenant_id: str = "default") -> str:
    """Process a response text, rendering all vis-flint blocks to inline charts.

    Each ```vis-flint``` block is replaced with:
    - An inline SVG (for SVG output) or
    - A markdown image with base64 data URI (for PNG output)

    Args:
        text: The raw agent response text containing vis-flint blocks
        tenant_id: Tenant identifier for data isolation

    Returns:
        The same text with vis-flint blocks replaced by rendered chart content
    """
    from app.agents.chart.flint_bridge import FlintChartBridge

    blocks = parse_vis_flint_blocks(text)
    if not blocks:
        return text

    bridge = FlintChartBridge(tenant_id=tenant_id)
    result_parts = []
    last_end = 0

    for block in blocks:
        # Keep text before this block
        result_parts.append(text[last_end : block.start_pos])

        if "error" in block.spec:
```

```
            result_parts.append(f"```vis-flint (parse error)\n{block.raw_json}\n```")
        else:
            render_result = await bridge.render(block.spec, backend="vegalite", output_format="svg")

            if render_result.get("svg"):
                # Embed SVG directly — works in most markdown renderers
                result_parts.append(f'<div class="vis-flint-chart">{render_result["svg"]}</div>')
            elif render_result.get("image_base64"):
                # Embed PNG as data URI
                result_parts.append(
                    f'<img src="{render_result["image_base64"]}" '
                    f'alt="{block.spec.get("chartType", "Chart")}" '
                    f'class="vis-flint-chart" />'
                )
            else:
                # Render failed — keep original block with error annotation
                result_parts.append(f"```vis-flint (render failed)\n{block.raw_json}\n```")

        last_end = block.end_pos

    # Append remaining text after last block
    result_parts.append(text[last_end:])

    return "".join(result_parts)
```

backend/app/agents/chart_gen_agent.py

226 lines · 8.3 KB

```python
"""Chart generation agent — produces ChartAssemblyInput specs for Flint MCP.

This is the upgraded ChartGenAgent that replaces the old G2-only approach.
It produces Flint-first ChartAssemblyInput specs but can fall back to the
GPT-Vis protocol for interactive dashboard charts.
"""

from app.agents.base import AgentResult, BaseAgent
from app.agents.chart.flint_bridge import FlintChartBridge
from app.agents.registry import register_agent


@register_agent("chart_gen")
class ChartGenAgent(BaseAgent):
    """Generates chart specifications from data using Flint MCP.

    Input: SQL result set + optional chart type hint
    Output: ChartAssemblyInput spec + rendered PNG/SVG
    """

    name = "chart_gen"
    description = "Chart specification generator and renderer (Flint MCP)"

    async def execute(
        self,
        data: list[dict] | None = None,
        query: str | None = None,
        intent: str | None = None,
        preferred_chart_type: str | None = None,
        tenant_id: str = "default",
        user_id: str = "",
        **kwargs,
    ) -> AgentResult:
        """Generate a chart spec from data or description."""
        import time

        start = time.time()

        # If no data provided, this is a standalone chart request
        if data is None and query:
            return await self._handle_standalone_chart_request(query, start)

        if data is None:
            return AgentResult(
                agent_name=self.name,
                success=False,
                output={},
                errors=["No data or query provided to ChartGenAgent"],
            )

        # Generate chart spec from data
        spec = await self._generate_chart_spec(
            data, preferred_chart_type, tenant_id=tenant_id, user_id=user_id
        )

        # Render via Flint MCP
        bridge = FlintChartBridge(tenant_id=tenant_id)
        render_result = await bridge.render(spec)

        return self._timed_result(
            AgentResult(
                agent_name=self.name,
                success=render_result.get("success", False),
                output={
                    "chart_spec": spec,
                    "image_base64": render_result.get("image_base64"),
                    "svg": render_result.get("svg"),
                    "backend": render_result.get("backend"),
                },
                warnings=render_result.get("warnings", []),
                errors=render_result.get("errors", []),
            ),
            start,
        )

    async def _generate_chart_spec(
        self,
        data: list[dict],
        preferred_chart_type: str | None = None,
        tenant_id: str = "default",
        user_id: str = "",
    ) -> dict:
        """Generate a ChartAssemblyInput spec using the centralized LLM client.

        Routes through llm_client (not a direct ChatAnthropic instance) so the
        call gets retry, token-budget enforcement, JSON extraction, and audit
        logging — required by CLAUDE.md §6 for every LLM call.
        """
        import json

        from app.core.llm_client import LLMCallOptions, get_llm_client

        # Infer semantic types from data
        semantic_types = self._infer_semantic_types(data)
        chart_type = preferred_chart_type or self._suggest_chart_type(data, semantic_types)
        encodings = self._suggest_encodings(data, semantic_types, chart_type)

        prompt = f"""Generate a Flint ChartAssemblyInput specification for this data.

DATA (first 10 rows):
```

```
{json.dumps(data[:10], ensure_ascii=False, default=str)}

SEMANTIC TYPES INFERRED:
{json.dumps(semantic_types)}

SUGGESTED CHART TYPE: {chart_type}
SUGGESTED ENCODINGS: {json.dumps(encodings)}

Return ONLY a valid JSON object with these keys: chartType, encodings, baseSize, data.
The data key should contain the full data as inline values.
Do NOT include any explanatory text — only the JSON object."""

        result = await get_llm_client().invoke(
            messages=prompt,
            use_reasoning=False,  # fast model for spec generation
            options=LLMCallOptions(
                temperature=0,
                max_tokens=2048,
                response_format="json",
            ),
            tenant_id=tenant_id,
            user_id=user_id,
        )

        # The client already extracts JSON via _extract_json; fall back only if
        # the model returned nothing parseable.
        if result.parsed:
            return result.parsed
        try:
            return json.loads(result.content)
        except (json.JSONDecodeError, TypeError):
            return {
                "chartType": chart_type,
                "encodings": encodings,
                "baseSize": {"width": 600, "height": 400},
                "data": {"values": data},
            }

    def _infer_semantic_types(self, data: list[dict]) -> dict[str, str]:
        """Infer Flint semantic types (Category, Quantity, Temporal) from column data."""
        if not data:
            return {}

        types = {}
        for col in data[0]:
            sample_values = [row.get(col) for row in data[:20] if row.get(col) is not None]
            if not sample_values:
                types[col] = "Category"
                continue

            if all(isinstance(v, (int, float)) for v in sample_values):
                # Check if it could be a year/timestamp
                if all(1000 <= v <= 2100 for v in sample_values):
                    types[col] = "Temporal"
                else:
                    types[col] = "Quantity"
            elif all(isinstance(v, str) for v in sample_values):
                # Check for date-like strings
                import re

                date_pattern = re.compile(r"\d{4}-\d{2}-\d{2}")
                if all(date_pattern.match(str(v)) for v in sample_values):
                    types[col] = "Temporal"
                else:
                    types[col] = "Category"
            else:
                types[col] = "Category"

        return types

    def _suggest_chart_type(self, data: list[dict], semantic_types: dict[str, str]) -> str:
        """Suggest a chart type based on data characteristics."""
        categories = [k for k, v in semantic_types.items() if v == "Category"]
        quantities = [k for k, v in semantic_types.items() if v == "Quantity"]
        temporals = [k for k, v in semantic_types.items() if v == "Temporal"]

        if temporals and quantities:
            return "Line Chart"
        if len(categories) == 1 and len(quantities) == 1:
            return "Bar Chart"
        if len(quantities) >= 2:
            return "Scatter Chart"
        if len(categories) == 1 and len(quantities) == 1 and len(data) <= 8:
            return "Pie Chart"
        return "Bar Chart"

    def _suggest_encodings(
        self, data: list[dict], semantic_types: dict[str, str], chart_type: str
    ) -> dict:
        """Suggest visual encodings based on semantic types and chart type."""
        categories = [k for k, v in semantic_types.items() if v == "Category"]
        quantities = [k for k, v in semantic_types.items() if v == "Quantity"]
        temporals = [k for k, v in semantic_types.items() if v == "Temporal"]

        encodings = {}
        # x-axis: temporal > category
        x_field = temporals[0] if temporals else (categories[0] if categories else None)
        if x_field:
            encodings["x"] = {"field": x_field}

        # y-axis: first quantity
        if quantities:
            encodings["y"] = {"field": quantities[0]}

        # color: second category or second quantity for multi-series
```

```
        if len(categories) > 1:
            encodings["color"] = {"field": categories[1]}
        elif len(quantities) > 1 and chart_type == "Scatter Chart":
            encodings["color"] = {"field": quantities[1]}

        return encodings

    async def _handle_standalone_chart_request(self, query: str, start: float) -> AgentResult:
        """Handle a chart request without pre-existing data — the agent needs to generate
        a chart spec directly from the user's description (e.g., inline data in the prompt)."""
        # This path is for when the user provides data inline or describes a chart
        # without needing SQL first
        return self._timed_result(
            AgentResult(
                agent_name=self.name,
                success=False,
                output={},
                errors=["Standalone chart generation without data not yet implemented"],
            ),
            start,
        )
```

backend/app/agents/narrative/__init__.py

1 lines · 0.1 KB

```
1 """Narrative agent — generates insight paragraphs from data."""
```

backend/app/agents/narrative/narrative_agent.py

213 lines · 7.5 KB

```python
"""NarrativeAgent — data storytelling and insight generation.

Writes concise, actionable insight paragraphs from data and chart context.
Designed for business stakeholders: lead with the finding, use specific numbers,
and include a suggested action.

Model: claude-haiku-4 (fast, direct generation)
Prompt: .claude/prompts/narrative-system.md
"""

import re
import time
from uuid import uuid4

from app.agents.base import AgentResult, BaseAgent
from app.agents.registry import register_agent
from app.core.llm_client import LLMCallOptions, get_llm_client, load_prompt
from app.core.logging import logger


@register_agent("narrative")
class NarrativeAgent(BaseAgent):
    """Generates data insight narratives from query results and charts.

    Input: User's original question + data summary + chart context
    Output: 3-5 sentence insight paragraph in flowing prose
    """

    name = "narrative"
    description = "Data storyteller — generates concise insight paragraphs"

    async def execute(
        self,
        query: str = "",
        data_summary: dict | None = None,
        chart_context: dict | None = None,
        sql: str = "",
        tenant_id: str = "default",
        user_id: str = "",
        session_id: str = "",
        **kwargs,
    ) -> AgentResult:
        """Generate a narrative insight from data and chart.

        Args:
            query: The user's original natural language question.
            data_summary: Summary statistics from the query result (rows, totals, trends).
            chart_context: Chart type, encodings, and key visual patterns.
            sql: The generated SQL (for context, not included in narrative).
            tenant_id: Multi-tenant identifier.
            user_id: User identifier for audit logging.
            session_id: Session identifier for audit logging.
        """
        start = time.time()
        run_id = str(uuid4())
        session_id = session_id or run_id

        logger.info(
            "NarrativeAgent starting",
            run_id=run_id,
            has_data=bool(data_summary),
            has_chart=bool(chart_context),
        )

        if not data_summary and not chart_context:
            return self._timed_result(
                AgentResult(
                    agent_name=self.name,
                    success=False,
                    output={},
                    errors=["No data summary or chart context provided to NarrativeAgent"],
                ),
                start,
            )

        # Build the prompt
        system_prompt = load_prompt("narrative-system")
        user_message = self._build_user_message(
            query=query,
            data_summary=data_summary,
            chart_context=chart_context,
        )

        try:
            client = get_llm_client()
            result = await client.invoke(
                messages=user_message,
                system=system_prompt,
                use_reasoning=False,  # Haiku — fast, direct generation
                options=LLMCallOptions(
                    temperature=0.3,  # Slight warmth for natural prose
                    max_tokens=512,  # 3-5 sentences
                    timeout_seconds=30,
                    max_retries=2,
                ),
                user_id=user_id,
                tenant_id=tenant_id,
                session_id=session_id,
            )

```

```
            narrative = result.content.strip()

            # Quality checks
            checks = self._quality_checks(narrative)

            return self._timed_result(
                AgentResult(
                    agent_name=self.name,
                    success=bool(narrative),
                    output={
                        "narrative": narrative,
                        "query": query,
                        "run_id": run_id,
                    },
                    input_tokens=result.input_tokens,
                    output_tokens=result.output_tokens,
                    model_version=result.model_name,
                    warnings=checks,
                ),
                start,
            )

        except Exception as e:
            logger.error(f"NarrativeAgent failed: {e}")
            return self._timed_result(
                AgentResult(
                    agent_name=self.name,
                    success=False,
                    output={"run_id": run_id},
                    errors=[str(e)],
                ),
                start,
            )

    def _build_user_message(
        self,
        query: str,
        data_summary: dict | None,
        chart_context: dict | None,
    ) -> str:
        """Build the user message with all available context."""
        import json

        parts = []

        # User's question
        if query:
            parts.append(f"**User asked**: {query}")

        # Data summary
        if data_summary:
            parts.append("\n## Data Summary\n")
            if data_summary.get("head"):
                parts.append("### Sample Rows")
                parts.append("```json")
                parts.append(json.dumps(data_summary["head"][:5], default=str))
                parts.append("```")

            stats = []
            if data_summary.get("row_count"):
                stats.append(f"- Total rows: {data_summary['row_count']}")
            if data_summary.get("total_revenue"):
                stats.append(f"- Total revenue: ${data_summary['total_revenue']:,.2f}")
            if data_summary.get("avg_value"):
                stats.append(f"- Average: {data_summary['avg_value']:,.2f}")
            if data_summary.get("trend"):
                stats.append(f"- Trend: {data_summary['trend']}")
            if data_summary.get("notable_outlier"):
                stats.append(f"- Notable outlier: {data_summary['notable_outlier']}")

            if stats:
                parts.append("\n### Key Statistics\n" + "\n".join(stats))

        # Chart context
        if chart_context:
            parts.append("\n## Chart Context\n")
            parts.append(f"- Chart type: {chart_context.get('chartType', 'unknown')}")
            if chart_context.get("encodings"):
                enc = chart_context["encodings"]
                parts.append(f"- X-axis: {enc.get('x', {}).get('field', 'N/A')}")
                parts.append(f"- Y-axis: {enc.get('y', {}).get('field', 'N/A')}")
                if enc.get("color"):
                    parts.append(f"- Color/Series: {enc['color'].get('field', 'N/A')}")

        return "\n".join(parts)

    def _quality_checks(self, narrative: str) -> list[str]:
        """Run quality checks on the generated narrative.

        Returns advisory warnings — the narrative is still returned even with warnings.
        """
        warnings = []

        # Check length (3-5 sentences is ideal)
        sentences = [
            s.strip() for s in narrative.replace("!", ".").replace("?", ".").split(".") if s.strip()
        ]
        if len(sentences) < 2:
            warnings.append("Narrative is short (< 2 sentences) — may lack actionable insight")
        if len(sentences) > 8:
            warnings.append(
                "Narrative is long (> 8 sentences) — consider tightening to key findings"
            )

        # Check for vague language
```

```
        vague_terms = ["many", "large", "small", "significant", "a lot", "various", "several"]
        found_vague = [t for t in vague_terms if re.search(rf"\b{t}\b", narrative.lower())]
        if found_vague:
            warnings.append(
                f"Vague terms found: {', '.join(found_vague)} — use specific numbers instead"
            )

        return warnings
```

backend/app/agents/nl2sql/__init__.py

1 lines · 0.1 KB

```
1| """NL2SQL agent — converts natural language to read-only SQL queries."""
```

backend/app/agents/nl2sql/nl2sql_agent.py

274 lines · 10.2 KB

```python
"""NL2SQLAgent -- natural language to SQL query generation.

Uses schema context compression (top-k relevant tables via pgvector) and
few-shot examples to generate safe, read-only SQL queries.

Model: claude-opus-4 (reasoning with thinking blocks)
Prompt: .claude/prompts/nl2sql-system.md
"""

import json
import time
from uuid import uuid4

from app.agents.base import AgentResult, BaseAgent
from app.agents.registry import register_agent
from app.core.llm_client import LLMCallOptions, get_llm_client, load_prompt
from app.core.logging import logger


@register_agent("nl2sql")
class NL2SQLAgent(BaseAgent):
    """Generates read-only SQL queries from natural language questions.

    Input: User prompt + schema context (top-k tables/columns from pgvector)
    Output: SQL query + explanation + table references + assumptions
    """

    name = "nl2sql"
    description = "Natural language to SQL generator with schema-aware context"

    # Destructive SQL patterns that must NEVER appear in generated queries
    DESTRUCTIVE_PATTERNS = [
        r"\bDROP\s+(TABLE|DATABASE|SCHEMA|INDEX|VIEW|FUNCTION|PROCEDURE)\b",
        r"\bDELETE\s+FROM\b",
        r"\bTRUNCATE\s+(TABLE\s+)?",
        r"\bALTER\s+TABLE\b.*\bDROP\b",
        r"\bALTER\s+TABLE\b.*\bRENAME\b",
        r"\bUPDATE\s+\w+\s+SET\b",
        r"\bINSERT\s+INTO\b",
        r"\bCREATE\s+(USER|ROLE|GROUP)\b",
        r"\bGRANT\b",
        r"\bREVOKE\b",
        r"\bCOPY\b.*\bFROM\b",
    ]

    async def execute(
        self,
        query: str,
        schema_context: list[dict] | None = None,
        few_shot_examples: list[dict] | None = None,
        metric_definitions: list[dict] | str | None = None,
        tenant_id: str = "default",
        user_id: str = "",
        session_id: str = "",
        history: list[dict] | None = None,
        **kwargs,
    ) -> AgentResult:
        """Generate SQL from a natural language query.

        Args:
            query: The user\'s natural language question.
            schema_context: Top-k relevant tables/columns from pgvector search.
            few_shot_examples: Similar validated NL/SQL pairs for in-context learning.
            metric_definitions: Business metrics from the semantic layer (Cube.dev).
                Can be a pre-formatted string from CubeClient.get_agent_context()
                or a legacy list of dicts.
            tenant_id: Multi-tenant identifier for data isolation.
            user_id: User identifier for audit logging.
            session_id: Session identifier for audit logging.
            history: Prior conversation turns (Phase 14) — enables follow-up
                queries like "now break that down by region". Each dict has
                role/content/generated_sql keys.
        """
        start = time.time()
        run_id = str(uuid4())
        session_id = session_id or run_id

        logger.info(
            "NL2SQLAgent starting",
            run_id=run_id,
            query=query[:200],
            tables=len(schema_context or []),
            examples=len(few_shot_examples or []),
            history_turns=len(history or []),
            tenant_id=tenant_id,
        )

        # Build the prompt
        system_prompt = load_prompt("nl2sql-system")
        user_message = self._build_user_message(
            query=query,
            schema_context=schema_context,
            few_shot_examples=few_shot_examples,
            metric_definitions=metric_definitions,
            tenant_id=tenant_id,
            history=history,
        )

        try:
            client = get_llm_client()
```

```
            result = await client.invoke(
                messages=user_message,
                system=system_prompt,
                use_reasoning=True,
                options=LLMCallOptions(
                    temperature=0.0,
                    max_tokens=4096,
                    response_format="json",
                    timeout_seconds=90,
                    max_retries=3,
                ),
                user_id=user_id,
                tenant_id=tenant_id,
                session_id=session_id,
            )

            # Parse the structured output
            output = result.parsed or {}
            sql = output.get("sql", "")

            # Destructive pattern check
            safety_warnings = self._check_destructive_patterns(sql)

            # Build final result
            return self._timed_result(
                AgentResult(
                    agent_name=self.name,
                    success=bool(sql) and not safety_warnings,
                    output={
                        "sql": sql,
                        "explanation": output.get("explanation", ""),
                        "tables_used": output.get("tables_used", []),
                        "assumptions": output.get("assumptions", []),
                        "run_id": run_id,
                    },
                    input_tokens=result.input_tokens,
                    output_tokens=result.output_tokens,
                    model_version=result.model_name,
                    warnings=output.get("warnings", []) + safety_warnings,
                    errors=[] if sql else ["No SQL generated"],
                ),
                start,
            )

        except Exception as e:
            logger.error(f"NL2SQLAgent failed: {e}")
            return self._timed_result(
                AgentResult(
                    agent_name=self.name,
                    success=False,
                    output={"run_id": run_id},
                    errors=[str(e)],
                ),
                start,
            )

    def _build_user_message(
        self,
        query: str,
        schema_context: list[dict] | None,
        few_shot_examples: list[dict] | None,
        metric_definitions: list[dict] | str | None,
        tenant_id: str,
        history: list[dict] | None = None,
    ) -> str:
        """Build the full user message with schema context, examples, and metrics."""
        parts = []

        # 1. The user's question
        parts.append(f"## User Question\n\n{query}")

        # 1b. Conversation history (Phase 14): prior turns give referents for
        # follow-up questions ("now break that down by region"). The CURRENT
        # question always takes precedence.
        if history:
            parts.append("\n## Conversation History (previous turns)\n")
            for turn in history[-6:]:
                role = turn.get("role", "user")
                content = str(turn.get("content", ""))[:500]
                if role == "user":
                    parts.append(f"User: {content}")
                else:
                    sql = turn.get("generated_sql")
                    suffix = f" [SQL: {sql[:200]}]" if sql else ""
                    parts.append(f"Assistant: {content[:300]}{suffix}")

        # 2. Schema context (compressed top-k)
        if schema_context:
            parts.append("\n## Relevant Schema Context\n")
            for table in schema_context:
                table_name = table.get("table_name", "unknown")
                columns = table.get("columns", [])
                parts.append(f"### Table: {table_name}")
                for col in columns:
                    col_name = col.get("column_name", col) if isinstance(col, dict) else col
                    col_desc = col.get("description", "") if isinstance(col, dict) else ""
                    parts.append(f"- `{col_name}`" + (f" -- {col_desc}" if col_desc else ""))
                if table.get("sample_rows"):
                    parts.append(f"\nSample rows ({len(table['sample_rows'])}):")
                    parts.append("```json")
                    parts.append(json.dumps(table["sample_rows"][:3], default=str))
                    parts.append("```")
        else:
            parts.append("\n## Schema Context\n\nNo schema context available.")

```

```
        # 3. Metric definitions (from Cube.dev semantic layer)
        if metric_definitions:
            if isinstance(metric_definitions, str):
                parts.append(metric_definitions)
            else:
                parts.append("\n## Available Metrics\n")
                for metric in metric_definitions:
                    parts.append(f"- **{metric['name']}**: {metric.get('description', '')}")

        # 4. Few-shot examples
        if few_shot_examples:
            parts.append("\n## Example Queries\n")
            for i, example in enumerate(few_shot_examples[:5]):
                parts.append(f"### Example {i + 1}")
                parts.append(f"Question: {example['nl_query']}")
                parts.append(f"SQL: {example['expected_sql']}")

        # 5. Tenant context
        parts.append(f"\n## Session Context\n- Tenant ID: {tenant_id}")
        parts.append("- Access: Read-only SELECT queries only")
        parts.append("- Row Limit: 1000 (unless user specifies otherwise)")
        parts.append("- Schema: Use fully qualified table names (schema.table)")

        return "\n".join(parts)

    def _check_destructive_patterns(self, sql: str) -> list[str]:
        """Check generated SQL for destructive patterns."""
        import re

        if not sql:
            return []

        warnings = []
        sql_upper = sql.upper()

        for pattern in self.DESTRUCTIVE_PATTERNS:
            if re.search(pattern, sql_upper):
                warnings.append(f"Destructive pattern detected: {pattern}")

        return warnings


# ---------------------------------------------------------------------------
# Schema context retrieval (pgvector semantic search)
# ---------------------------------------------------------------------------


async def retrieve_schema_context(
    query: str,
    tenant_id: str,
    db_session=None,
    top_k: int = 5,
) -> list[dict]:
    """Retrieve the top-k most relevant tables for a query."""

    logger.info("Retrieving schema context", query=query[:100], top_k=top_k, tenant_id=tenant_id)
    logger.warning("Schema context retrieval not yet implemented -- run scripts/embed_schema.py")
    return []


async def retrieve_few_shot_examples(
    query: str,
    agent_name: str = "nl2sql",
    tenant_id: str = "default",
    top_k: int = 3,
) -> list[dict]:
    """Retrieve similar validated NL/SQL pairs for few-shot learning."""
    logger.info("Retrieving few-shot examples", agent=agent_name, top_k=top_k)
    return []
```

backend/app/agents/registry.py

27 lines · 0.7 KB

```python
"""Agent registry — maps agent names to their classes for dynamic dispatch."""

from app.agents.base import BaseAgent

# Agent registry — populated by the RouterAgent
_registry: dict[str, type[BaseAgent]] = {}


def register_agent(name: str):
    """Decorator to register an agent class in the global registry."""

    def decorator(cls: type[BaseAgent]) -> type[BaseAgent]:
        cls.name = name
        _registry[name] = cls
        return cls

    return decorator


def get_agent(name: str) -> type[BaseAgent] | None:
    """Look up an agent class by name."""
    return _registry.get(name)


def list_agents() -> list[str]:
    """List all registered agent names."""
    return list(_registry.keys())
```

backend/app/agents/router_agent.py

101 lines · 3.6 KB

```python
"""RouterAgent — intent classification and agent dispatch.

Classifies the user's natural language query into one or more agent pipelines:
- chat_data: NL → SQL → execute → chart + narrative
- chat_knowledge: RAG over documents / knowledge base
- chat_visualize: Direct chart generation from inline data or file
- chat_explore: Schema exploration and data profiling
"""

import json

from app.agents.base import AgentResult, BaseAgent
from app.agents.registry import register_agent


@register_agent("router")
class RouterAgent(BaseAgent):
    """Routes user queries to the appropriate agent pipeline."""

    name = "router"
    description = "Intent classifier and agent dispatch router"

    INTENTS = ["chat_data", "chat_visualize", "chat_knowledge", "chat_explore", "chat_report"]

    async def execute(
        self, query: str, conversation_history: list[dict] | None = None, **kwargs
    ) -> AgentResult:
        """Classify the query intent and return a dispatch plan."""
        import time

        start = time.time()

        intent = await self._classify_intent(query)
        plan = self._build_dispatch_plan(intent, query)

        return self._timed_result(
            AgentResult(
                agent_name=self.name,
                success=True,
                output={"intent": intent, "dispatch_plan": plan},
            ),
            start,
        )

    async def _classify_intent(self, query: str) -> str:
        """Classify query intent using a fast LLM call."""
        from langchain_anthropic import ChatAnthropic

        from app.core.config import settings

        llm = ChatAnthropic(
            model=settings.LLM_FAST_MODEL,
            temperature=0,
            max_tokens=256,
            api_key=settings.ANTHROPIC_API_KEY,
        )

        prompt = f"""Classify this user query into exactly one intent category:
- chat_data: asking for data from a database (SQL query needed)
- chat_visualize: asking to create a chart or visualization
- chat_knowledge: asking about concepts, documentation, or knowledge
- chat_explore: asking to explore or understand a data schema
- chat_report: asking to generate a full report with charts and narrative

Query: {query}

Return ONLY a JSON object with the key "intent"."""

        response = await llm.ainvoke(prompt)
        try:
            result = json.loads(response.content)
            return result.get("intent", "chat_data")
        except json.JSONDecodeError:
            return "chat_data"  # Default fallback

    def _build_dispatch_plan(self, intent: str, query: str) -> list[dict]:
        """Build an ordered list of agent calls based on intent."""
        plans = {
            "chat_data": [
                {"agent": "nl2sql", "input": {"query": query}},
                {"agent": "validation", "input": {}},
                {"agent": "chart_gen", "input": {}},
                {"agent": "narrative", "input": {}},
            ],
            "chat_visualize": [
                {"agent": "chart_gen", "input": {"query": query}},
            ],
            "chat_report": [
                {"agent": "nl2sql", "input": {"query": query}},
                {"agent": "validation", "input": {}},
                {"agent": "chart_gen", "input": {}},
                {"agent": "narrative", "input": {}},
            ],
            "chat_knowledge": [
                {"agent": "narrative", "input": {"query": query, "mode": "rag"}},
            ],
            "chat_explore": [
                {"agent": "nl2sql", "input": {"query": query, "mode": "explore"}},
            ],
        }
```

```
        return plans.get(intent, [{"agent": "nl2sql", "input": {"query": query}}])
```

backend/app/agents/validation/__init__.py

1 lines · 0.1 KB

```
1| """Validation agent — SQL safety gate, deterministic checks only."""
```

backend/app/agents/validation/chart_validator.py

737 lines · 24.0 KB

```python
"""Chart hallucination detection and correction.

Deterministic validation of LLM-generated chart specifications against
the underlying data. Catches common LLM hallucination modes:

1. **Field hallucination**: spec references columns that don't exist in data
2. **Type mismatch**: chart type requires encodings that the data can't satisfy
   (e.g., Pie Chart with quantitative x-axis, Line Chart with no temporal data)
3. **Data range hallucination**: axis values imply impossibilities
   (e.g., negative revenue, dates in the future, percentages > 100%)
4. **Aggregate mismatch**: chart implies aggregation that doesn't match data
   (e.g., bar chart shows 5 bars but data has 3 categories)
5. **Empty/null data**: chart spec doesn't handle missing values
6. **Structurally invalid**: malformed Vega-Lite spec that would fail to render

Correction strategies:
    - Field mismatch → rewrite encoding field names from available columns
    - Type mismatch → downgrade to a compatible chart type
    - Range issues → adjust axis bounds to data min/max
    - Structural issues → rebuild with safe defaults

Design principle: The validator should NEVER reject a chart outright unless
the data is completely unplottable. Prefer correction over rejection.
"""

from __future__ import annotations

from dataclasses import dataclass, field
from enum import StrEnum

from app.core.logging import logger

# ---------------------------------------------------------------------------
# Chart type constraints
# ---------------------------------------------------------------------------


class ChartType(StrEnum):
    """Flint-supported chart types with encoding requirements."""

    BAR = "Bar Chart"
    LINE = "Line Chart"
    PIE = "Pie Chart"
    SCATTER = "Scatter Chart"
    AREA = "Area Chart"
    HEATMAP = "Heatmap"
    TABLE = "Table"


# Each chart type requires certain semantic types for x/y encodings
CHART_TYPE_REQUIREMENTS: dict[str, dict] = {
    "Bar Chart": {
        "x_needs": ["Category", "Temporal"],
        "y_needs": ["Quantity"],
        "x_needs_single": True,  # one category per bar
        "max_categories": 30,  # readability limit
    },
    "Line Chart": {
        "x_needs": ["Temporal", "Category"],
        "y_needs": ["Quantity"],
        "prefers_temporal_x": True,
    },
    "Pie Chart": {
        "x_needs": ["Category"],
        "y_needs": ["Quantity"],
        "max_categories": 10,  # slices become unreadable
        "x_needs_single": True,
    },
    "Scatter Chart": {
        "x_needs": ["Quantity"],
        "y_needs": ["Quantity"],
        "min_points": 5,
    },
    "Area Chart": {
        "x_needs": ["Temporal", "Category"],
        "y_needs": ["Quantity"],
    },
    "Heatmap": {
        "x_needs": ["Category"],
        "y_needs": ["Category"],
        "color_needs": ["Quantity"],
    },
    "Table": {
        # Tables accept anything
    },
}


# ---------------------------------------------------------------------------
# Data types
# ---------------------------------------------------------------------------


@dataclass
class ValidationIssue:
    """A single hallucination finding."""

    code: str
    severity: str  # "error", "warning", "info"
    field: str | None = None
```

```
    message: str = ""
    fix_applied: str | None = None


@dataclass
class ChartValidationResult:
    """Full validation output for a chart spec."""

    is_valid: bool
    issues: list[ValidationIssue] = field(default_factory=list)
    corrected_spec: dict | None = None
    suggested_chart_type: str | None = None

    @property
    def has_errors(self) -> bool:
        return any(i.severity == "error" for i in self.issues)

    @property
    def warnings(self) -> list[str]:
        return [i.message for i in self.issues if i.severity == "warning"]

    @property
    def fix_summary(self) -> list[str]:
        return [f"{i.code}: {i.fix_applied}" for i in self.issues if i.fix_applied]


# ---------------------------------------------------------------------------
# Data inference
# ---------------------------------------------------------------------------


def infer_column_types(data: list[dict]) -> dict[str, str]:
    """Infer semantic types for each column in the dataset.

    Returns mapping of column_name → "Category" | "Quantity" | "Temporal"
    """
    if not data:
        return {}

    types: dict[str, str] = {}
    for col in data[0]:
        values = [row.get(col) for row in data if row.get(col) is not None]
        if not values:
            types[col] = "Category"
            continue

        sample = values[:20]

        # Numeric check
        all_numeric = all(isinstance(v, (int, float)) for v in sample)
        if all_numeric:
            # Could be a year (Temporal) or a regular number (Quantity)
            if all(1900 <= v <= 2100 for v in sample if isinstance(v, (int, float))):
                types[col] = "Temporal"
            else:
                types[col] = "Quantity"
            continue

        # String check
        all_string = all(isinstance(v, str) for v in sample)
        if all_string:
            # Check for date patterns
            import re

            date_pat = re.compile(r"\d{4}-\d{2}-\d{2}")
            if all(date_pat.match(str(v)) for v in sample):
                types[col] = "Temporal"
            else:
                types[col] = "Category"
            continue

        types[col] = "Category"

    return types


def compute_column_stats(data: list[dict]) -> dict[str, dict]:
    """Compute basic statistics for each column."""
    stats: dict[str, dict] = {}

    for col in data[0].keys() if data else []:
        values = [row.get(col) for row in data if row.get(col) is not None]
        col_stats = {
            "count": len(values),
            "null_count": len(data) - len(values),
            "distinct_count": len(set(str(v) for v in values)),
        }

        numeric_vals = [v for v in values if isinstance(v, (int, float))]
        if numeric_vals:
            col_stats["min"] = min(numeric_vals)
            col_stats["max"] = max(numeric_vals)
            col_stats["mean"] = sum(numeric_vals) / len(numeric_vals)
            col_stats["sum"] = sum(numeric_vals)
            col_stats["is_negative"] = col_stats["min"] < 0

        stats[col] = col_stats

    return stats


# ---------------------------------------------------------------------------
# Validators
# ---------------------------------------------------------------------------

```

```python

def validate_field_existence(spec: dict, data: list[dict]) -> list[ValidationIssue]:
    """Check that all encoding fields exist in the data columns."""
    if not data:
        return [
            ValidationIssue(
                code="EMPTY_DATA",
                severity="error",
                message="No data to generate chart from",
            )
        ]

    issues: list[ValidationIssue] = []
    available_columns = set(data[0].keys())

    encodings = spec.get("encodings", {})
    for _, encoding in encodings.items():
        if not isinstance(encoding, dict):
            continue
        field = encoding.get("field", "")
        if field and field not in available_columns:
            # Try to find a similar column to fix
            fixed_field = _find_best_match(field, available_columns)
            issues.append(
                ValidationIssue(
                    code="FIELD_NOT_FOUND",
                    severity="error",
                    field=field,
                    message=f"Chart references column '{field}' which doesn't exist in data",
                    fix_applied=f"Corrected to '{fixed_field}'" if fixed_field else None,
                )
            )
            if fixed_field:
                encoding["field"] = fixed_field

    return issues


def validate_chart_type_compatibility(
    spec: dict, data: list[dict], column_types: dict[str, str]
) -> list[ValidationIssue]:
    """Check that the chart type is compatible with the data structure."""
    if not data:
        return []

    chart_type = spec.get("chartType", "Bar Chart")
    issues: list[ValidationIssue] = []

    reqs = CHART_TYPE_REQUIREMENTS.get(chart_type, {})
    if not reqs:
        return []  # Unknown chart type — pass through

    encodings = spec.get("encodings", {})
    x_field = encodings.get("x", {}).get("field", "")
    y_field = encodings.get("y", {}).get("field", "")

    # Check x-axis
    if "x_needs" in reqs and x_field:
        x_type = column_types.get(x_field, "Category")
        if x_type not in reqs["x_needs"]:
            issues.append(
                ValidationIssue(
                    code="X_AXIS_TYPE_MISMATCH",
                    severity="error",
                    field=x_field,
                    message=(
                        f"Chart type '{chart_type}' needs x-axis of type "
                        f"{reqs['x_needs']} but column '{x_field}' is '{x_type}'"
                    ),
                )
            )

    # Check y-axis
    if "y_needs" in reqs and y_field:
        y_type = column_types.get(y_field, "Category")
        if y_type not in reqs["y_needs"]:
            issues.append(
                ValidationIssue(
                    code="Y_AXIS_TYPE_MISMATCH",
                    severity="error",
                    field=y_field,
                    message=(
                        f"Chart type '{chart_type}' needs y-axis of type "
                        f"{reqs['y_needs']} but column '{y_field}' is '{y_type}'"
                    ),
                )
            )

    # Pie chart: check category count
    if chart_type == "Pie Chart":
        max_cats = reqs.get("max_categories", 10)
        distinct = len(set(str(row.get(x_field)) for row in data if x_field))
        if distinct > max_cats:
            issues.append(
                ValidationIssue(
                    code="PIE_TOO_MANY_SLICES",
                    severity="warning",
                    field=x_field,
                    message=(
                        f"Pie chart has {distinct} categories (max recommended: {max_cats}). "
                        f"Consider using a Bar Chart instead."
                    ),
                )
            )

```

```python
    # Scatter: check point count
    if chart_type == "Scatter Chart":
        min_points = reqs.get("min_points", 5)
        non_null = sum(1 for r in data if r.get(x_field) is not None and r.get(y_field) is not None)
        if non_null < min_points:
            issues.append(
                ValidationIssue(
                    code="SCATTER_TOO_FEW_POINTS",
                    severity="error",
                    message=(
                        f"Scatter chart has {non_null} valid points "
                        f"(minimum recommended: {min_points})"
                    ),
                )
            )

    # Line chart: prefer temporal x
    if chart_type == "Line Chart" and reqs.get("prefers_temporal_x") and x_field:
        x_type = column_types.get(x_field, "Category")
        if x_type != "Temporal":
            issues.append(
                ValidationIssue(
                    code="LINE_NON_TEMPORAL_X",
                    severity="warning",
                    field=x_field,
                    message=(
                        f"Line Chart x-axis '{x_field}' is '{x_type}', not Temporal. "
                        f"Line order may appear random."
                    ),
                )
            )

    return issues


def validate_data_ranges(
    spec: dict, data: list[dict], column_stats: dict[str, dict]
) -> list[ValidationIssue]:
    """Check that chart axis values are within reasonable bounds."""
    if not data:
        return []

    issues: list[ValidationIssue] = []

    encodings = spec.get("encodings", {})
    for channel in ("x", "y"):
        encoding = encodings.get(channel, {})
        if not isinstance(encoding, dict):
            continue
        field = encoding.get("field", "")
        if not field or field not in column_stats:
            continue

        stats = column_stats[field]

        # Check for impossible values in Quantity fields
        if "min" in stats and "max" in stats:
            col_min, col_max = stats["min"], stats["max"]

            # Revenue/cost should not be negative in business contexts
            field_lower = field.lower()
            if (
                any(
                    keyword in field_lower
                    for keyword in ("revenue", "amount", "price", "sales", "value")
                )
                and col_min < 0
            ):
                issues.append(
                    ValidationIssue(
                        code="NEGATIVE_VALUE",
                        severity="warning",
                        field=field,
                        message=f"Column '{field}' contains negative values (min={col_min}) but appears to be a business metric",
                    )
                )

            # Check spec axis bounds against data
            scale = encoding.get("scale", {})
            domain_min = scale.get("domainMin")
            domain_max = scale.get("domainMax")

            if domain_min is not None and domain_min < col_min:
                issues.append(
                    ValidationIssue(
                        code="AXIS_BOUND_TOO_LOW",
                        severity="warning",
                        field=field,
                        message=f"Axis minimum {domain_min} is below data minimum {col_min}",
                        fix_applied=f"Axis min adjusted to {col_min}",
                    )
                )
                scale["domainMin"] = col_min

            if domain_max is not None and domain_max > col_max * 1.5 and col_max > 0:
                issues.append(
                    ValidationIssue(
                        code="AXIS_BOUND_TOO_HIGH",
                        severity="warning",
                        field=field,
                        message=f"Axis maximum {domain_max} far exceeds data range (max={col_max})",
                    )
                )

    return issues
```

```python


def validate_spec_structure(spec: dict) -> list[ValidationIssue]:
    """Check that the chart spec has all required fields for rendering."""
    issues: list[ValidationIssue] = []

    required_keys = ("chartType", "data")
    for key in required_keys:
        if key not in spec:
            issues.append(
                ValidationIssue(
                    code="MISSING_REQUIRED_KEY",
                    severity="error",
                    field=key,
                    message=f"Chart spec is missing required key '{key}'",
                )
            )

    # Check data presence
    data_block = spec.get("data", {})
    if isinstance(data_block, dict) and not data_block.get("values"):
        issues.append(
            ValidationIssue(
                code="EMPTY_DATA_VALUES",
                severity="error",
                message="Chart data.values is empty — nothing to render",
            )
        )

    # Check encodings have field names
    encodings = spec.get("encodings", {})
    if not encodings:
        issues.append(
            ValidationIssue(
                code="NO_ENCODINGS",
                severity="warning",
                message="Chart has no encodings — will render as empty canvas",
            )
        )

    return issues


def validate_null_handling(spec: dict, data: list[dict]) -> list[ValidationIssue]:
    """Check that the chart handles null/missing values appropriately."""
    if not data:
        return []

    issues: list[ValidationIssue] = []

    encodings = spec.get("encodings", {})
    for _, encoding in encodings.items():
        if not isinstance(encoding, dict):
            continue
        field = encoding.get("field", "")
        if not field:
            continue

        null_count = sum(1 for row in data if row.get(field) is None)
        if null_count > 0:
            null_pct = (null_count / len(data)) * 100
            if null_pct > 20:
                issues.append(
                    ValidationIssue(
                        code="HIGH_NULL_RATIO",
                        severity="warning",
                        field=field,
                        message=(
                            f"Column '{field}' has {null_count} null values "
                            f"({null_pct:.0f}% of data). Chart may be misleading."
                        ),
                    )
                )

    return issues


def validate_data_size(spec: dict, data: list[dict]) -> list[ValidationIssue]:
    """Check that the data size is appropriate for the chart type."""
    issues: list[ValidationIssue] = []

    chart_type = spec.get("chartType", "Bar Chart")
    row_count = len(data)

    if row_count > 10000 and chart_type not in ("Table", "Heatmap"):
        issues.append(
            ValidationIssue(
                code="DATA_TOO_LARGE",
                severity="warning",
                message=(
                    f"Chart data has {row_count} rows. Rendering performance "
                    f"may degrade. Consider limiting or aggregating."
                ),
            )
        )

    if row_count == 0:
        issues.append(
            ValidationIssue(
                code="NO_DATA",
                severity="error",
                message="Data is empty — cannot generate chart",
            )
        )

```

```python
    return issues


# ----------------------------------------------------------------------------
# Correction engine
# ----------------------------------------------------------------------------


def suggest_chart_type_fix(
    chart_type: str,
    column_types: dict[str, str],
    data: list[dict],
    has_errors: bool,
) -> str | None:
    """If the requested chart type has incompatibilities, suggest a fallback."""
    if not has_errors:
        return None

    categories = [k for k, v in column_types.items() if v == "Category"]
    quantities = [k for k, v in column_types.items() if v == "Quantity"]
    temporals = [k for k, v in column_types.items() if v == "Temporal"]

    # Fallback chain
    if chart_type == "Pie Chart" and len(data) > 10:
        return "Bar Chart"

    if chart_type == "Scatter Chart" and len(quantities) < 2:
        if temporals and quantities:
            return "Line Chart"
        if categories and quantities:
            return "Bar Chart"

    if chart_type == "Line Chart" and not temporals and categories and quantities:
        return "Bar Chart"

    if chart_type == "Heatmap" and not quantities and categories:
        return "Table"

    # Universal fallback
    if not categories and not quantities:
        return "Table"

    return "Bar Chart"


# ----------------------------------------------------------------------------
# Main validator
# ----------------------------------------------------------------------------


def validate_chart_spec(
    spec: dict,
    data: list[dict],
    chart_type_override: str | None = None,
    auto_correct: bool = True,
) -> ChartValidationResult:
    """Validate a chart spec against data and correct hallucinations.

    Args:
        spec: The generated chart specification (ChartAssemblyInput format)
        data: The underlying SQL result data
        chart_type_override: Force a specific chart type (bypasses LLM's choice)
        auto_correct: If True, attempt to fix issues rather than just flagging them

    Returns:
        ChartValidationResult with issues, warnings, and optionally corrected spec
    """
    all_issues: list[ValidationIssue] = []

    if not data:
        return ChartValidationResult(
            is_valid=False,
            issues=[
                ValidationIssue(
                    code="NO_DATA",
                    severity="error",
                    message="No data provided for chart validation",
                )
            ],
            corrected_spec=spec,
        )

    # Override chart type if requested
    if chart_type_override:
        original_type = spec.get("chartType", "")
        spec["chartType"] = chart_type_override
        if original_type != chart_type_override:
            all_issues.append(
                ValidationIssue(
                    code="CHART_TYPE_OVERRIDDEN",
                    severity="info",
                    field="chartType",
                    message=f"Chart type overridden from '{original_type}' to '{chart_type_override}'",
                )
            )
            chart_type = chart_type_override
        else:
            chart_type = original_type
    else:
        chart_type = spec.get("chartType", "Bar Chart")

    # Compute column metadata
    column_types = infer_column_types(data)
    column_stats = compute_column_stats(data)

```

```
    # Run all validators
    all_issues.extend(validate_spec_structure(spec))
    all_issues.extend(validate_data_size(spec, data))

    if spec.get("encodings"):
        all_issues.extend(validate_field_existence(spec, data))
        all_issues.extend(validate_chart_type_compatibility(spec, data, column_types))
        all_issues.extend(validate_data_ranges(spec, data, column_stats))
        all_issues.extend(validate_null_handling(spec, data))

    # Determine validity
    has_errors = any(i.severity == "error" for i in all_issues)
    is_valid = not has_errors

    # Attempt correction if auto_correct is enabled and there are errors
    corrected_spec = spec
    suggested_type = None

    if not is_valid and auto_correct:
        # Re-validate to apply in-place fixes (fields may have been corrected)
        if any(i.code == "FIELD_NOT_FOUND" for i in all_issues):
            all_issues.extend(validate_field_existence(spec, data))

        # Suggest chart type correction
        suggested_type = suggest_chart_type_fix(
            chart_type,
            column_types,
            data,
            has_errors=True,
        )
        if suggested_type and suggested_type != chart_type:
            corrected_spec = dict(spec)  # shallow copy
            corrected_spec["chartType"] = suggested_type
            all_issues.append(
                ValidationIssue(
                    code="CHART_TYPE_CORRECTED",
                    severity="warning",
                    field="chartType",
                    message=f"Chart type corrected from '{chart_type}' to '{suggested_type}' due to data incompatibility",
                    fix_applied=f"Changed to {suggested_type}",
                )
            )

        # Re-check validity after corrections
        has_errors_after = any(
            i.severity == "error" and i.code not in ("FIELD_NOT_FOUND",) for i in all_issues
        )
        is_valid = not has_errors_after

    # Log validation summary
    error_codes = [i.code for i in all_issues if i.severity == "error"]
    warning_codes = [i.code for i in all_issues if i.severity == "warning"]
    logger.info(
        "Chart validation complete",
        is_valid=is_valid,
        chart_type=chart_type,
        errors=error_codes,
        warnings=warning_codes,
        corrections=len([i for i in all_issues if i.fix_applied]),
    )

    return ChartValidationResult(
        is_valid=is_valid,
        issues=all_issues,
        corrected_spec=corrected_spec if corrected_spec is not spec else None,
        suggested_chart_type=suggested_type,
    )


# ---------------------------------------------------------------------------
# Helpers
# ---------------------------------------------------------------------------


def _find_best_match(field: str, available: set[str]) -> str | None:
    """Find the best matching column name using fuzzy matching."""
    field_lower = field.lower().strip()

    # Exact match (case-insensitive)
    for col in available:
        if col.lower() == field_lower:
            return col

    # Contains match
    for col in available:
        if field_lower in col.lower() or col.lower() in field_lower:
            return col

    # Levenshtein-like: find closest by shared prefix length
    best = None
    best_score = 0
    for col in available:
        score = _common_prefix_len(field_lower, col.lower())
        if score > best_score:
            best_score = score
            best = col

    if best_score >= 3 and best:
        return best

    return None


def _common_prefix_len(a: str, b: str) -> int:
    """Length of common prefix between two strings."""
```

```
    n = 0
    for ca, cb in zip(a, b, strict=False):
        if ca == cb:
            n += 1
        else:
            break
    return n
```

backend/app/agents/validation/validation_agent.py

346 lines · 13.9 KB

```python
"""ValidationAgent — SQL safety gate before execution.

This agent runs deterministically (no LLM call) on every generated SQL query
before it's executed against a data source. It checks:

1. Destructive pattern detection (DROP, DELETE, TRUNCATE, etc.)
2. Read-only enforcement (only SELECT statements)
3. Dry-run EXPLAIN plan for cost estimation
4. Row count limits (>1M rows requires user confirmation)
5. Statement-timeout policy reporting (the actual `SET LOCAL statement_timeout`
   is enforced by the connector at execution time — see PostgreSQLConnector)
6. Multi-statement detection (prevents `;` injection attacks)

The agent is a *safety gate*: it returns the validated SQL plus metadata. It
deliberately does NOT mutate the SQL to inject `SET LOCAL` — that is the
connector's responsibility (the single point of execution), so the same
timeout/read-only enforcement applies to every code path that hits the
database, not just the ones that remembered to call the validator.

IMPORTANT: This file is in the DO NOT MODIFY (Protected) list in CLAUDE.md.
Any relaxation of safety rules requires explicit approval and security review.
"""

import re
import time

from app.agents.base import AgentResult, BaseAgent
from app.agents.registry import register_agent
from app.core.logging import logger


@register_agent("validation")
class ValidationAgent(BaseAgent):
    """Deterministic SQL safety validator — no LLM call.

    Every generated SQL MUST pass through this agent before execution.
    The agent is intentionally conservative: when in doubt, reject.

    Attributes:
        MAX_ROW_ESTIMATE: Queries estimated to scan more than this many rows
            require explicit user confirmation (not rejection).
    """

    name = "validation"
    description = "SQL safety gate — destructive pattern detection, EXPLAIN dry-run, row limits"

    MAX_ROW_ESTIMATE: int = 1_000_000
    STATEMENT_TIMEOUT: str = "30s"

    # Patterns that are ALWAYS rejected (no exceptions)
    DESTRUCTIVE_PATTERNS: list[tuple[str, str]] = [
        # (regex pattern, description)
        (r"\bDROP\s+TABLE\b", "DROP TABLE"),
        (r"\bDROP\s+DATABASE\b", "DROP DATABASE"),
        (r"\bDROP\s+SCHEMA\b", "DROP SCHEMA"),
        (r"\bDROP\s+INDEX\b", "DROP INDEX"),
        (r"\bDROP\s+VIEW\b", "DROP VIEW"),
        (r"\bDROP\s+FUNCTION\b", "DROP FUNCTION"),
        (r"\bDROP\s+PROCEDURE\b", "DROP PROCEDURE"),
        (r"\bDELETE\s+FROM\b", "DELETE FROM"),
        (r"\bTRUNCATE\s+(TABLE\s+)?", "TRUNCATE"),
        (r"\bALTER\s+TABLE\b.*\bDROP\b", "ALTER TABLE DROP"),
        (r"\bALTER\s+TABLE\b.*\bRENAME\b", "ALTER TABLE RENAME"),
        (r"\bINSERT\s+INTO\b", "INSERT INTO"),
        (r"\bCREATE\s+USER\b", "CREATE USER"),
        (r"\bCREATE\s+ROLE\b", "CREATE ROLE"),
        (r"\bGRANT\b", "GRANT"),
        (r"\bREVOKE\b", "REVOKE"),
        (r"\bCOPY\b.*\bFROM\b", "COPY FROM (data export)"),
    ]

    # Patterns that trigger warnings but not rejection
    WARNING_PATTERNS: list[tuple[str, str]] = [
        (r"\bUPDATE\s+\w+\s+SET\b", "UPDATE — only allowed via explicit service methods"),
        (r"\bINSERT\s", "INSERT — only allowed via explicit service methods"),
        (r"\bCREATE\s+TABLE\b", "CREATE TABLE — requires admin privileges"),
        (r"\bCREATE\s+INDEX\b", "CREATE INDEX — requires admin privileges"),
    ]

    async def execute(
        self,
        sql: str | None = None,
        tenant_id: str = "default",
        user_roles: list[str] | None = None,
        connector=None,
        **kwargs,
    ) -> AgentResult:
        """Validate a SQL query for safety.

        Args:
            sql: The generated SQL query to validate.
            tenant_id: Tenant identifier.
            user_roles: User's roles for permission checks.
            connector: Optional read connector with an ``explain(sql)`` method.
                When provided (and the SQL is otherwise valid), an EXPLAIN
                dry-run produces a real row estimate for the >MAX_ROW_ESTIMATE
                confirmation check. Runs under the connector's RLS tenant GUC.
                Fail-open: EXPLAIN problems never block a query.

        Returns:
```

```
            AgentResult with:
                - success=True if SQL is safe to execute
                - warnings: list of advisory warnings
                - errors: list of blocking errors (if any, SQL is rejected)
                - output: validation details (explain_plan, row_estimate, etc.)
        """
        start = time.time()

        if not sql:
            return self._timed_result(
                AgentResult(
                    agent_name=self.name,
                    success=False,
                    output={},
                    errors=["No SQL provided for validation"],
                ),
                start,
            )

        errors: list[str] = []
        warnings: list[str] = []

        # 1. Multi-statement injection check
        multi_stmt_warnings = self._check_multi_statement(sql)
        if multi_stmt_warnings:
            errors.extend(multi_stmt_warnings)

        # 2. Destructive pattern detection
        destructive_errors, destructive_warnings = self._check_destructive_patterns(sql)
        errors.extend(destructive_errors)
        warnings.extend(destructive_warnings)

        # 3. Read-only enforcement (must be SELECT)
        if not self._is_select_only(sql):
            errors.append(
                "Query is not a SELECT statement. All data access through the "
                "query engine must be read-only. Write operations require "
                "explicit service method authorization."
            )

        # 4. Statement timeout policy — enforced by the connector at execution
        # time (SET LOCAL statement_timeout). Reported here as metadata only;
        # the validator must not mutate the SQL, otherwise the connector's own
        # read-only/SELECT gate rejects the wrapped string.

        # 5. Basic sanity checks
        if not self._has_required_clauses(sql):
            warnings.append("Query may be missing FROM or WHERE clause")

        # 5b. Role-based restrictions (Phase 15). Roles gate specific query
        # shapes; a denied shape is a WARNING (the connector enforces hard
        # limits regardless) unless the role lacks query rights entirely.
        role_errors, role_warnings = self._check_role_restrictions(sql, user_roles or [])
        errors.extend(role_errors)
        warnings.extend(role_warnings)

        # Build validation output
        valid = len(errors) == 0
        explain_plan = None
        row_estimate = None

        # 6. EXPLAIN dry-run (only if query is otherwise valid). The
        # connector's explain() is itself fail-open; this belt-and-braces
        # guard keeps the agent safe against any connector implementation.
        if valid and connector is not None:
            try:
                explain_result = await connector.explain(sql)
                row_estimate = int(explain_result.get("estimated_rows", 0) or 0)
                explain_plan = explain_result.get("plan_text")
            except Exception as e:
                logger.warning("EXPLAIN dry-run failed (non-blocking): %s", e)
                explain_plan = None
                row_estimate = None

        # 7. Row count check (>1M requires user confirmation)
        if row_estimate and row_estimate > self.MAX_ROW_ESTIMATE:
            warnings.append(
                f"Query estimated to scan ~{row_estimate:,} rows (limit: {self.MAX_ROW_ESTIMATE:,}). "
                "This requires explicit user confirmation before execution."
            )

        logger.info(
            "Validation complete",
            valid=valid,
            errors=len(errors),
            warnings=len(warnings),
            row_estimate=row_estimate,
        )

        return self._timed_result(
            AgentResult(
                agent_name=self.name,
                success=valid,
                output={
                    # Clean, unmutated SQL — the connector enforces timeout +
                    # read-only via SET LOCAL at execution time.
                    "validated_sql": sql,
                    "statement_timeout": self.STATEMENT_TIMEOUT,
                    "explain_plan": explain_plan,
                    "row_estimate": row_estimate,
                    "requires_confirmation": bool(
                        row_estimate and row_estimate > self.MAX_ROW_ESTIMATE
                    ),
                },
                warnings=warnings,
```

```
                errors=errors,
            ),
            start,
        )

    def _check_multi_statement(self, sql: str) -> list[str]:
        """Detect multi-statement SQL (possible injection)."""
        errors = []

        # Remove string literals to avoid false positives on semicolons in strings
        cleaned = self._remove_string_literals(sql)

        # Count statements (naive: split on `;` outside strings)
        statements = [s.strip() for s in cleaned.split(";") if s.strip()]
        if len(statements) > 1:
            errors.append(
                f"Multi-statement query detected ({len(statements)} statements). "
                "Only single SELECT statements are allowed through the query engine. "
                "Multiple statements indicate a potential SQL injection attack."
            )

        return errors

    def _check_destructive_patterns(self, sql: str) -> tuple[list[str], list[str]]:
        """Check SQL for destructive patterns.

        Returns:
            Tuple of (errors, warnings) — errors block execution, warnings are advisory.
        """
        sql_upper = sql.upper()
        errors = []
        warnings = []

        for pattern, description in self.DESTRUCTIVE_PATTERNS:
            if re.search(pattern, sql_upper):
                errors.append(
                    f"Destructive operation detected: {description}. "
                    "All queries through the query engine are read-only."
                )

        for pattern, description in self.WARNING_PATTERNS:
            if re.search(pattern, sql_upper):
                warnings.append(
                    f"Potentially unsafe operation: {description}. "
                    "This query will be flagged for review."
                )

        return errors, warnings

    def _is_select_only(self, sql: str) -> bool:
        """Check that the SQL is a SELECT statement (or WITH/CTE)."""
        sql_stripped = sql.strip().upper()
        # Strip leading comments and whitespace
        while sql_stripped.startswith(("--", "/*", "(")):
            if sql_stripped.startswith("--"):
                sql_stripped = (
                    sql_stripped.split("\n", 1)[-1].strip() if "\n" in sql_stripped else ""
                )
            elif sql_stripped.startswith("/*"):
                end = sql_stripped.find("*/")
                sql_stripped = sql_stripped[end + 2 :].strip() if end != -1 else ""
            elif sql_stripped.startswith("("):
                # CTE or subquery — scan for SELECT after leading parens
                depth = 0
                for i, c in enumerate(sql_stripped):
                    if c == "(":
                        depth += 1
                    elif c == ")":
                        depth -= 1
                        if depth == 0:
                            sql_stripped = sql_stripped[i + 1 :].strip()
                            break
                else:
                    break

        return sql_stripped.startswith(
            ("SELECT", "WITH", "EXPLAIN", "EXPLAIN ANALYZE", "SHOW", "DESCRIBE")
        )

    # Roles that constrain what SQL shapes a user may run (Phase 15).
    # The connector's read-only gate and RLS remain the hard enforcement —
    # this is a policy layer on top.
    QUERY_ROLES = ("admin", "analyst", "user", "viewer")

    # The 'viewer' role: denormalized single-table lookups only.
    VIEWER_ROLE_RESTRICTIONS: list[tuple[str, str]] = [
        (r"\bJOIN\b", "JOINs (viewer role)"),
        (r"\bUNION\b", "UNION (viewer role)"),
        (r"\bWITH\b", "CTEs (viewer role)"),
    ]

    def _check_role_restrictions(
        self, sql: str, user_roles: list[str]
    ) -> tuple[list[str], list[str]]:
        """Role-based query-shape restrictions.

        - Empty/missing roles: unrestricted (tests, internal calls) — the
          JWT-authenticated API path always provides roles.
        - 'viewer': JOIN/UNION/CTE shapes are rejected (single-table
          lookups only).
        """
        errors: list[str] = []
        warnings: list[str] = []
        sql_upper = sql.upper()

```

```
        if user_roles and not any(role in user_roles for role in self.QUERY_ROLES):
            errors.append(
                f"User roles {user_roles} are not authorized to run queries "
                f"(requires one of {list(self.QUERY_ROLES)})"
            )
            return errors, warnings

        if "viewer" in user_roles:
            for pattern, description in self.VIEWER_ROLE_RESTRICTIONS:
                if re.search(pattern, sql_upper):
                    errors.append(f"Query uses {description}, not permitted for the viewer role")
                    break

        return errors, warnings

    def _has_required_clauses(self, sql: str) -> bool:
        """Basic sanity check: SELECT should have FROM or be a simple expression."""
        sql_upper = sql.upper()
        # SELECT 1, SELECT now(), etc. are valid
        if re.search(r"\bFROM\b", sql_upper):
            return True
        # Simple functions without FROM are fine
        if re.search(
            r"SELECT\s+(NOW\(\)|CURRENT_DATE|CURRENT_TIMESTAMP|VERSION\(\))",
            sql_upper,
        ):
            return True
        return True  # Don't block on missing FROM — could be a simple expression

    def _remove_string_literals(self, sql: str) -> str:
        """Remove string literals to avoid false positives in pattern matching."""
        # Replace single-quoted strings with placeholder
        cleaned = re.sub(r"'[^']*'", "''", sql)
        # Replace double-quoted identifiers with placeholder
        cleaned = re.sub(r'"[^"]*"', '""', cleaned)
        return cleaned
```

backend/app/api/__init__.py

0 lines · 0.0 KB

```
1|
```

backend/app/api/v1/__init__.py

0 lines · 0.0 KB

```
1|
```

backend/app/api/v1/auth.py

109 lines · 3.5 KB

```python
"""Authentication endpoints — user login with failed-attempt throttling."""

import uuid

from fastapi import APIRouter, Depends, HTTPException, status
from pydantic import BaseModel, Field
from sqlalchemy import text
from sqlalchemy.ext.asyncio import AsyncSession

from app.core.auth import create_access_token
from app.core.cache import get_cache
from app.core.config import settings
from app.core.security import verify_password
from app.db.session import get_auth_db

router = APIRouter()


class LoginRequest(BaseModel):
    email: str = Field(min_length=3, max_length=255)
    password: str = Field(min_length=6, max_length=128)
    # Optional: disambiguate when the same email exists in multiple tenants.
    tenant_id: str | None = None


class UserOut(BaseModel):
    id: str
    email: str
    name: str
    tenant_id: str
    roles: list[str]


class LoginResponse(BaseModel):
    access_token: str
    token_type: str = "bearer"
    user: UserOut


async def _reject(request_email: str) -> HTTPException:
    """Record a failed attempt and return the standard 401."""
    await get_cache().register_failed_login(request_email)
    return HTTPException(
        status_code=status.HTTP_401_UNAUTHORIZED,
        detail={"code": "INVALID_CREDENTIALS", "message": "Invalid email or password"},
    )


@router.post("/login", response_model=LoginResponse)
async def login(request: LoginRequest, db: AsyncSession = Depends(get_auth_db)) -> LoginResponse:
    """Authenticate a user by email + password and mint a JWT access token.

    Throttled: LOGIN_MAX_FAILURES failed attempts per email within
    LOGIN_LOCKOUT_SECONDS → 429. Indistinguishable 401s for every failure
    mode (no account enumeration).
    """
    email = request.email.strip().lower()

    # Lockout check before any DB work (the genbi_auth lookup is a
    # cross-tenant read oracle — throttle brute force).
    cache = get_cache()
    if await cache.login_failures(email) >= settings.LOGIN_MAX_FAILURES:
        raise HTTPException(
            status_code=status.HTTP_429_TOO_MANY_REQUESTS,
            detail={
                "code": "TOO_MANY_ATTEMPTS",
                "message": "Too many failed attempts. Try again later.",
            },
        )

    try:
        tenant_uuid = uuid.UUID(request.tenant_id) if request.tenant_id else None
    except ValueError:
        raise await _reject(email) from None

    rows = (
        await db.execute(
            text(
                # CAST(... AS uuid), not ::uuid — SQLAlchemy text() treats
                # '::' as an escaped literal colon and breaks the bind.
                "SELECT id, tenant_id, email, hashed_password, roles "
                "FROM users "
                "WHERE email = :email "
                "AND (CAST(:tenant_id AS uuid) IS NULL OR tenant_id = CAST(:tenant_id AS uuid)) "
                "LIMIT 2"
            ),
            {"email": email, "tenant_id": tenant_uuid},
        )
    ).all()

    if len(rows) != 1 or not verify_password(request.password, rows[0].hashed_password):
        raise await _reject(email)

    user = rows[0]
    roles = list(user.roles or ["user"])
    token = create_access_token(user_id=str(user.id), tenant_id=str(user.tenant_id), roles=roles)

    await cache.clear_failed_logins(email)

    return LoginResponse(
```

```
        access_token=token,
        user=UserOut(
            id=str(user.id),
            email=user.email,
            name=user.email.split("@")[0],
            tenant_id=str(user.tenant_id),
            roles=roles,
        ),
    )
```

backend/app/api/v1/charts.py

26 lines · 0.9 KB

```
"""Chart endpoints — render charts via Flint MCP and manage chart history."""

from fastapi import APIRouter, Depends

from app.core.auth import get_current_user
from app.models.chart import ChartRenderRequest, ChartRenderResponse

router = APIRouter()


@router.post("/render", response_model=ChartRenderResponse)
async def render_chart(
    request: ChartRenderRequest,
    user: dict = Depends(get_current_user),
):
    """Render a chart from a ChartAssemblyInput spec using Flint MCP.
    Supports Vega-Lite, ECharts, and Chart.js backends. Returns PNG or SVG."""
    from app.agents.chart.flint_bridge import FlintChartBridge

    bridge = FlintChartBridge(tenant_id=user["tenant_id"])
    result = await bridge.render(
        spec=request.spec,
        backend=request.backend or "vegalite",
        output_format=request.format or "png",
    )
    return ChartRenderResponse(**result)
```

backend/app/api/v1/chat.py

84 lines · 2.7 KB

```python
"""Chat endpoint — natural language to SQL + chart + narrative pipeline."""

from fastapi import APIRouter, Depends, HTTPException, status
from pydantic import BaseModel, Field

from app.core.auth import get_current_user
from app.models.chat import ChatRequest, ChatResponse
from app.services.chat_service import ChatService

router = APIRouter()


class FeedbackRequest(BaseModel):
    """Thumbs up/down on a chat response (updates audit_log.feedback_score)."""

    session_id: str = Field(min_length=36, max_length=36)
    score: int = Field(ge=-1, le=1, description="1 = up, -1 = down, 0 = clear")


@router.post("", response_model=ChatResponse)
async def chat(
    request: ChatRequest,
    user: dict = Depends(get_current_user),
):
    """Main chat endpoint. Accepts a natural language query and returns
    SQL, chart spec, and narrative insight in a streaming or single response."""
    service = ChatService(tenant_id=user["tenant_id"])
    return await service.process_query(
        query=request.query,
        user_id=user["sub"],
        roles=user.get("roles", []),
        conversation_id=request.conversation_id,
        confirm_large_query=request.confirm_large_query,
    )


@router.post("/feedback")
async def chat_feedback(
    request: FeedbackRequest,
    user: dict = Depends(get_current_user),
):
    """Record thumbs-up/down feedback on a chat response.

    The score lands on the audit rows of the session (the SSE start event
    / sync response session id — NOT the conversation id).
    """
    from app.services.audit import record_feedback

    ok = await record_feedback(
        session_id=request.session_id,
        tenant_id=user["tenant_id"],
        score=request.score,
    )
    if not ok:
        raise HTTPException(
            status_code=status.HTTP_503_SERVICE_UNAVAILABLE,
            detail={
                "code": "FEEDBACK_UNAVAILABLE",
                "message": "Could not record feedback (no matching audit rows "
                "or the audit store is unreachable)",
            },
        )
    return {"status": "recorded", "session_id": request.session_id, "score": request.score}


@router.post("/stream")
async def chat_stream(
    request: ChatRequest,
    user: dict = Depends(get_current_user),
):
    """Streaming variant — yields SSE events as the agent pipeline progresses."""
    from fastapi.responses import StreamingResponse

    service = ChatService(tenant_id=user["tenant_id"])
    return StreamingResponse(
        service.process_query_stream(
            query=request.query,
            user_id=user["sub"],
            roles=user.get("roles", []),
            conversation_id=request.conversation_id,
            confirm_large_query=request.confirm_large_query,
        ),
        media_type="text/event-stream",
    )
```

backend/app/api/v1/conversations.py

97 lines · 2.9 KB

```python
"""Conversation endpoints — multi-turn chat history (Phase 14)."""

import uuid

from fastapi import APIRouter, Depends, HTTPException, Query, status
from pydantic import BaseModel, Field

from app.core.auth import get_current_user

router = APIRouter()


class ConversationSummary(BaseModel):
    id: str
    title: str
    created_at: str
    updated_at: str


class ConversationListResponse(BaseModel):
    conversations: list[ConversationSummary]
    count: int


class MessageOut(BaseModel):
    role: str = Field(pattern="^(user|assistant)$")
    content: str
    generated_sql: str | None = None
    created_at: str


class MessageListResponse(BaseModel):
    messages: list[MessageOut]
    count: int


@router.get("", response_model=ConversationListResponse)
async def list_conversations(
    user: dict = Depends(get_current_user),
    limit: int = Query(default=50, ge=1, le=100),
):
    """List the current user's conversations, most recently active first."""
    from app.services import conversations as conversations_service

    try:
        rows = await conversations_service.list_conversations(
            user_id=user["sub"], tenant_id=user["tenant_id"], limit=limit
        )
    except Exception as e:
        raise HTTPException(
            status_code=status.HTTP_503_SERVICE_UNAVAILABLE,
            detail={
                "code": "PERSISTENCE_UNAVAILABLE",
                "message": f"Conversation store unavailable: {type(e).__name__}",
            },
        ) from None

    return ConversationListResponse(
        conversations=[ConversationSummary(**row) for row in rows], count=len(rows)
    )


@router.get("/{conversation_id}/messages", response_model=MessageListResponse)
async def list_messages(
    conversation_id: str,
    user: dict = Depends(get_current_user),
    limit: int = Query(default=50, ge=1, le=200),
):
    """List a conversation's most recent turns (chronological order).

    RLS scopes rows to the caller's tenant; other tenants' conversation ids
    simply return zero messages.
    """
    from app.services import conversations as conversations_service

    try:
        uuid.UUID(conversation_id)
    except ValueError:
        raise HTTPException(
            status_code=status.HTTP_400_BAD_REQUEST,
            detail={"code": "INVALID_CONVERSATION", "message": "Not a valid conversation id"},
        ) from None

    try:
        rows = await conversations_service.list_messages(
            conversation_id=conversation_id, tenant_id=user["tenant_id"], limit=limit
        )
    except Exception as e:
        raise HTTPException(
            status_code=status.HTTP_503_SERVICE_UNAVAILABLE,
            detail={
                "code": "PERSISTENCE_UNAVAILABLE",
                "message": f"Conversation store unavailable: {type(e).__name__}",
            },
        ) from None

    return MessageListResponse(messages=[MessageOut(**row) for row in rows], count=len(rows))
```

backend/app/api/v1/datasources.py

64 lines · 1.8 KB

```python
"""Datasource endpoints — introspect the semantic layer's cubes."""

from fastapi import APIRouter, Depends, HTTPException, status
from pydantic import BaseModel

from app.core.auth import get_current_user

router = APIRouter()


class DatasourceSummary(BaseModel):
    name: str
    title: str
    measures: int
    dimensions: int


class DatasourceListResponse(BaseModel):
    datasources: list[DatasourceSummary]
    count: int


@router.get("", response_model=DatasourceListResponse)
async def list_datasources(
    user: dict = Depends(get_current_user),
):
    """List the semantic layer's cubes (data sources) for the current tenant."""
    from app.semantic.cube_client import get_cube_client

    try:
        meta = await get_cube_client().get_meta()
    except Exception as e:
        raise HTTPException(
            status_code=status.HTTP_503_SERVICE_UNAVAILABLE,
            detail={
                "code": "CUBE_UNAVAILABLE",
                "message": f"Semantic layer unavailable: {type(e).__name__}",
            },
        ) from None

    summaries = [
        DatasourceSummary(
            name=cube.get("name", ""),
            title=cube.get("title") or cube.get("name", ""),
            measures=len(cube.get("measures", [])),
            dimensions=len(cube.get("dimensions", [])),
        )
        for cube in meta.cubes
    ]

    return DatasourceListResponse(datasources=summaries, count=len(summaries))


@router.post("/test")
async def test_connection(
    user: dict = Depends(get_current_user),
):
    """Test a user-configured external database connection.

    Deliberately still a stub: this targets a different feature (admin-
    configured external warehouses), not the semantic layer's built-in
    Postgres source (which the readiness probe and /datasources cover).
    """
    return {"status": "not_implemented"}
```

backend/app/api/v1/health.py

58 lines · 2.0 KB

```python
"""Health check endpoints."""

from fastapi import APIRouter, Response, status
from sqlalchemy import text

from app.core.cache import get_cache
from app.core.config import settings
from app.db.session import async_session

router = APIRouter()


@router.get("/health")
async def health_check():
    """Liveness probe — returns 200 if the server is running."""
    return {"status": "ok", "version": "0.1.0"}


@router.get("/health/ready")
async def readiness_check(response: Response):
    """Readiness probe — actually pings DB and Redis.

    Returns 200 only when both dependencies are reachable; 503 otherwise, with
    per-dependency status so operators can see what's broken. This is the probe
    used by Docker healthchecks (Task 5) and ``make verify`` (Task 8) — it must
    never report "ready" when a dependency is down.

    Note on ``mcp_flint``: Flint runs as a per-render subprocess, not a
    persistent server, so there is no socket to ping. We report its configured
    state rather than fake "connected".
    """
    checks = {}

    # Database — real ping via SELECT 1 on the async engine
    db_ok = False
    try:
        async with async_session() as session:
            await session.execute(text("SELECT 1"))
        db_ok = True
    except Exception as e:
        checks["database_error"] = str(e)[:200]
    checks["database"] = "connected" if db_ok else "unreachable"

    # Redis — real ping (re-checks every call, not memoized like _ensure)
    redis_ok = False
    try:
        redis_ok = await get_cache().ping()
    except Exception as e:
        checks["redis_error"] = str(e)[:200]
    checks["redis"] = "connected" if redis_ok else "unreachable"

    # Flint — configured, not a live socket (subprocess-spawned per render)
    checks["mcp_flint"] = "configured" if settings.FLINT_MCP_BACKENDS else "unconfigured"

    ready = db_ok and redis_ok
    if not ready:
        response.status_code = status.HTTP_503_SERVICE_UNAVAILABLE
    return {"status": "ready" if ready else "degraded", **checks}
```

backend/app/api/v1/metrics.py

206 lines · 6.5 KB

```python
"""Metrics endpoints — query the semantic layer via Cube.dev."""

from fastapi import APIRouter, Depends, HTTPException, status
from pydantic import BaseModel, Field

from app.core.auth import get_current_user

router = APIRouter()


class MetricSummary(BaseModel):
    name: str
    title: str
    description: str
    metric_type: str
    cube_name: str
    measure_name: str
    dimensions: list[str]
    time_dimensions: list[str]


class MetricListResponse(BaseModel):
    metrics: list[MetricSummary]
    count: int


class TimeDimensionSpec(BaseModel):
    dimension: str
    granularity: str = Field(pattern="^(day|week|month|quarter|year|hour)$")
    date_range: list[str] | None = Field(default=None, max_length=2)


class FilterSpec(BaseModel):
    member: str
    operator: str
    values: list[str] | None = None


class MetricQueryRequest(BaseModel):
    measures: list[str] = Field(min_length=1, max_length=5)
    dimensions: list[str] | None = Field(default=None, max_length=5)
    time_dimensions: list[TimeDimensionSpec] | None = Field(default=None, max_length=3)
    filters: list[FilterSpec] | None = None
    order: list[list[str]] | None = None
    limit: int = Field(default=100, ge=1, le=1000)
    offset: int = Field(default=0, ge=0, le=100000)
    timezone: str = "UTC"


class MetricQueryResponse(BaseModel):
    data: list[dict]
    annotation: dict
    total: float | None
    query: dict
    latency_ms: float
    cached: bool


@router.get("/list", response_model=MetricListResponse)
async def list_metrics(
    user: dict = Depends(get_current_user),
):
    """List all available metrics from the semantic layer (Cube /meta)."""
    from app.semantic.cube_client import get_cube_client

    try:
        metrics = await get_cube_client().list_metrics()
    except Exception as e:
        raise HTTPException(
            status_code=status.HTTP_503_SERVICE_UNAVAILABLE,
            detail={
                "code": "CUBE_UNAVAILABLE",
                "message": f"Semantic layer unavailable: {type(e).__name__}",
            },
        ) from None

    # Skip the bare-measure-name aliases: one entry per measure, keyed by
    # the canonical `cube.measure` name.
    seen_names: set[str] = set()
    summaries: list[MetricSummary] = []
    for metric in metrics.values():
        if metric.name in seen_names:
            continue
        seen_names.add(metric.name)
        summaries.append(
            MetricSummary(
                name=metric.name,
                title=metric.title,
                description=metric.description,
                metric_type=metric.metric_type.value,
                cube_name=metric.cube_name,
                measure_name=metric.measure_name,
                dimensions=metric.dimensions,
                time_dimensions=metric.time_dimensions,
            )
        )

    return MetricListResponse(metrics=summaries, count=len(summaries))

```

```python
@router.post("/query", response_model=MetricQueryResponse)
async def query_metrics(
    request: MetricQueryRequest,
    user: dict = Depends(get_current_user),
):
    """Execute a tenant-scoped metric query against Cube.dev.

    The tenant claim from the JWT is forwarded to Cube, which selects the
    per-tenant driver pool whose connections carry the RLS GUC — so results
    are tenant-isolated at the database layer (ADR 008).
    """
    from app.core.cache import get_cache
    from app.semantic.cube_client import get_cube_client

    tenant_id = user["tenant_id"]
    cube = get_cube_client()
    cache = get_cache()

    # Validate measures against the catalog before hitting Cube.
    try:
        catalog = await cube.list_metrics()
    except Exception as e:
        raise HTTPException(
            status_code=status.HTTP_503_SERVICE_UNAVAILABLE,
            detail={
                "code": "CUBE_UNAVAILABLE",
                "message": f"Semantic layer unavailable: {type(e).__name__}",
            },
        ) from None

    unknown = [m for m in request.measures if m not in catalog]
    if unknown:
        raise HTTPException(
            status_code=status.HTTP_400_BAD_REQUEST,
            detail={
                "code": "INVALID_METRIC",
                "message": f"Unknown measure(s): {', '.join(unknown)}",
            },
        )

    # Canonical query form — used as the cache key, so every result-affecting
    # parameter must be included.
    cube_query: dict = {
        "measures": request.measures,
        "limit": request.limit,
        "offset": request.offset,
        "timezone": request.timezone,
    }
    if request.dimensions:
        cube_query["dimensions"] = request.dimensions
    if request.time_dimensions:
        cube_query["timeDimensions"] = [
            {
                "dimension": td.dimension,
                "granularity": td.granularity,
                **({"dateRange": td.date_range} if td.date_range else {}),
            }
            for td in request.time_dimensions
        ]
    if request.filters:
        cube_query["filters"] = [
            {
                "member": f.member,
                "operator": f.operator,
                **({"values": f.values} if f.values is not None else {}),
            }
            for f in request.filters
        ]
    if request.order:
        cube_query["order"] = {field: direction for field, direction in request.order}

    cached_result = await cache.get_cube_query_result(cube_query, tenant_id)
    if cached_result is not None:
        return MetricQueryResponse(cached=True, **cached_result)

    try:
        result = await cube.query(
            metrics=request.measures,
            dimensions=request.dimensions,
            time_dimensions=cube_query.get("timeDimensions"),
            filters=cube_query.get("filters"),
            order=request.order,
            limit=request.limit,
            offset=request.offset,
            timezone=request.timezone,
            tenant_id=tenant_id,
        )
    except Exception as e:
        raise HTTPException(
            status_code=status.HTTP_503_SERVICE_UNAVAILABLE,
            detail={
                "code": "CUBE_UNAVAILABLE",
                "message": f"Semantic layer query failed: {type(e).__name__}",
            },
        ) from None

    payload = {
        "data": result.data,
        "annotation": result.annotation,
        "total": result.total,
        "query": result.query,
        "latency_ms": round(result.query_latency_ms, 2),
    }
    await cache.set_cube_query_result(cube_query, tenant_id, payload)

```

```
    return MetricQueryResponse(cached=result.cached, **payload)
```

backend/app/api/v1/reports.py

150 lines · 4.4 KB

```python
"""Report endpoints — assemble multi-chart reports with narrative (Phase 16)."""

import uuid

from fastapi import APIRouter, Depends, HTTPException, Query, status
from pydantic import BaseModel, Field

from app.core.auth import get_current_user

router = APIRouter()


class ReportGenerateRequest(BaseModel):
    prompt: str = Field(min_length=1, max_length=2000)
    max_sections: int = Field(default=3, ge=2, le=4)


class ReportSectionOut(BaseModel):
    position: int
    metric_name: str
    section_title: str
    chart_spec: dict
    chart_svg: str | None = None
    data_total: float | None = None
    row_count: int = 0
    narrative: str | None = None


class ReportOut(BaseModel):
    report_id: str
    title: str
    prompt: str
    summary: str | None = None
    status: str
    created_at: str
    sections: list[ReportSectionOut]
    warnings: list[str] = []


class ReportSummaryOut(BaseModel):
    id: str
    title: str
    created_at: str
    section_count: int


class ReportListResponse(BaseModel):
    reports: list[ReportSummaryOut]
    count: int


@router.post("/generate", response_model=ReportOut)
async def generate_report(
    request: ReportGenerateRequest,
    user: dict = Depends(get_current_user),
):
    """Generate a multi-chart report from a natural language prompt.

    LLM-planned (metric selection from the semantic layer catalog) with
    deterministic execution: per-section tenant-scoped Cube query, chart
    render, and an overall narrative. Sections that return no data are
    skipped with a warning; the report persists best-effort.
    """
    from app.services.reports import ReportGenerationError, generate_report

    try:
        report = await generate_report(
            prompt=request.prompt,
            tenant_id=user["tenant_id"],
            user_id=user["sub"],
            max_sections=request.max_sections,
        )
    except ReportGenerationError as e:
        raise HTTPException(
            status_code=status.HTTP_503_SERVICE_UNAVAILABLE,
            detail={
                "code": "REPORT_GENERATION_FAILED",
                "message": f"Could not plan a report for this prompt: {e}",
            },
        ) from None
    except Exception as e:
        raise HTTPException(
            status_code=status.HTTP_503_SERVICE_UNAVAILABLE,
            detail={
                "code": "REPORT_GENERATION_FAILED",
                "message": f"Report generation failed: {type(e).__name__}",
            },
        ) from None

    return ReportOut(**report)


@router.get("", response_model=ReportListResponse)
async def list_reports(
    user: dict = Depends(get_current_user),
    limit: int = Query(default=50, ge=1, le=100),
):
    """List the current user's reports, newest first."""
    from app.services import reports as reports_service

```

```
    try:
        rows = await reports_service.list_reports(
            user_id=user["sub"], tenant_id=user["tenant_id"], limit=limit
        )
    except Exception as e:
        raise HTTPException(
            status_code=status.HTTP_503_SERVICE_UNAVAILABLE,
            detail={
                "code": "PERSISTENCE_UNAVAILABLE",
                "message": f"Report store unavailable: {type(e).__name__}",
            },
        ) from None

    return ReportListResponse(reports=[ReportSummaryOut(**row) for row in rows], count=len(rows))


@router.get("/{report_id}", response_model=ReportOut)
async def get_report(
    report_id: str,
    user: dict = Depends(get_current_user),
):
    """Retrieve a previously generated report (RLS tenant-scoped)."""
    from app.services import reports as reports_service

    try:
        uuid.UUID(report_id)
    except ValueError:
        raise HTTPException(
            status_code=status.HTTP_400_BAD_REQUEST,
            detail={"code": "INVALID_REPORT", "message": "Not a valid report id"},
        ) from None

    try:
        report = await reports_service.get_report(report_id=report_id, tenant_id=user["tenant_id"])
    except Exception as e:
        raise HTTPException(
            status_code=status.HTTP_503_SERVICE_UNAVAILABLE,
            detail={
                "code": "PERSISTENCE_UNAVAILABLE",
                "message": f"Report store unavailable: {type(e).__name__}",
            },
        ) from None

    if report is None:
        raise HTTPException(
            status_code=status.HTTP_404_NOT_FOUND,
            detail={"code": "REPORT_NOT_FOUND", "message": "No such report"},
        )

    return ReportOut(**report)
```

backend/app/api/v1/router.py

16 lines · 0.8 KB

```
"""API v1 router — aggregates all versioned route modules."""

from fastapi import APIRouter

from app.api.v1 import auth, charts, chat, conversations, datasources, health, metrics, reports

api_router = APIRouter()

api_router.include_router(auth.router, prefix="/auth", tags=["auth"])
api_router.include_router(health.router, tags=["health"])
api_router.include_router(chat.router, prefix="/chat", tags=["chat"])
api_router.include_router(conversations.router, prefix="/conversations", tags=["conversations"])
api_router.include_router(charts.router, prefix="/charts", tags=["charts"])
api_router.include_router(reports.router, prefix="/reports", tags=["reports"])
api_router.include_router(metrics.router, prefix="/metrics", tags=["metrics"])
api_router.include_router(datasources.router, prefix="/datasources", tags=["datasources"])
```

backend/app/connectors/__init__.py

0 lines · 0.0 KB

```
1|
```

backend/app/connectors/base.py

34 lines · 0.8 KB

```python
"""Database connector base and registry."""

from abc import ABC, abstractmethod


class BaseConnector(ABC):
    """Abstract base for database connectors.

    All connectors enforce read-only access. Write operations must go through
    explicit service methods with authorization checks.
    """

    name: str = "base"
    _read_only: bool = True

    @abstractmethod
    async def connect(self) -> None:
        """Establish a connection."""
        ...

    @abstractmethod
    async def execute(self, sql: str) -> list[dict]:
        """Execute a read-only SQL query."""
        ...

    @abstractmethod
    async def get_schema(self) -> dict:
        """Retrieve table and column metadata."""
        ...

    @abstractmethod
    async def disconnect(self) -> None:
        """Close the connection."""
        ...
```

backend/app/connectors/postgresql_connector.py

334 lines · 12.2 KB

```python
"""
PostgreSQL connector implementation — asyncpg-backed, read-only enforced.

This connector implements BaseConnector for PostgreSQL with:

- Read-only transaction enforcement (SET TRANSACTION READ ONLY)
- 30-second statement_timeout injection
- Parameterized query support via asyncpg
- Schema metadata retrieval (tables, columns, types, descriptions)
- Connection pooling via SQLAlchemy async engine
- Destructive SQL blocking at the driver level

Usage:
    connector = PostgreSQLConnector(connection_url="postgresql+asyncpg://...")
    await connector.connect()
    results = await connector.execute("SELECT region, SUM(revenue) FROM sales GROUP BY region")
    schema = await connector.get_schema()
    await connector.disconnect()
"""

from typing import Any

from sqlalchemy import text
from sqlalchemy.ext.asyncio import AsyncSession, async_sessionmaker, create_async_engine

from app.connectors.base import BaseConnector
from app.core.logging import logger


class PostgreSQLConnector(BaseConnector):
    """PostgreSQL database connector with read-only enforcement.

    All queries go through a connection that sets:
    - SET TRANSACTION READ ONLY (prevents any write at DB level)
    - SET statement_timeout = '30s' (caps query execution time)
    - SET application_name = 'genbi' (identifies queries in pg_stat_activity)

    The connector uses a separate async engine instance per data source
    so it can connect to the tenant's actual data warehouse, not just
    the metadata database.
    """

    name = "postgresql"
    _read_only = True

    def __init__(
        self,
        connection_url: str,
        schema: str = "public",
        tenant_id: str | None = None,
        pool_size: int = 10,
        max_overflow: int = 5,
        echo: bool = False,
    ):
        """Initialize the connector.

        Args:
            connection_url: SQLAlchemy asyncpg connection string.
            schema: Default schema for query resolution.
            tenant_id: Tenant identifier. When set, the per-transaction GUC
                ``app.current_tenant_id`` is injected so PostgreSQL row-level
                security policies (see infra/postgres/init.sql) scope every
                query to this tenant. Required for multi-tenant queries.
            pool_size: Connection pool size.
            max_overflow: Max overflow connections.
            echo: SQL echo mode for debugging.
        """
        self.connection_url = connection_url
        self.schema = schema
        self.tenant_id = tenant_id
        self._engine = None
        self._session_factory = None
        self._connected = False
        self._pool_size = pool_size
        self._max_overflow = max_overflow
        self._echo = echo

    async def connect(self) -> None:
        """Establish connection pool."""
        if self._connected:
            return

        logger.info(
            "Connecting to PostgreSQL",
            schema=self.schema,
            pool_size=self._pool_size,
        )

        self._engine = create_async_engine(
            self.connection_url,
            echo=self._echo,
            pool_size=self._pool_size,
            max_overflow=self._max_overflow,
            pool_pre_ping=True,
            pool_recycle=3600,
            connect_args={
                "server_settings": {
                    "application_name": "genbi",
                },
            },
```

```
        )

        self._session_factory = async_sessionmaker(
            self._engine,
            class_=AsyncSession,
            expire_on_commit=False,
        )

        # Test the connection
        async with self._session_factory() as session:
            await session.execute(text("SELECT 1"))

        self._connected = True
        logger.info("PostgreSQL connected successfully")

    async def execute(self, sql: str, params: dict[str, Any] | None = None) -> list[dict]:
        """Execute a read-only SQL query and return results as dicts.

        Args:
            sql: SQL query string (parameterized with :param placeholders).
            params: Query parameters for parameterized queries.

        Returns:
            List of dicts, one per row.

        Raises:
            ConnectionError: If not connected.
            ValueError: If the SQL is not a SELECT or contains destructive patterns.
        """
        if not self._connected:
            raise ConnectionError("Not connected. Call connect() first.")

        # Sanity check — must be SELECT
        sql_stripped = sql.strip().upper()
        if not sql_stripped.startswith(("SELECT", "WITH", "EXPLAIN", "SHOW", "DESCRIBE")):
            raise ValueError(
                f"Only SELECT/WITH/EXPLAIN queries are allowed through this connector. "
                f"Received: {sql[:100]}"
            )

        # Read-only enforcement + timeout — each SET must be its own execute()
        # because the asyncpg extended (prepared-statement) protocol does not
        # allow multiple semicolon-separated statements in a single round-trip.
        # Issuing them separately keeps the connector the single source of these
        # guarantees for every code path that hits the database.
        async with self._session_factory() as session:
            # Mark the transaction read-only at the DB level. Must precede any
            # data statement; emitted first so the transaction begins read-only.
            await session.execute(text("SET TRANSACTION READ ONLY"))
            await session.execute(text("SET LOCAL statement_timeout = '30s'"))
            await session.execute(text("SET LOCAL default_transaction_read_only = on"))
            # Tenant GUC — drives row-level security policies (see init.sql).
            # Parameterized via the GUC string only when a tenant is bound, so
            # RLS scopes every SELECT to this tenant.
            if self.tenant_id:
                await session.execute(
                    text("SET LOCAL app.current_tenant_id = :tid"),
                    {"tid": str(self.tenant_id)},
                )

            result = await session.execute(
                text(sql),
                params or {},
            )

            # Convert rows to dicts
            if result.returns_rows:
                columns = list(result.keys())
                rows = []
                for row in result.fetchall():
                    row_dict = {}
                    for i, col in enumerate(columns):
                        val = row[i]
                        # Convert non-serializable types
                        if hasattr(val, "isoformat"):
                            val = val.isoformat()
                        row_dict[col] = val
                    rows.append(row_dict)
                return rows

            return []

    async def execute_raw(self, sql: str) -> Any:
        """Execute SQL and return raw cursor result.

        For EXPLAIN plans, introspection queries, and other non-data queries.
        Applies the same timeout + read-only enforcement as ``execute``.
        """
        if not self._connected:
            raise ConnectionError("Not connected. Call connect() first.")

        async with self._session_factory() as session:
            await session.execute(text("SET TRANSACTION READ ONLY"))
            await session.execute(text("SET LOCAL statement_timeout = '30s'"))
            if self.tenant_id:
                await session.execute(
                    text("SET LOCAL app.current_tenant_id = :tid"),
                    {"tid": str(self.tenant_id)},
                )
            result = await session.execute(text(sql))
            return result

    async def explain(self, sql: str) -> dict:
        """Get EXPLAIN plan for a SQL query.

```

```
        Returns:
            Dict with plan_type, plan_text, estimated_cost, and estimated_rows.
        """
        try:
            result = await self.execute_raw(f"EXPLAIN (FORMAT JSON) {sql}")
            plan = result.scalar_one()
            import json

            plan_data = json.loads(plan) if isinstance(plan, str) else plan

            first_plan = (
                plan_data[0]["Plan"] if isinstance(plan_data, list) else plan_data.get("Plan", {})
            )
            return {
                "plan_type": "EXPLAIN (FORMAT JSON)",
                "plan_text": json.dumps(plan_data, indent=2),
                "estimated_cost": first_plan.get("Total Cost", 0),
                "estimated_rows": first_plan.get("Plan Rows", 0),
            }
        except Exception as e:
            logger.warning(f"EXPLAIN failed: {e}")
            return {
                "plan_type": "EXPLAIN",
                "plan_text": f"EXPLAIN failed: {e}",
                "estimated_cost": 0,
                "estimated_rows": 0,
            }

    async def get_schema(self) -> dict:
        """Retrieve table and column metadata for the configured schema.

        Returns:
            Dict with tables list, each containing name, columns, row_count, and description.
        """
        if not self._connected:
            await self.connect()

        query = """
            SELECT
                t.table_name,
                c.column_name,
                c.data_type,
                c.is_nullable,
                pg_catalog.obj_description(
                    (quote_ident(t.table_schema) || '.' || quote_ident(t.table_name))::regclass,
                    'pg_class'
                ) AS table_description,
                pg_catalog.col_description(
                    (quote_ident(t.table_schema) || '.' || quote_ident(t.table_name))::regclass,
                    c.ordinal_position
                ) AS column_description
            FROM information_schema.tables t
            JOIN information_schema.columns c
                ON t.table_schema = c.table_schema
                AND t.table_name = c.table_name
            WHERE t.table_schema = :schema
                AND t.table_type = 'BASE TABLE'
            ORDER BY t.table_name, c.ordinal_position
        """

        rows = await self.execute(query, {"schema": self.schema})

        # Group by table
        tables: dict[str, dict] = {}
        for row in rows:
            table_name = row["table_name"]
            if table_name not in tables:
                tables[table_name] = {
                    "table_name": table_name,
                    "schema": self.schema,
                    "description": row.get("table_description") or "",
                    "columns": [],
                }
            tables[table_name]["columns"].append(
                {
                    "column_name": row["column_name"],
                    "data_type": row["data_type"],
                    "is_nullable": row["is_nullable"] == "YES",
                    "description": row.get("column_description") or "",
                }
            )

        # Get approximate row counts
        for table_name in tables:
            try:
                count_result = await self.execute(
                    f"SELECT COUNT(*) AS cnt FROM {self.schema}.{table_name}"
                )
                tables[table_name]["row_count"] = count_result[0]["cnt"] if count_result else 0
            except Exception:
                tables[table_name]["row_count"] = -1  # unknown

        return {
            "schema": self.schema,
            "tables": list(tables.values()),
            "table_count": len(tables),
        }

    async def get_sample_rows(
        self, table_name: str, schema: str | None = None, limit: int = 5
    ) -> list[dict]:
        """Get a sample of rows from a table for schema context building.

        Args:
            table_name: Name of the table.
```

```
            schema: Schema override (uses configured schema if None).
            limit: Number of sample rows.

        Returns:
            List of dicts representing sample rows.
        """
        schema = schema or self.schema
        # Use parameterized identifiers
        query = f'SELECT * FROM "{schema}"."{table_name}" LIMIT {limit}'
        return await self.execute(query)

    async def disconnect(self) -> None:
        """Close connection pool."""
        if self._engine:
            await self._engine.dispose()
            self._connected = False
            logger.info("PostgreSQL disconnected")

    async def __aenter__(self):
        await self.connect()
        return self

    async def __aexit__(self, exc_type, exc_val, exc_tb):
        await self.disconnect()
```

backend/app/connectors/registry.py

32 lines · 0.8 KB

```python
"""Connector registry — maps data source types to connector classes."""

from app.connectors.base import BaseConnector

_registry: dict[str, type[BaseConnector]] = {}


def register_connector(name: str):
    """Decorator to register a connector class."""

    def decorator(cls: type[BaseConnector]) -> type[BaseConnector]:
        cls.name = name
        _registry[name] = cls
        return cls

    return decorator


def get_connector(name: str) -> type[BaseConnector] | None:
    """Look up a connector class by name."""
    return _registry.get(name)


def list_connectors() -> list[str]:
    """List all registered connector names."""
    return list(_registry.keys())


# Register built-in connectors
from app.connectors.postgresql_connector import PostgreSQLConnector  # noqa: E402

register_connector("postgresql")(PostgreSQLConnector)
```

backend/app/core/__init__.py

5 lines · 0.2 KB

```
1 """Core package -- config, auth, logging, middleware, masking, caching."""
2
3 from app.core.cache import CacheService, CacheTTL, get_cache
4
5 __all__ = ["CacheService", "CacheTTL", "get_cache"]
```

backend/app/core/auth.py

55 lines · 1.8 KB

```python
"""Authentication and authorization — JWT + RBAC."""

from datetime import UTC, datetime, timedelta

from fastapi import Depends, HTTPException, status
from fastapi.security import HTTPAuthorizationCredentials, HTTPBearer
from jose import JWTError, jwt

from app.core.config import settings

security = HTTPBearer()


def create_access_token(
    user_id: str,
    tenant_id: str,
    roles: list[str] | None = None,
    expires_minutes: int | None = None,
) -> str:
    """Create a JWT access token."""
    expires_delta = timedelta(minutes=expires_minutes or settings.JWT_EXPIRE_MINUTES)
    expire = datetime.now(UTC) + expires_delta

    payload = {
        "sub": user_id,
        "tenant_id": tenant_id,
        "roles": roles or ["user"],
        "exp": expire,
        "iat": datetime.now(UTC),
    }
    return jwt.encode(payload, settings.JWT_SECRET_KEY, algorithm=settings.JWT_ALGORITHM)


def decode_token(token: str) -> dict:
    """Decode and validate a JWT token. Raises JWTError on failure."""
    return jwt.decode(token, settings.JWT_SECRET_KEY, algorithms=[settings.JWT_ALGORITHM])


async def get_current_user(
    credentials: HTTPAuthorizationCredentials = Depends(security),
) -> dict:
    """FastAPI dependency: extract and validate the current user from JWT."""
    try:
        payload = decode_token(credentials.credentials)
        if payload.get("sub") is None:
            raise HTTPException(
                status_code=status.HTTP_401_UNAUTHORIZED,
                detail={"code": "INVALID_TOKEN", "message": "Token missing subject claim"},
            )
        return payload
    except JWTError:
        raise HTTPException(
            status_code=status.HTTP_401_UNAUTHORIZED,
            detail={"code": "INVALID_TOKEN", "message": "Token is expired or invalid"},
        ) from None
```

backend/app/core/cache.py

727 lines · 24.7 KB

```python
"""Redis caching layer — schema, metrics, and query results.

Centralized caching strategy for the GenBI platform. Every cacheable entity
routes through this module with consistent TTL policies and cache invalidation.

Cache tiers:
    Level 1: In-memory LRU (sub-millisecond, per-process, 1000-entry max)
    Level 2: Redis (millisecond, shared across workers, persistent)

What's cached:
    - Schema embeddings (TTL: 24h — only changes on nightly sync)
    - Metric definitions (TTL: 5min — Cube.dev /meta)
    - Query results (TTL: 5min — read-heavy analytics)
    - LLM response cache (TTL: 1h — same query/schema = same SQL)
    - Rate limit counters (TTL: window-based)
    - Session state (TTL: 30min idle)

Cache keys are namespaced: genbi:{tenant_id}:{entity_type}:{hash}

Usage:
    cache = get_cache()

    # Schema
    schema = await cache.get_schema_context(query, tenant_id)
    await cache.set_schema_context(query, tenant_id, schema)

    # Metrics
    metrics = await cache.get_metric_definitions(tenant_id)
    await cache.set_metric_definitions(tenant_id, metrics)

    # Query results
    result = await cache.get_query_result(sql_hash, tenant_id)
    await cache.set_query_result(sql_hash, tenant_id, rows)

    # Invalidate
    await cache.invalidate_tenant(tenant_id)
    await cache.invalidate_schema(tenant_id)
"""

import contextlib
import hashlib
import json
import time
from collections import OrderedDict
from dataclasses import dataclass
from typing import Any

from app.core.config import settings
from app.core.logging import logger

# ---------------------------------------------------------------------------
# TTL configuration
# ---------------------------------------------------------------------------


class CacheTTL:
    """Time-to-live constants for each cache category."""

    SCHEMA_EMBEDDINGS = 86_400  # 24 hours
    METRIC_DEFINITIONS = 300  # 5 minutes
    QUERY_RESULTS = 300  # 5 minutes
    LLM_RESPONSE = 3_600  # 1 hour
    RATE_LIMIT = 60  # 1 minute window
    SESSION_STATE = 1_800  # 30 minutes
    CHART_SPEC = 3_600  # 1 hour


# ---------------------------------------------------------------------------
# Cache namespace helpers
# ---------------------------------------------------------------------------


def _key(tenant_id: str, entity: str, identifier: str) -> str:
    """Build a namespaced cache key."""
    return f"genbi:{tenant_id}:{entity}:{identifier}"


def _hash_query(query: str) -> str:
    """Hash a query string for deterministic cache keys."""
    return hashlib.sha256(query.encode("utf-8")).hexdigest()[:16]


def _hash_sql(sql: str) -> str:
    """Hash a SQL string — normalize before hashing."""
    normalized = " ".join(sql.lower().split())
    return hashlib.sha256(normalized.encode("utf-8")).hexdigest()[:16]


def _hash_data(data: list[dict]) -> str:
    """Hash query result data for cache identity."""
    canonical = json.dumps(data[:10], sort_keys=True, default=str)
    return hashlib.sha256(canonical.encode("utf-8")).hexdigest()[:16]


# ---------------------------------------------------------------------------
# In-memory LRU cache (Level 1)
# ---------------------------------------------------------------------------


class LRUCache:
```

```
    """Simple LRU cache with max size and TTL support.

    Used as the Level 1 in-memory cache before hitting Redis.
    Thread-safe via Python's GIL for the async single-thread model.
    """

    def __init__(self, max_size: int = 1000):
        self._max_size = max_size
        self._store: OrderedDict[str, tuple[Any, float]] = OrderedDict()

    def get(self, key: str, ttl_seconds: int = 300) -> Any | None:
        """Get a value if it exists and hasn't expired."""
        if key not in self._store:
            return None

        value, stored_at = self._store[key]
        age = time.time() - stored_at

        if age > ttl_seconds:
            del self._store[key]
            return None

        # Move to end (most recently used)
        self._store.move_to_end(key)
        return value

    def set(self, key: str, value: Any) -> None:
        """Set a value, evicting the least recently used if at capacity."""
        if key in self._store:
            self._store.move_to_end(key)

        self._store[key] = (value, time.time())

        # Evict LRU if over capacity
        while len(self._store) > self._max_size:
            self._store.popitem(last=False)

    def invalidate(self, prefix: str | None = None) -> int:
        """Invalidate all entries, or those matching a prefix."""
        count = 0
        if prefix is None:
            count = len(self._store)
            self._store.clear()
        else:
            keys_to_delete = [k for k in self._store if k.startswith(prefix)]
            for k in keys_to_delete:
                del self._store[k]
            count = len(keys_to_delete)
        return count

    def stats(self) -> dict:
        """Return cache stats."""
        return {
            "size": len(self._store),
            "max_size": self._max_size,
            "utilization_pct": round(len(self._store) / self._max_size * 100, 1),
        }


# ---------------------------------------------------------------------------
# Redis-based cache (Level 2)
# ---------------------------------------------------------------------------


class RedisCache:
    """Redis-backed cache — shared across workers, persistent.

    Uses connection pooling and graceful degradation: if Redis is down,
    operations are no-ops rather than failures.
    """

    def __init__(self):
        self._redis = None
        self._available = False
        self._init_attempted = False

    async def ping(self) -> bool:
        """Health probe — True if Redis is reachable right now.

        Distinct from ``_ensure``: ``_ensure`` is lazy and memoizes a failure for
        the process lifetime (so a transient Redis blip at startup never
        recovers). ``ping`` re-checks on every call, making it suitable for
        readiness probes.
        """
        try:
            import redis.asyncio as aioredis

            client = aioredis.from_url(
                settings.REDIS_URL,
                socket_connect_timeout=2,
            )
            try:
                await client.ping()
                return True
            finally:
                await client.aclose()
        except Exception as e:
            logger.debug(f"Redis ping failed: {e}")
            return False

    async def _ensure(self) -> bool:
        """Lazy-init Redis connection. Returns True if available."""
        if self._init_attempted:
            return self._available

```

```
        self._init_attempted = True
        try:
            import redis.asyncio as aioredis

            self._redis = aioredis.from_url(
                settings.REDIS_URL,
                decode_responses=False,
                socket_connect_timeout=3,
                socket_keepalive=True,
                health_check_interval=30,
                max_connections=20,
            )
            await self._redis.ping()
            self._available = True
            logger.info("Redis cache connected")
        except ImportError:
            logger.warning("redis package not installed — Redis cache disabled")
            self._available = False
        except Exception as e:
            logger.warning(f"Redis unavailable, using in-memory cache only — {e}")
            self._available = False

        return self._available

    async def get(self, key: str) -> Any | None:
        """Get a raw value from Redis."""
        if not await self._ensure():
            return None

        try:
            raw = await self._redis.get(key)
            if raw is None:
                return None

            data = json.loads(raw if isinstance(raw, (str, bytes)) else raw.decode("utf-8"))
            return data
        except Exception:
            return None

    async def set(self, key: str, value: Any, ttl: int = 300) -> None:
        """Set a value with TTL in Redis."""
        if not await self._ensure():
            return

        try:
            await self._redis.setex(key, ttl, json.dumps(value, default=str))
        except Exception as e:
            logger.debug(f"Redis set failed (non-fatal): {e}")

    async def delete(self, key: str) -> None:
        """Delete a key from Redis."""
        if not await self._ensure():
            return

        with contextlib.suppress(Exception):
            await self._redis.delete(key)

    async def delete_pattern(self, pattern: str) -> int:
        """Delete all keys matching a pattern. Returns count of deleted keys."""
        if not await self._ensure():
            return 0

        try:
            cursor = 0
            count = 0
            while True:
                cursor, keys = await self._redis.scan(cursor, match=pattern, count=100)
                if keys:
                    await self._redis.delete(*keys)
                    count += len(keys)
                if cursor == 0:
                    break
            return count
        except Exception:
            return 0

    async def exists(self, key: str) -> bool:
        """Check if a key exists in Redis."""
        if not await self._ensure():
            return False

        try:
            return bool(await self._redis.exists(key))
        except Exception:
            return False

    async def close(self) -> None:
        """Close the Redis connection."""
        if self._redis:
            with contextlib.suppress(Exception):
                await self._redis.close()
            self._redis = None
            self._available = False


# ---------------------------------------------------------------------------
# Unified Cache Service
# ---------------------------------------------------------------------------


@dataclass
class CacheStats:
    """Aggregate cache statistics."""

    l1_size: int = 0
```

```
    l1_max: int = 1000
    l1_hits: int = 0
    l1_misses: int = 0
    l2_hits: int = 0
    l2_misses: int = 0
    writes: int = 0
    invalidations: int = 0

    @property
    def l1_hit_rate(self) -> float:
        total = self.l1_hits + self.l1_misses
        return self.l1_hits / total * 100 if total > 0 else 0.0

    @property
    def l2_hit_rate(self) -> float:
        total = self.l2_hits + self.l2_misses
        return self.l2_hits / total * 100 if total > 0 else 0.0


class CacheService:
    """Two-level cache: in-memory LRU (L1) + Redis (L2).

    Read path: L1 → L2 → miss → compute → L2 → L1
    Write path: compute → L2 + L1 simultaneously

    All operations are async-safe. Cache misses are returned as None —
    the caller is responsible for computing and re-caching the value.
    """

    def __init__(self, l1_max_size: int = 1000):
        self._l1 = LRUCache(max_size=l1_max_size)
        self._l2 = RedisCache()
        self._stats = CacheStats()

    # ----------------------------------------------------------------
    # Schema context cache
    # ----------------------------------------------------------------

    async def get_schema_context(self, query: str, tenant_id: str) -> list[dict] | None:
        """Get cached schema context for a query."""
        query_hash = _hash_query(query)
        key = _key(tenant_id, "schema", query_hash)

        # L1
        value = self._l1.get(key, CacheTTL.SCHEMA_EMBEDDINGS)
        if value is not None:
            self._stats.l1_hits += 1
            return value

        # L2
        value = await self._l2.get(key)
        if value is not None:
            self._stats.l2_hits += 1
            self._l1.set(key, value)  # Promote to L1
            return value

        self._stats.l2_misses += 1
        return None

    async def set_schema_context(self, query: str, tenant_id: str, context: list[dict]) -> None:
        """Cache schema context for a query."""
        query_hash = _hash_query(query)
        key = _key(tenant_id, "schema", query_hash)

        self._l1.set(key, context)
        await self._l2.set(key, context, ttl=CacheTTL.SCHEMA_EMBEDDINGS)
        self._stats.writes += 1

    # ----------------------------------------------------------------
    # Few-shot examples cache
    # ----------------------------------------------------------------

    async def get_few_shot_examples(self, query: str, tenant_id: str) -> list[dict] | None:
        """Get cached few-shot examples for a query."""
        query_hash = _hash_query(query)
        key = _key(tenant_id, "fewshot", query_hash)

        value = self._l1.get(key, CacheTTL.LLM_RESPONSE)
        if value is not None:
            self._stats.l1_hits += 1
            return value

        value = await self._l2.get(key)
        if value is not None:
            self._stats.l2_hits += 1
            self._l1.set(key, value)
            return value

        self._stats.l2_misses += 1
        return None

    async def set_few_shot_examples(self, query: str, tenant_id: str, examples: list[dict]) -> None:
        """Cache few-shot examples for a query."""
        query_hash = _hash_query(query)
        key = _key(tenant_id, "fewshot", query_hash)

        self._l1.set(key, examples)
        await self._l2.set(key, examples, ttl=CacheTTL.LLM_RESPONSE)
        self._stats.writes += 1

    # ----------------------------------------------------------------
    # Metric definitions cache
    # ----------------------------------------------------------------

    async def get_metric_definitions(self, tenant_id: str) -> dict | None:
```

```
        """Get cached metric definitions from Cube.dev."""
        key = _key(tenant_id, "metrics", "all")

        # L1
        value = self._l1.get(key, CacheTTL.METRIC_DEFINITIONS)
        if value is not None:
            self._stats.l1_hits += 1
            return value

        # L2
        value = await self._l2.get(key)
        if value is not None:
            self._stats.l2_hits += 1
            self._l1.set(key, value)
            return value

        self._stats.l2_misses += 1
        return None

    async def set_metric_definitions(self, tenant_id: str, metrics: dict) -> None:
        """Cache metric definitions."""
        key = _key(tenant_id, "metrics", "all")
        self._l1.set(key, metrics)
        await self._l2.set(key, metrics, ttl=CacheTTL.METRIC_DEFINITIONS)
        self._stats.writes += 1

    # ----------------------------------------------------------------
    # Query result cache
    # ----------------------------------------------------------------

    async def get_query_result(self, sql: str, tenant_id: str) -> list[dict] | None:
        """Get cached query result."""
        sql_hash = _hash_sql(sql)
        key = _key(tenant_id, "query", sql_hash)

        # L1
        value = self._l1.get(key, CacheTTL.QUERY_RESULTS)
        if value is not None:
            self._stats.l1_hits += 1
            logger.debug("Query result cache hit (L1)", key=key[:40])
            return value

        # L2
        value = await self._l2.get(key)
        if value is not None:
            self._stats.l2_hits += 1
            self._l1.set(key, value)
            logger.debug("Query result cache hit (L2)", key=key[:40])
            return value

        self._stats.l2_misses += 1
        return None

    async def set_query_result(self, sql: str, tenant_id: str, rows: list[dict]) -> None:
        """Cache a query result."""
        sql_hash = _hash_sql(sql)
        key = _key(tenant_id, "query", sql_hash)

        self._l1.set(key, rows)
        await self._l2.set(key, rows, ttl=CacheTTL.QUERY_RESULTS)
        self._stats.writes += 1

    # ----------------------------------------------------------------

    async def get_cube_query_result(self, cube_query: dict, tenant_id: str) -> dict | None:
        """Get a cached Cube (semantic-layer) query result.

        Keyed by tenant + a hash of the canonical query JSON — same query,
        same tenant, same answer within the QUERY_RESULTS TTL.
        """
        key = _key(tenant_id, "cube_query", _hash_sql(json.dumps(cube_query, sort_keys=True)))

        value = self._l1.get(key, CacheTTL.QUERY_RESULTS)
        if value is not None:
            self._stats.l1_hits += 1
            return value

        value = await self._l2.get(key)
        if value is not None:
            self._stats.l2_hits += 1
            self._l1.set(key, value)
            return value

        self._stats.l2_misses += 1
        return None

    async def set_cube_query_result(self, cube_query: dict, tenant_id: str, result: dict) -> None:
        """Cache a Cube (semantic-layer) query result."""
        key = _key(tenant_id, "cube_query", _hash_sql(json.dumps(cube_query, sort_keys=True)))

        self._l1.set(key, result)
        await self._l2.set(key, result, ttl=CacheTTL.QUERY_RESULTS)
        self._stats.writes += 1

    # ----------------------------------------------------------------
    # LLM response cache
    # ----------------------------------------------------------------

    async def get_llm_response(self, prompt_hash: str, tenant_id: str) -> dict | None:
        """Get a cached LLM response.

        LLM responses are cached by prompt hash + tenant_id.
        Same question + same schema = same SQL (deterministic at temp=0).
        """
        key = _key(tenant_id, "llm", prompt_hash)
```

```python

        # L2 only for LLM responses (too large for L1 with many variants)
        value = await self._l2.get(key)
        if value is not None:
            self._stats.l2_hits += 1
            return value

        self._stats.l2_misses += 1
        return None

    async def set_llm_response(self, prompt_hash: str, tenant_id: str, response: dict) -> None:
        """Cache an LLM response."""
        key = _key(tenant_id, "llm", prompt_hash)
        await self._l2.set(key, response, ttl=CacheTTL.LLM_RESPONSE)
        self._stats.writes += 1

    # ----------------------------------------------------------------
    # Chart spec cache
    # ----------------------------------------------------------------

    async def get_chart_spec(self, data_hash: str, tenant_id: str) -> dict | None:
        """Get a cached chart spec for a data signature."""
        key = _key(tenant_id, "chart", data_hash)

        value = self._l1.get(key, CacheTTL.CHART_SPEC)
        if value is not None:
            self._stats.l1_hits += 1
            return value

        value = await self._l2.get(key)
        if value is not None:
            self._stats.l2_hits += 1
            self._l1.set(key, value)
            return value

        self._stats.l2_misses += 1
        return None

    async def set_chart_spec(self, data_hash: str, tenant_id: str, spec: dict) -> None:
        """Cache a chart spec."""
        key = _key(tenant_id, "chart", data_hash)
        self._l1.set(key, spec)
        await self._l2.set(key, spec, ttl=CacheTTL.CHART_SPEC)
        self._stats.writes += 1

    # ----------------------------------------------------------------
    # Session state
    # ----------------------------------------------------------------

    async def get_session(self, session_id: str) -> dict | None:
        """Get cached session state."""
        key = f"genbi:session:{session_id}"
        return await self._l2.get(key)

    async def set_session(
        self, session_id: str, state: dict, ttl: int = CacheTTL.SESSION_STATE
    ) -> None:
        """Cache session state."""
        key = f"genbi:session:{session_id}"
        await self._l2.set(key, state, ttl=ttl)

    # ----------------------------------------------------------------
    # Rate limiting
    # ----------------------------------------------------------------

    async def check_rate_limit(
        self, user_id: str, max_requests: int = 30, window_seconds: int = 60
    ) -> tuple[bool, int]:
        """Rate-limit check using Redis INCR + TTL.

        Returns:
            (allowed, remaining_requests)
        """
        if not await self._l2._ensure():
            return True, max_requests  # Allow all if Redis is down

        key = f"genbi:ratelimit:{user_id}"
        try:
            current = await self._l2._redis.incr(key)
            if current == 1:
                await self._l2._redis.expire(key, window_seconds)

            remaining = max(0, max_requests - current)
            return current <= max_requests, remaining
        except Exception:
            return True, max_requests

    # ----------------------------------------------------------------

    async def register_failed_login(self, email: str) -> int:
        """Count a failed login attempt for an email (Redis INCR + TTL).

        Returns the failure count in the current lockout window. Fails open
        (returns 0) when Redis is unavailable — availability over lockout.
        """
        from app.core.config import settings

        if not await self._l2._ensure():
            return 0

        key = f"genbi:loginfail:{email.strip().lower()}"
        with contextlib.suppress(Exception):
            current = await self._l2._redis.incr(key)
            if current == 1:
                await self._l2._redis.expire(key, settings.LOGIN_LOCKOUT_SECONDS)
```

```
            return int(current)
        return 0

    async def login_failures(self, email: str) -> int:
        """Current failed-login count for an email (0 when Redis is down)."""
        if not await self._l2._ensure():
            return 0

        with contextlib.suppress(Exception):
            value = await self._l2._redis.get(f"genbi:loginfail:{email.strip().lower()}")
            return int(value or 0)
        return 0

    async def clear_failed_logins(self, email: str) -> None:
        """Reset the failure counter after a successful login."""
        if not await self._l2._ensure():
            return

        with contextlib.suppress(Exception):
            await self._l2._redis.delete(f"genbi:loginfail:{email.strip().lower()}")

    # ----------------------------------------------------------------
    # Invalidation
    # ----------------------------------------------------------------

    async def invalidate_tenant(self, tenant_id: str) -> int:
        """Invalidate all cached data for a tenant."""
        pattern = f"genbi:{tenant_id}:*"
        l1_count = self._l1.invalidate(prefix=f"genbi:{tenant_id}:")
        l2_count = await self._l2.delete_pattern(pattern)
        total = l1_count + l2_count
        logger.info(f"Invalidated {total} cache entries for tenant {tenant_id}")
        self._stats.invalidations += total
        return total

    async def invalidate_schema(self, tenant_id: str) -> int:
        """Invalidate schema-related caches (after nightly embed sync)."""
        schema_pattern = f"genbi:{tenant_id}:schema:*"
        l1_count = self._l1.invalidate(prefix=f"genbi:{tenant_id}:schema:")
        l2_count = await self._l2.delete_pattern(schema_pattern)
        total = l1_count + l2_count
        logger.info(f"Invalidated {total} schema cache entries")
        self._stats.invalidations += total
        return total

    async def invalidate_metrics(self, tenant_id: str) -> int:
        """Invalidate metric definition caches."""
        metrics_pattern = f"genbi:{tenant_id}:metrics:*"
        l1_count = self._l1.invalidate(prefix=f"genbi:{tenant_id}:metrics:")
        l2_count = await self._l2.delete_pattern(metrics_pattern)
        total = l1_count + l2_count
        self._stats.invalidations += total
        return total

    # ----------------------------------------------------------------
    # Stats
    # ----------------------------------------------------------------

    async def ping(self) -> bool:
        """Health probe — True if Redis (L2) is reachable right now.

        Used by the /health/ready readiness check. Does not probe L1 (in-memory),
        which is always available if the process is running.
        """
        return await self._l2.ping()

    def get_stats(self) -> dict:
        """Get cache performance metrics."""
        return {
            "l1": self._l1.stats(),
            "l1_hits": self._stats.l1_hits,
            "l1_misses": self._stats.l1_misses,
            "l1_hit_rate": f"{self._stats.l1_hit_rate:.1f}%",
            "l2_hits": self._stats.l2_hits,
            "l2_misses": self._stats.l2_misses,
            "l2_hit_rate": f"{self._stats.l2_hit_rate:.1f}%",
            "writes": self._stats.writes,
            "invalidations": self._stats.invalidations,
        }

    # ----------------------------------------------------------------
    # Cleanup
    # ----------------------------------------------------------------

    async def close(self) -> None:
        """Close Redis connection."""
        await self._l2.close()


# ---------------------------------------------------------------------------
# Singleton
# ---------------------------------------------------------------------------

_cache_service: CacheService | None = None


def get_cache() -> CacheService:
    """Get or create the singleton CacheService."""
    global _cache_service
    if _cache_service is None:
        _cache_service = CacheService(l1_max_size=1000)
    return _cache_service
```

backend/app/core/config.py

138 lines · 4.9 KB

```python
"""Application configuration via pydantic-settings.

All settings load from environment variables with sensible defaults.
Secrets are NEVER hardcoded — always use env vars or .env file.
"""

from pydantic import model_validator
from pydantic_settings import BaseSettings, SettingsConfigDict

# Secret values that must not ship to non-development environments. Any match
# here in production/staging fails fast at Settings() construction time rather
# than silently signing JWTs with a publicly known key.
_INSECURE_SECRET_VALUES = frozenset(
    {
        "change-me",
        "change-me-in-production",
        "change-me-in-production-use-openssl-rand-64",
        "",
    }
)


class Settings(BaseSettings):
    """GenBI application settings."""

    model_config = SettingsConfigDict(
        env_file=".env",
        env_file_encoding="utf-8",
        extra="ignore",
    )

    # --- App ---
    APP_ENV: str = "development"
    DEBUG: bool = True
    LOG_LEVEL: str = "DEBUG"
    # Stored as a raw string (comma-separated) so pydantic-settings can parse it
    # from .env files without requiring JSON list syntax. Use `cors_origins_list`
    # to get the parsed list in code (e.g. CORSMiddleware).
    CORS_ORIGINS: str = "http://localhost:3000"

    @property
    def cors_origins_list(self) -> list[str]:
        """Parse CORS_ORIGINS into a list, accepting JSON or comma-separated."""
        v = self.CORS_ORIGINS.strip()
        if v.startswith("["):
            import json

            try:
                return json.loads(v)
            except json.JSONDecodeError:
                pass
        return [origin.strip() for origin in v.split(",") if origin.strip()]

    # --- LLM ---
    ANTHROPIC_API_KEY: str = ""
    LLM_REASONING_MODEL: str = "claude-opus-4"
    LLM_FAST_MODEL: str = "claude-haiku-4"

    # --- Embeddings (schema/few-shot retrieval; OpenAI text-embedding-3-small
    # produces exactly the 1536 dims the VECTOR(1536) columns expect) ---
    OPENAI_API_KEY: str = ""
    EMBEDDING_MODEL: str = "text-embedding-3-small"
    EMBEDDING_DIMS: int = 1536

    # --- Database ---
    # Runtime role (non-superuser; RLS-bound). Owner credentials stay in
    # DATABASE_URL_SYNC for Alembic and admin scripts only.
    DATABASE_URL: str = "postgresql+asyncpg://genbi_app:genbi_app@localhost:5432/genbi"
    DATABASE_URL_SYNC: str = "postgresql://genbi:genbi@localhost:5432/genbi"
    # Login endpoint role (reads `users` across tenants). Blank = derive from
    # DATABASE_URL by swapping credentials, so host/database follow the main
    # URL in every environment.
    DATABASE_URL_AUTH: str = ""

    @property
    def database_url_auth(self) -> str:
        """Auth-engine URL: explicit override, or DATABASE_URL with genbi_auth creds."""
        if self.DATABASE_URL_AUTH:
            return self.DATABASE_URL_AUTH
        from sqlalchemy.engine import make_url

        return str(make_url(self.DATABASE_URL).set(username="genbi_auth", password="genbi_auth"))

    # --- Redis ---
    REDIS_URL: str = "redis://localhost:6379/0"

    # --- Semantic Layer ---
    CUBE_API_URL: str = "http://localhost:4000/cubejs-api/v1"
    CUBE_API_SECRET: str = ""

    # --- Observability ---
    LANGFUSE_SECRET_KEY: str = ""
    LANGFUSE_PUBLIC_KEY: str = ""
    LANGFUSE_HOST: str = "https://cloud.langfuse.com"

    # --- Auth ---
    JWT_SECRET_KEY: str = "change-me"
    JWT_ALGORITHM: str = "HS256"
    JWT_EXPIRE_MINUTES: int = 60
    # Login throttling: failed-attempt counter per normalized email,
```

```
    # Redis-backed, fails open (no lockout) when Redis is unavailable.
    LOGIN_MAX_FAILURES: int = 5
    LOGIN_LOCKOUT_SECONDS: int = 900

    # --- Tenant ---
    TENANT_ENCRYPTION_KEY: str = ""

    # --- Flint Chart MCP ---
    FLINT_MCP_BACKENDS: str = "vegalite,echarts,chartjs"
    FLINT_MCP_DATA_ROOTS: str = "/tmp/genbi-charts"

    # --- Prompts ---
    # Directory holding versioned prompt templates (loaded by llm_client.load_prompt).
    # Defaults to the repo's .claude/prompts/ on host runs; override for containers
    # where the working dir differs. The Dockerfile bakes the prompts into /app/.claude.
    PROMPT_DIR: str = ""

    @model_validator(mode="after")
    def _enforce_production_secrets(self):
        """Fail fast if production is launched with an insecure secret.

        A known-default or empty JWT secret in any non-development environment
        would let anyone forge tokens. Better to refuse to boot than to run
        with a publicly-known signing key.
        """
        if (
            self.APP_ENV not in ("development", "test")
            and self.JWT_SECRET_KEY in _INSECURE_SECRET_VALUES
        ):
            raise ValueError(
                "JWT_SECRET_KEY must be set to a strong random value in "
                f"APP_ENV={self.APP_ENV} (generate one with `openssl rand -hex 32`). "
                "Aborting boot to avoid running with a default/known secret."
            )
        return self


settings = Settings()
```

backend/app/core/embeddings.py

46 lines · 1.7 KB

```python
"""Embedding provider — vector search support for schema/few-shot retrieval.

Uses OpenAI embeddings: `text-embedding-3-small` produces exactly the 1536
dimensions the VECTOR(1536) columns (schema_embeddings, agent_examples)
expect. Anthropic has no public embeddings API — the original
`claude-embeddings-*` call in scripts/embed_schema.py never worked.
"""

from app.core.config import settings
from app.core.logging import logger


async def embed_text(text: str) -> list[float]:
    """Embed a text into a VECTOR(1536)-compatible float list.

    Raises RuntimeError when no API key is configured or the API call fails
    — callers (schema retrieval) fail open and treat this as no context.
    """
    if not settings.OPENAI_API_KEY:
        raise RuntimeError("OPENAI_API_KEY is not configured — embeddings unavailable")

    from openai import AsyncOpenAI

    client = AsyncOpenAI(api_key=settings.OPENAI_API_KEY)
    try:
        response = await client.embeddings.create(
            model=settings.EMBEDDING_MODEL,
            input=text,
            dimensions=settings.EMBEDDING_DIMS,
        )
    except Exception as e:
        logger.warning("Embedding API call failed: %s", type(e).__name__)
        raise RuntimeError("embedding API call failed") from e

    embedding = response.data[0].embedding
    if len(embedding) != settings.EMBEDDING_DIMS:
        raise RuntimeError(
            f"embedding model returned {len(embedding)} dims, "
            f"expected {settings.EMBEDDING_DIMS} (VECTOR column is fixed-size)"
        )
    return embedding


def vector_literal(embedding: list[float]) -> str:
    """Render an embedding as a Postgres vector literal ('[0.1,0.2,...]')."""
    return "[" + ",".join(str(v) for v in embedding) + "]"
```

backend/app/core/llm_client.py

379 lines · 12.9 KB

```python
"""Centralized LLM client — single wrapper around Anthropic API.

All GenBI agents MUST route LLM calls through this module. Never call
langchain_anthropic or the Anthropic SDK directly in agent code.

This client handles:
- Timeout + retry (3 attempts, exponential backoff with jitter)
- Token budget enforcement (warn on approach, reject on exceed)
- Structured output parsing (JSON extraction with fallback)
- Latency tracking and audit logging
- Model selection from settings (LLM_REASONING_MODEL / LLM_FAST_MODEL)
"""

import hashlib
import json
import re
import time
import uuid
from collections.abc import Awaitable, Callable
from dataclasses import dataclass

from langchain_anthropic import ChatAnthropic
from langchain_core.messages import BaseMessage, HumanMessage

from app.core.config import settings
from app.core.logging import logger

# ---------------------------------------------------------------------------
# Types
# ---------------------------------------------------------------------------


@dataclass
class LLMCallResult:
    """Result of a single LLM invocation."""

    content: str
    model_name: str
    input_tokens: int = 0
    output_tokens: int = 0
    latency_ms: float = 0
    attempt: int = 1
    parsed: dict | None = None  # Parsed JSON if structured output was requested


@dataclass
class LLMCallOptions:
    """Options for an LLM call."""

    temperature: float = 0.0
    max_tokens: int = 4096
    thinking: bool = False
    response_format: str | None = None  # "json" for structured output
    timeout_seconds: int = 60
    max_retries: int = 3
    token_budget: int = 100_000


# ---------------------------------------------------------------------------
# Client
# ---------------------------------------------------------------------------


class LLMClient:
    """Centralized Anthropic LLM client with retry, audit, and budget control.

    Usage:
        client = LLMClient()

        # Fast call (Haiku) for classification
        result = await client.invoke(
            messages="Classify this: ...",
            system="You are a classifier.",
            use_reasoning=False,
        )

        # Reasoning call (Opus) for SQL generation
        result = await client.invoke(
            messages="Generate SQL for: ...",
            system=load_prompt("nl2sql-system"),
            use_reasoning=True,
            response_format="json",
        )
    """

    def __init__(self):
        self._audit_callback: Callable[[dict], Awaitable[None]] | None = None

    def set_audit_callback(self, callback: Callable[[dict], Awaitable[None]]) -> None:
        """Register a callback for audit logging (called after every LLM invocation)."""
        self._audit_callback = callback

    async def invoke(
        self,
        messages: str | list[BaseMessage],
        system: str | None = None,
        use_reasoning: bool = False,
        options: LLMCallOptions | None = None,
        user_id: str | None = None,
        tenant_id: str | None = None,
```

```
        session_id: str | None = None,
        generated_sql: str | None = None,
    ) -> LLMCallResult:
        """Invoke the LLM with retry, audit, and structured output.

        Args:
            messages: Single string (converted to HumanMessage) or list of LangChain messages.
            system: System prompt string.
            use_reasoning: If True, uses the reasoning model (Opus) with thinking mode.
            options: Fine-grained call options.
            user_id: User identifier for audit logging.
            tenant_id: Tenant identifier for audit logging.
            session_id: Session identifier for audit logging.
            generated_sql: If this call is part of a SQL pipeline, the SQL it produced.

        Returns:
            LLMCallResult with content, token counts, latency, and parsed JSON if applicable.
        """
        opts = options or LLMCallOptions()
        if use_reasoning:
            opts.thinking = True
            opts.temperature = 0.0

        model_name = settings.LLM_REASONING_MODEL if use_reasoning else settings.LLM_FAST_MODEL
        start_time = time.time()
        last_error: Exception | None = None

        for attempt in range(1, opts.max_retries + 1):
            try:
                llm = self._build_llm(model_name, opts, attempt)
                langchain_messages = self._normalize_messages(messages)

                logger.debug(
                    "LLM call",
                    model=model_name,
                    attempt=attempt,
                    messages=len(langchain_messages),
                    thinking=opts.thinking,
                )

                response = await llm.ainvoke(
                    langchain_messages,
                    system=system or "",
                )

                latency = (time.time() - start_time) * 1000
                content = (
                    response.content if isinstance(response.content, str) else str(response.content)
                )

                result = LLMCallResult(
                    content=content,
                    model_name=model_name,
                    input_tokens=getattr(response, "usage_metadata", {}).get("input_tokens", 0)
                    or getattr(response, "input_tokens", 0),
                    output_tokens=getattr(response, "usage_metadata", {}).get("output_tokens", 0)
                    or getattr(response, "output_tokens", 0),
                    latency_ms=latency,
                    attempt=attempt,
                )

                # Token budget check
                total = result.input_tokens + result.output_tokens
                if total > opts.token_budget:
                    logger.warning(
                        "Token budget exceeded",
                        total=total,
                        budget=opts.token_budget,
                    )

                # Structured output parsing
                if opts.response_format == "json":
                    result.parsed = self._extract_json(content)

                # Audit
                await self._audit(
                    result=result,
                    messages=messages,
                    user_id=user_id,
                    tenant_id=tenant_id,
                    session_id=session_id,
                    generated_sql=generated_sql,
                )

                logger.info(
                    "LLM call complete",
                    model=result.model_name,
                    tokens_in=result.input_tokens,
                    tokens_out=result.output_tokens,
                    latency_ms=round(result.latency_ms),
                    attempt=result.attempt,
                )

                return result

            except Exception as e:
                last_error = e
                logger.warning(
                    "LLM call failed",
                    attempt=attempt,
                    model=model_name,
                    error=str(e),
                )

                if attempt < opts.max_retries:
```

```
                    backoff = min(2**attempt + (hash(str(e)) % 100) / 100, 30)
                    await self._sleep(backoff)
                continue

        # All retries exhausted
        logger.error("LLM call failed after all retries", error=str(last_error))
        raise last_error  # type: ignore[misc]

    def _build_llm(self, model_name: str, opts: LLMCallOptions, attempt: int) -> ChatAnthropic:
        """Build a ChatAnthropic instance with timeout and model kwargs."""
        timeout = opts.timeout_seconds + (attempt * 15)  # Scale timeout per attempt

        model_kwargs: dict = {}
        if opts.thinking:
            model_kwargs["thinking"] = {
                "type": "enabled",
                "budget_tokens": min(opts.max_tokens // 2, 4096),
            }

        return ChatAnthropic(
            model=model_name,
            temperature=opts.temperature,
            max_tokens=opts.max_tokens,
            api_key=settings.ANTHROPIC_API_KEY,
            timeout=timeout,
            max_retries=1,  # We handle retries ourselves
            model_kwargs=model_kwargs if model_kwargs else None,
        )

    def _normalize_messages(self, messages: str | list[BaseMessage]) -> list[BaseMessage]:
        """Normalize message input to a list of BaseMessage."""
        if isinstance(messages, str):
            return [HumanMessage(content=messages)]
        return messages

    def _extract_json(self, text: str) -> dict | None:
        """Extract a JSON object from LLM output.

        Tries multiple strategies:
        1. The text is pure JSON already
        2. JSON inside ```json ... ``` code blocks
        3. First { ... } pair in the text
        """
        text = text.strip()

        # Strategy 1: pure JSON
        if text.startswith("{") and text.endswith("}"):
            try:
                return json.loads(text)
            except json.JSONDecodeError:
                pass

        # Strategy 2: markdown code block
        match = re.search(r"```(?:json)?\s*(\{.*?\})\s*```", text, re.DOTALL)
        if match:
            try:
                return json.loads(match.group(1))
            except json.JSONDecodeError:
                pass

        # Strategy 3: first { ... } pair
        match = re.search(r"\{.*\}", text, re.DOTALL)
        if match:
            try:
                return json.loads(match.group(0))
            except json.JSONDecodeError:
                pass

        logger.warning("Failed to extract JSON from LLM output", preview=text[:200])
        return None

    async def _audit(
        self,
        result: LLMCallResult,
        messages: str | list[BaseMessage] | None = None,
        user_id: str | None = None,
        tenant_id: str | None = None,
        session_id: str | None = None,
        generated_sql: str | None = None,
    ) -> None:
        """Log this LLM call to the audit trail."""
        if self._audit_callback:
            try:
                await self._audit_callback(
                    {
                        "session_id": session_id or str(uuid.uuid4()),
                        "user_id": user_id or "unknown",
                        "tenant_id": tenant_id or "default",
                        # SHA-256 of the prompt itself (never store raw text)
                        "input_prompt_hash": hashlib.sha256(
                            str(messages or "").encode()
                        ).hexdigest(),
                        "generated_sql": generated_sql or "",
                        "model_name": result.model_name,
                        "model_version": "latest",
                        "input_tokens": result.input_tokens,
                        "output_tokens": result.output_tokens,
                        "latency_ms": result.latency_ms,
                    }
                )
            except Exception as e:
                logger.error("Audit callback failed", error=str(e))

    async def _sleep(self, seconds: float) -> None:
        """Async sleep helper."""
```

```
        import asyncio

        await asyncio.sleep(seconds)


# ---------------------------------------------------------------------------
# Prompt loading
# ---------------------------------------------------------------------------


def load_prompt(name: str) -> str:
    """Load a versioned prompt template from .claude/prompts/.

    Args:
        name: Prompt name without extension (e.g. "nl2sql-system", "chart-gen-system")

    Returns:
        The prompt text content, or "" if not found.

    Resolution order:
        1. ``settings.PROMPT_DIR`` if set (explicit override).
        2. ``/app/.claude/prompts`` if it exists (Docker container — the
           Dockerfile bakes prompts into /app/.claude).
        3. The repo root's ``.claude/prompts`` computed relative to this file
           (host/local dev, where this module is at backend/app/core/).

    The fallback (3) historically broke inside Docker because the relative path
    resolved to ``/.claude/prompts``; the explicit container path (2) fixes that.
    """
    from pathlib import Path

    from app.core.config import settings

    candidates = [
        Path(settings.PROMPT_DIR) if settings.PROMPT_DIR else None,
        Path("/app/.claude/prompts"),  # container
        Path(__file__).resolve().parent.parent.parent.parent / ".claude" / "prompts",  # host
    ]

    prompt_path: Path | None = None
    for base in candidates:
        if base is None:
            continue
        candidate = base / f"{name}.md"
        if candidate.exists():
            prompt_path = candidate
            break

    if prompt_path is None:
        logger.warning(f"Prompt file not found: {name}.md")
        return ""

    return prompt_path.read_text(encoding="utf-8")


# ---------------------------------------------------------------------------
# Singleton
# ---------------------------------------------------------------------------


_llm_client: LLMClient | None = None


def get_llm_client() -> LLMClient:
    """Get or create the singleton LLMClient instance."""
    global _llm_client
    if _llm_client is None:
        _llm_client = LLMClient()
    return _llm_client
```

backend/app/core/logging.py

37 lines · 0.9 KB

```python
"""Logging setup using loguru."""

import sys

from loguru import logger

from app.core.config import settings


def setup_logging() -> None:
    """Configure loguru for structured logging."""
    logger.remove()  # Remove default handler

    logger.add(
        sys.stdout,
        level=settings.LOG_LEVEL,
        format=(
            "<green>{time:HH:mm:ss}</green> | "
            "<level>{level: <8}</level> | "
            "<cyan>{name}</cyan>:<cyan>{function}</cyan>:<cyan>{line}</cyan> | "
            "<level>{message}</level>"
        ),
        colorize=True,
    )

    # In production, add JSON log file output
    if settings.APP_ENV == "production":
        logger.add(
            "logs/genbi_{time:YYYY-MM-DD}.json",
            level="INFO",
            serialize=True,
            rotation="1 day",
            retention="30 days",
        )


__all__ = ["logger", "setup_logging"]
```

backend/app/core/masking.py

211 lines · 6.9 KB

```python
"""PII masking — column-level data protection for query results.

Applies masking rules to sensitive fields before data reaches agents or the
frontend. Masks are configurable per-tenant via database settings.

Supported mask types:
- EMAIL: j***@company.com
- PHONE: ***-***-1234
- SSN: ***-**-1234
- CREDIT_CARD: ****-****-****-1234
- FULL_MASK: ****
- CUSTOM: regex-based pattern replacement

Architecture:
    Query Result → PII Masking Middleware → Agent Pipeline → Frontend

The masking happens at the service layer between data retrieval and the
agent pipeline, so no PII ever enters LLM context.
"""

import re
from dataclasses import dataclass
from enum import StrEnum
from typing import Any

from app.core.logging import logger


class MaskType(StrEnum):
    """Types of PII masking strategies."""

    EMAIL = "email"
    PHONE = "phone"
    SSN = "ssn"
    CREDIT_CARD = "credit_card"
    FULL_MASK = "full_mask"
    NONE = "none"


@dataclass
class ColumnMaskRule:
    """Masking rule applied to a specific column."""

    column_name: str
    mask_type: MaskType
    # For partial masking: how many characters to show at end
    visible_suffix_chars: int = 4


# ---------------------------------------------------------------------------
# Default masking rules — overridden per-tenant via DB settings
# ---------------------------------------------------------------------------

DEFAULT_MASK_RULES: list[ColumnMaskRule] = [
    ColumnMaskRule("email", MaskType.EMAIL),
    ColumnMaskRule("email_address", MaskType.EMAIL),
    ColumnMaskRule("phone", MaskType.PHONE),
    ColumnMaskRule("phone_number", MaskType.PHONE),
    ColumnMaskRule("mobile", MaskType.PHONE),
    ColumnMaskRule("ssn", MaskType.SSN),
    ColumnMaskRule("social_security", MaskType.SSN),
    ColumnMaskRule("tax_id", MaskType.SSN),
    ColumnMaskRule("credit_card", MaskType.CREDIT_CARD),
    ColumnMaskRule("card_number", MaskType.CREDIT_CARD),
    ColumnMaskRule("password", MaskType.FULL_MASK),
    ColumnMaskRule("secret", MaskType.FULL_MASK),
    ColumnMaskRule("api_key", MaskType.FULL_MASK),
    ColumnMaskRule("token", MaskType.FULL_MASK),
]


# ---------------------------------------------------------------------------
# Masking engine
# ---------------------------------------------------------------------------


class PIIMasker:
    """Applies column-level PII masking to query results.

    Usage:
        masker = PIIMasker(rules=[...])
        masked_data = masker.mask_rows(rows)
    """

    def __init__(self, rules: list[ColumnMaskRule] | None = None):
        self.rules = rules or DEFAULT_MASK_RULES
        self._rule_map: dict[str, ColumnMaskRule] = {r.column_name.lower(): r for r in self.rules}

    def mask_rows(self, rows: list[dict[str, Any]]) -> list[dict[str, Any]]:
        """Apply masking to all rows.

        Args:
            rows: List of row dicts from a query result.

        Returns:
            New list with masked values (original rows are not mutated).
        """
        if not rows:
            return rows

```

```
        masked = []
        for row in rows:
            masked_row = {}
            for column, value in row.items():
                rule = self._find_rule(column)
                if rule and rule.mask_type != MaskType.NONE and value is not None:
                    masked_row[column] = self._apply_mask(str(value), rule)
                else:
                    masked_row[column] = value
            masked.append(masked_row)

        masked_count = sum(1 for r in masked for v in r.values() if isinstance(v, str) and "*" in v)
        if masked_count:
            logger.info(f"PII masked {masked_count} values across {len(rows)} rows")

        return masked

    def _find_rule(self, column_name: str) -> ColumnMaskRule | None:
        """Find matching mask rule for a column name."""
        return self._rule_map.get(column_name.lower())

    def _apply_mask(self, value: str, rule: ColumnMaskRule) -> str:
        """Apply the appropriate masking strategy."""
        if not value:
            return value

        maskers = {
            MaskType.EMAIL: self._mask_email,
            MaskType.PHONE: self._mask_phone,
            MaskType.SSN: self._mask_ssn,
            MaskType.CREDIT_CARD: self._mask_credit_card,
            MaskType.FULL_MASK: lambda v, _: "****",
        }

        masker = maskers.get(rule.mask_type)
        if masker:
            return masker(value, rule.visible_suffix_chars)
        return value

    # --- Individual mask strategies ---

    @staticmethod
    def _mask_email(email: str, visible_suffix: int = 0) -> str:
        """jdoe@company.com → j***@company.com"""
        if "@" not in email:
            return "****"
        local, domain = email.split("@", 1)
        visible = local if len(local) <= 1 else local[0] + "***"
        return f"{visible}@{domain}"

    @staticmethod
    def _mask_phone(phone: str, visible_suffix: int = 4) -> str:
        """+1-555-123-4567 → ***-***-4567"""
        # Strip non-digit chars for counting
        digits = re.sub(r"\D", "", phone)
        if len(digits) <= visible_suffix:
            return "*" * len(digits)

        visible = digits[-visible_suffix:]
        masked_count = len(digits) - visible_suffix

        # Preserve original formatting loosely
        if "-" in phone or "(" in phone:
            return f"***-***-{visible}"
        return ("*" * masked_count) + visible

    @staticmethod
    def _mask_ssn(ssn: str, visible_suffix: int = 4) -> str:
        """123-45-6789 → ***-**-6789"""
        digits = re.sub(r"\D", "", ssn)
        if len(digits) <= visible_suffix:
            return "*" * len(digits)
        visible = digits[-visible_suffix:]
        return f"***-**-{visible}"

    @staticmethod
    def _mask_credit_card(card: str, visible_suffix: int = 4) -> str:
        """4111-1111-1111-1111 → ****-****-****-1111"""
        digits = re.sub(r"\D", "", card)
        if len(digits) <= visible_suffix:
            return "*" * len(digits)
        visible = digits[-visible_suffix:]
        return f"****-****-****-{visible}"


# ---------------------------------------------------------------------------
# Tenant-aware masking
# ---------------------------------------------------------------------------

# Per-tenant rule overrides (loaded from DB on startup)
_tenant_rules: dict[str, list[ColumnMaskRule]] = {}


def load_tenant_mask_rules(tenant_id: str, rules: list[ColumnMaskRule]) -> None:
    """Register custom masking rules for a tenant.

    Called during tenant initialization or when rules change.
    """
    _tenant_rules[tenant_id] = rules
    logger.info(f"Loaded {len(rules)} mask rules for tenant {tenant_id}")


def get_masker_for_tenant(tenant_id: str) -> PIIMasker:
    """Get a PIIMasker configured for a specific tenant.

```

```
    Falls back to DEFAULT_MASK_RULES if no tenant-specific rules exist.
    """
    custom_rules = _tenant_rules.get(tenant_id)
    if custom_rules:
        return PIIMasker(rules=custom_rules)
    return PIIMasker(rules=DEFAULT_MASK_RULES)
```

backend/app/core/observability.py

355 lines · 11.3 KB

```python
"""OpenTelemetry observability setup — tracing, metrics, and logging integration.

Wires up:
- OpenTelemetry SDK with automatic instrumentation for FastAPI, SQLAlchemy, httpx
- Langfuse exporter for LLM trace collection
- Prometheus metrics endpoint at /metrics
- Structured logging correlation via trace_id injection

Architecture:
    FastAPI Request → OTel Span → Agent Span → LLM Call Span → DB Query Span
                                        ↓
                              Langfuse Trace
                                     ↓
                              Audit Log Record
"""

import contextlib
import os

from opentelemetry import metrics, trace
from opentelemetry.exporter.otlp.proto.grpc.metric_exporter import OTLPMetricExporter
from opentelemetry.exporter.otlp.proto.grpc.trace_exporter import OTLPSpanExporter
from opentelemetry.exporter.prometheus import PrometheusMetricReader
from opentelemetry.instrumentation.fastapi import FastAPIInstrumentor
from opentelemetry.instrumentation.httpx import HTTPXClientInstrumentor
from opentelemetry.instrumentation.sqlalchemy import SQLAlchemyInstrumentor
from opentelemetry.sdk.metrics import MeterProvider
from opentelemetry.sdk.metrics.export import PeriodicExportingMetricReader
from opentelemetry.sdk.resources import DEPLOYMENT_ENVIRONMENT, SERVICE_NAME, Resource
from opentelemetry.sdk.trace import TracerProvider
from opentelemetry.sdk.trace.export import BatchSpanProcessor

from app.core.config import settings
from app.core.logging import logger

# ---------------------------------------------------------------------------
# Resource (identifies this service in traces)
# ---------------------------------------------------------------------------


def _build_resource() -> Resource:
    """Build the OTel resource with service metadata."""
    return Resource.create(
        {
            SERVICE_NAME: "genbi-backend",
            DEPLOYMENT_ENVIRONMENT: settings.APP_ENV,
            "service.version": "0.1.0",
            "service.namespace": "genbi",
        }
    )


# ---------------------------------------------------------------------------
# Tracing
# ---------------------------------------------------------------------------


def init_tracing() -> TracerProvider | None:
    """Initialize OpenTelemetry tracing.

    Exports spans to OTLP collector (if configured) and console in dev.

    Returns:
        TracerProvider or None if tracing is disabled.
    """
    otlp_endpoint = os.getenv("OTEL_EXPORTER_OTLP_ENDPOINT", "")
    if not otlp_endpoint and settings.APP_ENV == "production":
        logger.warning("OTEL_EXPORTER_OTLP_ENDPOINT not set — tracing disabled")
        return None

    resource = _build_resource()
    provider = TracerProvider(resource=resource)

    # OTLP exporter (gRPC)
    if otlp_endpoint:
        otlp_exporter = OTLPSpanExporter(endpoint=otlp_endpoint)
        provider.add_span_processor(
            BatchSpanProcessor(otlp_exporter, max_queue_size=2048, max_export_batch_size=512)
        )
        logger.info(f"OTel tracing enabled — exporting to {otlp_endpoint}")

    # Console exporter for development
    if settings.DEBUG:
        from opentelemetry.sdk.trace.export import ConsoleSpanExporter

        provider.add_span_processor(BatchSpanProcessor(ConsoleSpanExporter()))

    trace.set_tracer_provider(provider)
    return provider


def instrument_app(app):
    """Auto-instrument FastAPI, SQLAlchemy, and httpx.

    Call this after app creation but before any requests.
    """
    # FastAPI auto-instrumentation
    FastAPIInstrumentor.instrument_app(
        app,
        excluded_urls="/health,/metrics,/favicon.ico",
```

```
        tracer_provider=trace.get_tracer_provider(),
    )

    # HTTPX client instrumentation (for outbound HTTP calls — Cube.dev, LLM APIs)
    HTTPXClientInstrumentor().instrument()

    # SQLAlchemy instrumentation
    SQLAlchemyInstrumentor().instrument(
        enable_commenter=True,
        commenter_options={"enable_db_statement_parameters": False},
    )

    logger.info("OpenTelemetry instrumentation applied to FastAPI, httpx, SQLAlchemy")


# ---------------------------------------------------------------------------
# Metrics
# ---------------------------------------------------------------------------


def init_metrics() -> MeterProvider | None:
    """Initialize OpenTelemetry metrics with Prometheus endpoint.

    Returns:
        MeterProvider or None if metrics are disabled.
    """
    resource = _build_resource()
    readers = []

    # Prometheus endpoint (always available at /metrics)
    prometheus_reader = PrometheusMetricReader()
    readers.append(prometheus_reader)

    # OTLP metrics exporter (if configured)
    otlp_endpoint = os.getenv("OTEL_EXPORTER_OTLP_ENDPOINT", "")
    if otlp_endpoint:
        otlp_reader = PeriodicExportingMetricReader(
            OTLPMetricExporter(endpoint=otlp_endpoint),
            export_interval_millis=30_000,
        )
        readers.append(otlp_reader)

    provider = MeterProvider(resource=resource, metric_readers=readers)
    metrics.set_meter_provider(provider)

    # Create application-level meters
    meter = metrics.get_meter("genbi")

    # Chat pipeline metrics
    meter.create_counter(
        "genbi.chat.requests",
        description="Total chat requests processed",
        unit="requests",
    )
    meter.create_histogram(
        "genbi.chat.latency",
        description="End-to-end chat pipeline latency",
        unit="ms",
    )

    # SQL generation metrics
    meter.create_counter(
        "genbi.sql.generated",
        description="Total SQL queries generated",
        unit="queries",
    )
    meter.create_counter(
        "genbi.sql.validation_failures",
        description="SQL queries rejected by ValidationAgent",
        unit="queries",
    )

    # LLM call metrics
    meter.create_counter(
        "genbi.llm.tokens",
        description="Total LLM tokens consumed",
        unit="tokens",
    )
    meter.create_histogram(
        "genbi.llm.latency",
        description="LLM API call latency",
        unit="ms",
    )

    # Cache metrics
    meter.create_counter(
        "genbi.cache.hits",
        description="Cache hit count",
        unit="hits",
    )
    meter.create_counter(
        "genbi.cache.misses",
        description="Cache miss count",
        unit="misses",
    )

    logger.info("OTel metrics initialized — Prometheus endpoint at /metrics")
    return provider


# ---------------------------------------------------------------------------
# Langfuse integration
# ---------------------------------------------------------------------------


```

```python
class LangfuseTracer:
    """Langfuse LLM tracing integration.

    Every LLM call is traced through Langfuse for:
    - Input/output token tracking
    - Latency measurement
    - Model version tracking
    - Cost allocation per tenant/user

    Langfuse traces are tagged with:
        project=genbi, env={APP_ENV}, tenant_id, user_id, session_id
    """

    def __init__(self):
        self._client = None
        self._enabled = False

    async def _init(self) -> None:
        """Lazy-init Langfuse client."""
        if self._client is not None:
            return

        try:
            from langfuse import Langfuse

            self._client = Langfuse(
                secret_key=settings.LANGFUSE_SECRET_KEY,
                public_key=settings.LANGFUSE_PUBLIC_KEY,
                host=settings.LANGFUSE_HOST,
                release="0.1.0",
            )
            self._enabled = True
            logger.info("Langfuse tracing enabled")
        except ImportError:
            logger.warning("langfuse package not installed — LLM tracing disabled")
            self._enabled = False
        except Exception as e:
            logger.warning(f"Langfuse init failed — LLM tracing disabled: {e}")
            self._enabled = False

    async def trace_llm_call(
        self,
        name: str,
        model: str,
        input_tokens: int,
        output_tokens: int,
        latency_ms: float,
        user_id: str = "",
        tenant_id: str = "",
        session_id: str = "",
        input_prompt: str = "",
        output_content: str = "",
        metadata: dict | None = None,
    ) -> str | None:
        """Record an LLM call trace in Langfuse.

        Returns:
            Trace ID or None if tracing is disabled.
        """
        if not self._enabled:
            await self._init()
            if not self._enabled:
                return None

        try:
            trace = self._client.trace(
                name=name,
                user_id=user_id,
                session_id=session_id,
                metadata={
                    "model": model,
                    "tenant_id": tenant_id,
                    "input_tokens": input_tokens,
                    "output_tokens": output_tokens,
                    "latency_ms": latency_ms,
                    "env": settings.APP_ENV,
                    **(metadata or {}),
                },
            )

            # Record the generation span
            trace.generation(
                name=f"{name}-generation",
                model=model,
                input=input_prompt[:2000] if input_prompt else None,
                output=output_content[:2000] if output_content else None,
                usage={
                    "input": input_tokens,
                    "output": output_tokens,
                    "total": input_tokens + output_tokens,
                },
                metadata={
                    "latency_ms": latency_ms,
                    "tenant_id": tenant_id,
                },
            )

            return trace.id

        except Exception as e:
            logger.debug(f"Langfuse trace failed (non-fatal): {e}")
            return None

    async def flush(self) -> None:
        """Flush pending traces before shutdown."""
```

```
        if self._client and self._enabled:
            with contextlib.suppress(Exception):
                self._client.flush()


# ---------------------------------------------------------------------------
# Trace context helpers
# ---------------------------------------------------------------------------


def get_current_trace_id() -> str | None:
    """Get the current OpenTelemetry trace ID for log correlation."""
    span = trace.get_current_span()
    if span and span.get_span_context().is_valid:
        return format(span.get_span_context().trace_id, "032x")
    return None


def create_span(name: str, attributes: dict | None = None):
    """Create a manual span for agent operations.

    Usage:
        tracer = trace.get_tracer("genbi")
        with tracer.start_as_current_span("nl2sql.generate") as span:
            span.set_attribute("query.length", len(query))
            result = await agent.execute(query=query)
            span.set_attribute("sql.length", len(result.output.get("sql", "")))
    """
    tracer = trace.get_tracer("genbi")
    return tracer.start_as_current_span(name, attributes=attributes or {})


# ---------------------------------------------------------------------------
# Singleton
# ---------------------------------------------------------------------------

_langfuse_tracer: LangfuseTracer | None = None


def get_langfuse_tracer() -> LangfuseTracer:
    """Get or create the Langfuse tracer singleton."""
    global _langfuse_tracer
    if _langfuse_tracer is None:
        _langfuse_tracer = LangfuseTracer()
    return _langfuse_tracer
```

backend/app/core/security.py

20 lines · 0.7 KB

```python
"""Password hashing — bcrypt wrappers used by the auth service."""

import bcrypt


def hash_password(password: str) -> str:
    """Hash a plaintext password with a fresh bcrypt salt."""
    return bcrypt.hashpw(password.encode("utf-8"), bcrypt.gensalt()).decode("utf-8")


def verify_password(plain: str, hashed: str) -> bool:
    """Return True if the plaintext matches the stored bcrypt hash.

    Malformed stored hashes and passwords outside bcrypt's 72-byte limit
    surface as ValueError and are treated as a failed match.
    """
    try:
        return bcrypt.checkpw(plain.encode("utf-8"), hashed.encode("utf-8"))
    except ValueError:
        return False
```

backend/app/db/__init__.py

0 lines · 0.0 KB

```
1|
```

backend/app/db/graph_schema.py

503 lines · 15.9 KB

```python
"""Apache AGE graph database — data lineage and relationship analysis.

Apache AGE (A Graph Extension) is a PostgreSQL extension that provides
graph database capabilities via openCypher queries. GenBI uses it for:

1. Data lineage: tracking where metrics come from (source table → Cube measure)
2. Relationship analysis: discovering join paths between tables for SQL generation
3. Impact analysis: what downstream assets are affected by a schema change

Architecture:
    Schema Catalog → Graph Ingestion → AGE Graph (genbi_graph)
                                            ↓
    LineageAgent ← openCypher queries ← AGE
         ↓
    Graph context injected into NL2SQLAgent (join paths)

Every cypher statement is a complete inline literal with AGE parameter
binding (``$name`` placeholders inside the ``$$`` body; values ride in the
third ``ag_catalog.cypher`` argument, bound via the ``:params`` SQLAlchemy
named parameter) — never string-interpolated.
"""

import json
from dataclasses import dataclass, field
from enum import StrEnum
from uuid import uuid4

from sqlalchemy import text

from app.core.logging import logger

# ---------------------------------------------------------------------------
# Graph schema
# ---------------------------------------------------------------------------

# The genbi_graph graph has these vertex labels and edge types:
#
# VERTICES:
#   Table { name, schema, description }
#   Column { name, data_type, description, table_name }
#   Metric { name, description, metric_type }
#   Dashboard { name, description }
#   User { user_id, tenant_id }
#
# EDGES:
#   TABLE_CONTAINS (Table → Column) { ordinal_position }
#   METRIC_SOURCE (Metric → Column) { aggregation, expression }
#   METRIC_DEPENDS (Metric → Metric) { relationship }
#   DASHBOARD_USES (Dashboard → Metric) { position }
#   USER_CAN_ACCESS (User → Table | Metric) { role, granted_at }
#   TABLE_JOINS (Table → Table) { join_columns, join_type }


class VertexLabel(StrEnum):
    TABLE = "Table"
    COLUMN = "Column"
    METRIC = "Metric"
    DASHBOARD = "Dashboard"
    USER = "User"


class EdgeLabel(StrEnum):
    TABLE_CONTAINS = "TABLE_CONTAINS"
    METRIC_SOURCE = "METRIC_SOURCE"
    METRIC_DEPENDS = "METRIC_DEPENDS"
    DASHBOARD_USES = "DASHBOARD_USES"
    USER_CAN_ACCESS = "USER_CAN_ACCESS"
    TABLE_JOINS = "TABLE_JOINS"


# ---------------------------------------------------------------------------
# Graph initialization
# ---------------------------------------------------------------------------


async def init_age_graph(db_session) -> None:
    """Initialize the Apache AGE graph if it doesn't exist.

    Safe to call on startup — the create_graph failure for an existing
    graph is caught and rolled back.
    """
    try:
        await db_session.execute(
            text(
                "CREATE EXTENSION IF NOT EXISTS age; "
                "SET search_path = ag_catalog, public; "
                "SELECT * FROM ag_catalog.create_graph('genbi_graph');"
            )
        )
        await db_session.commit()
        logger.info("AGE graph 'genbi_graph' initialized successfully")
    except Exception as e:
        logger.info(f"AGE graph already exists or init skipped: {e}")
        await db_session.rollback()


async def create_indices(db_session) -> None:
    """Create vertex/edge labels on the graph (idempotent, best-effort)."""
    try:
        await db_session.execute(
```

```
            text("SELECT * FROM ag_catalog.create_vlabel('genbi_graph', 'Table')")
        )
    except Exception as e:
        logger.debug(f"Label creation skipped (may exist): {e}")
    try:
        await db_session.execute(
            text("SELECT * FROM ag_catalog.create_vlabel('genbi_graph', 'Column')")
        )
    except Exception as e:
        logger.debug(f"Label creation skipped (may exist): {e}")
    try:
        await db_session.execute(
            text("SELECT * FROM ag_catalog.create_vlabel('genbi_graph', 'Metric')")
        )
    except Exception as e:
        logger.debug(f"Label creation skipped (may exist): {e}")
    try:
        await db_session.execute(
            text("SELECT * FROM ag_catalog.create_vlabel('genbi_graph', 'Dashboard')")
        )
    except Exception as e:
        logger.debug(f"Label creation skipped (may exist): {e}")
    try:
        await db_session.execute(
            text("SELECT * FROM ag_catalog.create_elabel('genbi_graph', 'TABLE_CONTAINS')")
        )
    except Exception as e:
        logger.debug(f"Label creation skipped (may exist): {e}")
    try:
        await db_session.execute(
            text("SELECT * FROM ag_catalog.create_elabel('genbi_graph', 'METRIC_SOURCE')")
        )
    except Exception as e:
        logger.debug(f"Label creation skipped (may exist): {e}")
    try:
        await db_session.execute(
            text("SELECT * FROM ag_catalog.create_elabel('genbi_graph', 'DASHBOARD_USES')")
        )
    except Exception as e:
        logger.debug(f"Label creation skipped (may exist): {e}")
    try:
        await db_session.execute(
            text("SELECT * FROM ag_catalog.create_elabel('genbi_graph', 'TABLE_JOINS')")
        )
    except Exception as e:
        logger.debug(f"Label creation skipped (may exist): {e}")

    await db_session.commit()
    logger.info("AGE graph labels ensured for 'genbi_graph'")


# ---------------------------------------------------------------------------
# Data classes
# ---------------------------------------------------------------------------


@dataclass
class LineageNode:
    """A node in the data lineage graph."""

    id: str
    label: str
    properties: dict = field(default_factory=dict)


@dataclass
class LineageEdge:
    """An edge in the data lineage graph."""

    source_id: str
    target_id: str
    label: str
    properties: dict = field(default_factory=dict)


@dataclass
class LineagePath:
    """A path through the data lineage graph."""

    nodes: list[LineageNode] = field(default_factory=list)
    edges: list[LineageEdge] = field(default_factory=list)
    depth: int = 0


# ---------------------------------------------------------------------------
# Graph ingestion
# ---------------------------------------------------------------------------


async def ingest_table(
    db_session,
    table_name: str,
    schema: str = "public",
    columns: list[dict] | None = None,
) -> str:
    """Ingest a table and its columns into the lineage graph.

    Creates Table vertex and Column vertices connected by TABLE_CONTAINS edges.
    """
    table_id = str(uuid4())

    await db_session.execute(
        text(
            "SELECT * FROM ag_catalog.cypher('genbi_graph', $$ "
            "CREATE (t:Table { id: $id, name: $name, schema: $schema, description: '' }) "
```

```
            "RETURN t "
            "$$, :params) AS (result ag_catalog.agtype);"
        ),
        {"params": json.dumps({"id": table_id, "name": table_name, "schema": schema})},
    )
    await db_session.commit()

    if columns:
        for col in columns:
            await ingest_column(
                db_session,
                table_id,
                col.get("name", col.get("column_name", "")),
                col.get("type", col.get("data_type", "text")),
                col.get("description", ""),
            )

    logger.info(f"Ingested table '{schema}.{table_name}' into lineage graph")
    return table_id


async def ingest_column(
    db_session,
    table_id: str,
    column_name: str,
    data_type: str,
    description: str = "",
) -> str:
    """Add a Column vertex and TABLE_CONTAINS edge to the graph."""
    col_id = str(uuid4())

    await db_session.execute(
        text(
            "SELECT * FROM ag_catalog.cypher('genbi_graph', $$ "
            "MATCH (t:Table { id: $table_id }) "
            "CREATE (c:Column { id: $id, name: $name, data_type: $type, description: $desc }) "
            "CREATE (t)-[:TABLE_CONTAINS]->(c) "
            "RETURN c "
            "$$, :params) AS (result ag_catalog.agtype);"
        ),
        {
            "params": json.dumps(
                {
                    "table_id": table_id,
                    "id": col_id,
                    "name": column_name,
                    "type": data_type,
                    "desc": description,
                }
            )
        },
    )
    return col_id


async def ingest_metric(
    db_session,
    metric_name: str,
    description: str,
    metric_type: str,
    source_table_id: str,
    source_column_name: str,
) -> str:
    """Ingest a metric and link it to its source column."""
    metric_id = str(uuid4())

    await db_session.execute(
        text(
            "SELECT * FROM ag_catalog.cypher('genbi_graph', $$ "
            "MATCH (t:Table { id: $table_id })-[:TABLE_CONTAINS]->(c:Column { name: $column }) "
            "CREATE (m:Metric { id: $id, name: $name, description: $desc, metric_type: $mtype }) "
            "CREATE (m)-[:METRIC_SOURCE]->(c) "
            "RETURN m "
            "$$, :params) AS (result ag_catalog.agtype);"
        ),
        {
            "params": json.dumps(
                {
                    "table_id": source_table_id,
                    "column": source_column_name,
                    "id": metric_id,
                    "name": metric_name,
                    "desc": description,
                    "mtype": metric_type,
                }
            )
        },
    )
    logger.info(f"Ingested metric '{metric_name}' linked to column '{source_column_name}'")
    return metric_id


async def ingest_join_path(
    db_session,
    table_a_id: str,
    table_b_id: str,
    join_columns: str,
    join_type: str = "INNER",
) -> None:
    """Record a known join path between two tables."""
    await db_session.execute(
        text(
            "SELECT * FROM ag_catalog.cypher('genbi_graph', $$ "
            "MATCH (a:Table { id: $a }), (b:Table { id: $b }) "
            "CREATE (a)-[:TABLE_JOINS { join_columns: $cols, join_type: $jtype }]->(b) "
```

```
            "$$, :params) AS (result ag_catalog.agtype);"
        ),
        {
            "params": json.dumps(
                {"a": table_a_id, "b": table_b_id, "cols": join_columns, "jtype": join_type}
            )
        },
    )
    logger.info(f"Ingested join path: {table_a_id} -> {table_b_id} on {join_columns}")


# ---------------------------------------------------------------------------
# Graph traversal
# ---------------------------------------------------------------------------


async def get_metric_lineage(
    db_session,
    metric_name: str,
) -> list[LineagePath]:
    """Trace a metric back to its source tables and columns.

    Returns the full lineage path: Metric → Column → Table.
    """
    result = await db_session.execute(
        text(
            "SELECT * FROM ag_catalog.cypher('genbi_graph', $$ "
            "MATCH path = (m:Metric { name: $name })-[:METRIC_SOURCE]->(c:Column)"
            "<-[:TABLE_CONTAINS]-(t:Table) "
            "RETURN path "
            "$$, :params) AS (result ag_catalog.agtype);"
        ),
        {"params": json.dumps({"name": metric_name})},
    )
    rows = result.fetchall()

    paths = []
    for row in rows:
        # Parse AGE path format
        path_data = row[0] if row else None
        if path_data:
            # Simplified parsing — in production, use AGE's path parsing
            paths.append(
                LineagePath(
                    nodes=[
                        LineageNode(
                            id=str(uuid4()), label="Metric", properties={"name": metric_name}
                        ),
                    ],
                    depth=1,
                )
            )

    return paths


async def find_join_paths(
    db_session,
    table_a_id: str,
    table_b_id: str,
    max_depth: int = 3,
) -> list[LineagePath]:
    """Find all join paths between two tables (up to max_depth hops).

    This is used by NL2SQLAgent to discover how to join tables in generated SQL.
    """
    result = await db_session.execute(
        text(
            "SELECT * FROM ag_catalog.cypher('genbi_graph', $$ "
            "MATCH path = (a:Table { id: $a })-[:TABLE_JOINS*1..3]-(b:Table { id: $b }) "
            "RETURN path "
            "ORDER BY length(path) "
            "LIMIT 5 "
            "$$, :params) AS (result ag_catalog.agtype);"
        ),
        {"params": json.dumps({"a": table_a_id, "b": table_b_id})},
    )
    rows = result.fetchall()

    paths = []
    for _ in rows:
        paths.append(
            LineagePath(
                nodes=[
                    LineageNode(id=str(uuid4()), label="Table", properties={"id": table_a_id}),
                    LineageNode(id=str(uuid4()), label="Table", properties={"id": table_b_id}),
                ],
                depth=2,
            )
        )

    return paths


async def get_downstream_impact(
    db_session,
    table_name: str,
) -> list[dict]:
    """Find all downstream metrics and dashboards affected by a table change.

    Used for impact analysis before schema migrations.
    """
    result = await db_session.execute(
        text(
            "SELECT * FROM ag_catalog.cypher('genbi_graph', $$ "
```

```
            "MATCH (t:Table { name: $name })-[:TABLE_CONTAINS]->(c:Column)"
            "<-[:METRIC_SOURCE]-(m:Metric)-[:DASHBOARD_USES]-(d:Dashboard) "
            "RETURN t.name AS table_name, m.name AS metric_name, collect(d.name) AS dashboards "
            "$$, :params) AS (table_name ag_catalog.agtype, metric_name ag_catalog.agtype, "
            "dashboards ag_catalog.agtype);"
        ),
        {"params": json.dumps({"name": table_name})},
    )
    rows = result.fetchall()

    impact = []
    for row in rows:
        impact.append(
            {
                "table": str(row[0]) if row[0] else "",
                "metric": str(row[1]) if row[1] else "",
                "dashboards": str(row[2]) if row[2] else "[]",
            }
        )

    return impact


# ---------------------------------------------------------------------------
# Sync with semantic layer
# ---------------------------------------------------------------------------


async def sync_semantic_layer_to_graph(
    db_session,
    cube_client=None,
) -> dict[str, int]:
    """Sync Cube.dev metric definitions into the AGE lineage graph.

    Called nightly to keep the graph in sync with the semantic layer.
    """
    tables_ingested = 0
    metrics_ingested = 0
    joins_ingested = 0

    # Ingest from Cube meta if available
    if cube_client:
        try:
            meta = await cube_client.get_meta(force_refresh=True)
            for cube in meta.cubes:
                cube_name = cube.get("name", "")

                # Ingest table
                columns = []
                for dim in cube.get("dimensions", []):
                    columns.append(
                        {
                            "name": dim.get("name", ""),
                            "type": dim.get("type", "string"),
                            "description": dim.get("title", ""),
                        }
                    )

                table_id = await ingest_table(
                    db_session,
                    table_name=cube_name,
                    schema="public",
                    columns=columns,
                )
                tables_ingested += 1

                # Ingest measures as metrics
                for measure in cube.get("measures", []):
                    await ingest_metric(
                        db_session,
                        metric_name=f"{cube_name}.{measure.get('name', '')}",
                        description=measure.get("description", ""),
                        metric_type=measure.get("type", "sum"),
                        source_table_id=table_id,
                        source_column_name=measure.get("name", ""),
                    )
                    metrics_ingested += 1

        except Exception as e:
            logger.warning(f"Semantic layer sync via Cube failed: {e}")

    await db_session.commit()

    return {
        "tables_ingested": tables_ingested,
        "metrics_ingested": metrics_ingested,
        "joins_ingested": joins_ingested,
    }
```

backend/app/db/migrations/env.py

69 lines · 2.1 KB

```python
"""Generic single-database configuration (async engine).

See: https://alembic.sqlalchemy.org/en/latest/
The app ships only asyncpg (no psycopg2), so the sync URL is normalized to
the asyncpg driver and migrations run through an AsyncEngine with run_sync.
"""

import asyncio
from logging.config import fileConfig

from alembic import context
from sqlalchemy import pool
from sqlalchemy.engine import make_url
from sqlalchemy.ext.asyncio import async_engine_from_config

from app.core.config import settings
from app.db.models import Base

# Alembic Config object
config = context.config

if config.config_file_name is not None:
    fileConfig(config.config_file_name)

# Override sqlalchemy.url from settings — normalize to the asyncpg driver.
# NOTE: str(URL) masks the password as "***"; render explicitly.
url = make_url(settings.DATABASE_URL_SYNC)
if url.drivername in ("postgresql", "postgresql+psycopg2"):
    url = url.set(drivername="postgresql+asyncpg")
config.set_main_option("sqlalchemy.url", url.render_as_string(hide_password=False))

# Target metadata for autogenerate
target_metadata = Base.metadata


def run_migrations_offline() -> None:
    """Run migrations in 'offline' mode."""
    context.configure(
        url=url.render_as_string(hide_password=False),
        target_metadata=target_metadata,
        literal_binds=True,
        dialect_opts={"paramstyle": "named"},
    )
    with context.begin_transaction():
        context.run_migrations()


def do_run_migrations(connection) -> None:
    context.configure(connection=connection, target_metadata=target_metadata)
    with context.begin_transaction():
        context.run_migrations()


async def run_migrations_online() -> None:
    """Run migrations in 'online' mode via an async engine."""
    connectable = async_engine_from_config(
        config.get_section(config.config_ini_section, {}),
        prefix="sqlalchemy.",
        poolclass=pool.NullPool,
    )
    async with connectable.connect() as connection:
        await connection.run_sync(do_run_migrations)
    await connectable.dispose()


if context.is_offline_mode():
    run_migrations_offline()
else:
    asyncio.run(run_migrations_online())
```

backend/app/db/migrations/script.py.mako

26 lines · 0.6 KB

```
"""${message}

Revision ID: ${up_revision}
Revises: ${down_revision | comma,n}
Create Date: ${create_date}

"""
from typing import Sequence, Union

from alembic import op
import sqlalchemy as sa
${imports if imports else ""}

# revision identifiers, used by Alembic.
revision: str = ${repr(up_revision)}
down_revision: Union[str, None] = ${repr(down_revision)}
branch_labels: Union[str, Sequence[str], None] = ${repr(branch_labels)}
depends_on: Union[str, Sequence[str], None] = ${repr(depends_on)}


def upgrade() -> None:
    ${upgrades if upgrades else "pass"}


def downgrade() -> None:
    ${downgrades if downgrades else "pass"}
```

backend/app/db/migrations/versions/.gitkeep

0 lines · 0.0 KB

```
1|
```

backend/app/db/migrations/versions/0001_baseline_schema.py

263 lines · 9.5 KB

```python
"""baseline schema — mirrors infra/postgres/init.sql.

Revision ID: 0001_baseline
Revises:
Create Date: 2026-08-05

This migration is the Alembic mirror of the schema that init.sql creates on
first Docker boot. Two equally valid paths to a schema'd database:

  1. Docker entrypoint (init.sql) — applies on container creation, then this
     migration is a no-op because every CREATE uses IF NOT EXISTS.
  2. Pure Alembic (fresh DB, no init.sql) — this migration creates everything.

After this baseline, ALL schema changes go through Alembic only (no edits to
init.sql). See docs/adr/005-age-and-pgvector-image.md.
"""

from collections.abc import Sequence

import sqlalchemy as sa
from alembic import op
from sqlalchemy.dialects import postgresql

# revision identifiers, used by Alembic.
revision: str = "0001_baseline"
down_revision: str | None = None
branch_labels: str | Sequence[str] | None = None
depends_on: str | Sequence[str] | None = None


def upgrade() -> None:
    # Extensions (idempotent)
    op.execute('CREATE EXTENSION IF NOT EXISTS "pgcrypto"')
    op.execute('CREATE EXTENSION IF NOT EXISTS "vector"')

    # tenants (control table — not RLS-enrolled)
    op.create_table(
        "tenants",
        sa.Column(
            "id",
            postgresql.UUID(as_uuid=True),
            primary_key=True,
            server_default=sa.text("gen_random_uuid()"),
        ),
        sa.Column("name", sa.String(255), nullable=False, unique=True),
        sa.Column(
            "created_at",
            sa.DateTime(timezone=True),
            nullable=False,
            server_default=sa.text("NOW()"),
        ),
        sa.Column(
            "updated_at",
            sa.DateTime(timezone=True),
            nullable=False,
            server_default=sa.text("NOW()"),
        ),
    )
    op.execute(
        "INSERT INTO tenants (id, name) VALUES "
        "('00000000-0000-0000-0000-000000000001', 'default') "
        "ON CONFLICT (id) DO NOTHING"
    )

    # users
    op.create_table(
        "users",
        sa.Column(
            "id",
            postgresql.UUID(as_uuid=True),
            primary_key=True,
            server_default=sa.text("gen_random_uuid()"),
        ),
        sa.Column(
            "tenant_id", postgresql.UUID(as_uuid=True), sa.ForeignKey("tenants.id"), nullable=False
        ),
        sa.Column("email", sa.String(255), nullable=False),
        sa.Column("hashed_password", sa.String(255), nullable=False),
        sa.Column(
            "roles", postgresql.JSONB, nullable=False, server_default=sa.text("'[\"user\"]'")
        ),
        sa.Column(
            "created_at",
            sa.DateTime(timezone=True),
            nullable=False,
            server_default=sa.text("NOW()"),
        ),
        sa.Column(
            "updated_at",
            sa.DateTime(timezone=True),
            nullable=False,
            server_default=sa.text("NOW()"),
        ),
        sa.UniqueConstraint("tenant_id", "email"),
    )

    # audit_log
    op.create_table(
        "audit_log",
        sa.Column(
```

```
            "id",
            postgresql.UUID(as_uuid=True),
            primary_key=True,
            server_default=sa.text("gen_random_uuid()"),
        ),
        sa.Column("session_id", postgresql.UUID(as_uuid=True), nullable=False),
        sa.Column("user_id", postgresql.UUID(as_uuid=True), nullable=False),
        sa.Column(
            "tenant_id", postgresql.UUID(as_uuid=True), sa.ForeignKey("tenants.id"), nullable=False
        ),
        sa.Column("input_prompt_hash", sa.String(64), nullable=False),
        sa.Column("generated_sql", sa.Text),
        sa.Column("model_name", sa.String(100), nullable=False),
        sa.Column("model_version", sa.String(50), nullable=False),
        sa.Column("input_tokens", sa.Integer, nullable=False, server_default=sa.text("0")),
        sa.Column("output_tokens", sa.Integer, nullable=False, server_default=sa.text("0")),
        sa.Column("latency_ms", sa.Float, nullable=False, server_default=sa.text("0")),
        sa.Column("feedback_score", sa.Integer),
        sa.Column(
            "hallucination_flag", sa.Boolean, nullable=False, server_default=sa.text("FALSE")
        ),
        sa.Column(
            "created_at",
            sa.DateTime(timezone=True),
            nullable=False,
            server_default=sa.text("NOW()"),
        ),
    )
    op.create_index("idx_audit_log_session", "audit_log", ["session_id"])
    op.create_index("idx_audit_log_tenant", "audit_log", ["tenant_id"])
    op.create_index("idx_audit_log_created", "audit_log", ["created_at"])

    # conversations
    op.create_table(
        "conversations",
        sa.Column(
            "id",
            postgresql.UUID(as_uuid=True),
            primary_key=True,
            server_default=sa.text("gen_random_uuid()"),
        ),
        sa.Column("user_id", postgresql.UUID(as_uuid=True), nullable=False),
        sa.Column(
            "tenant_id", postgresql.UUID(as_uuid=True), sa.ForeignKey("tenants.id"), nullable=False
        ),
        sa.Column("title", sa.String(500)),
        sa.Column(
            "created_at",
            sa.DateTime(timezone=True),
            nullable=False,
            server_default=sa.text("NOW()"),
        ),
        sa.Column(
            "updated_at",
            sa.DateTime(timezone=True),
            nullable=False,
            server_default=sa.text("NOW()"),
        ),
    )
    op.create_index("idx_conversations_tenant", "conversations", ["tenant_id"])

    # schema_embeddings (one row per table; matches embed_schema.py)
    op.create_table(
        "schema_embeddings",
        sa.Column(
            "id",
            postgresql.UUID(as_uuid=True),
            primary_key=True,
            server_default=sa.text("gen_random_uuid()"),
        ),
        sa.Column(
            "tenant_id", postgresql.UUID(as_uuid=True), sa.ForeignKey("tenants.id"), nullable=False
        ),
        sa.Column(
            "table_schema", sa.String(255), nullable=False, server_default=sa.text("'public'")
        ),
        sa.Column("table_name", sa.String(255), nullable=False),
        sa.Column("full_name", sa.String(512), nullable=False),
        sa.Column("table_description", sa.Text, nullable=False, server_default=sa.text("''")),
        sa.Column("columns_json", postgresql.JSONB, nullable=False, server_default=sa.text("'[]'")),
        sa.Column("embedding_text", sa.Text, nullable=False, server_default=sa.text("''")),
        sa.Column("embedding", postgresql.ARRAY(sa.Float), nullable=True),  # see note below
        sa.Column(
            "created_at",
            sa.DateTime(timezone=True),
            nullable=False,
            server_default=sa.text("NOW()"),
        ),
        sa.Column(
            "updated_at",
            sa.DateTime(timezone=True),
            nullable=False,
            server_default=sa.text("NOW()"),
        ),
        sa.UniqueConstraint("tenant_id", "table_schema", "table_name"),
    )
    # Convert the embedding column to pgvector type if the extension is present.
    # Using ARRAY above keeps the migration portable; this cast makes it a real
    # VECTOR(1536) column where pgvector is installed.
    op.execute(
        "DO $$ BEGIN "
        "  ALTER TABLE schema_embeddings ALTER COLUMN embedding TYPE VECTOR(1536); "
        "EXCEPTION WHEN undefined_object THEN NULL; END $$;"
    )
    op.execute(
```

```
        "CREATE INDEX IF NOT EXISTS idx_schema_embeddings_lookup "
        "ON schema_embeddings USING ivfflat (embedding vector_cosine_ops) WITH (lists = 100)"
    )
    op.create_index("idx_schema_embeddings_tenant", "schema_embeddings", ["tenant_id"])

    # agent_examples
    op.create_table(
        "agent_examples",
        sa.Column(
            "id",
            postgresql.UUID(as_uuid=True),
            primary_key=True,
            server_default=sa.text("gen_random_uuid()"),
        ),
        sa.Column("agent_name", sa.String(100), nullable=False),
        sa.Column("nl_query", sa.Text, nullable=False),
        sa.Column("expected_sql", sa.Text, nullable=False),
        sa.Column("tags", postgresql.JSONB, server_default=sa.text("'[]'")),
        sa.Column("embedding", postgresql.ARRAY(sa.Float), nullable=True),
        sa.Column(
            "tenant_id", postgresql.UUID(as_uuid=True), sa.ForeignKey("tenants.id"), nullable=False
        ),
        sa.Column(
            "created_at",
            sa.DateTime(timezone=True),
            nullable=False,
            server_default=sa.text("NOW()"),
        ),
    )
    op.execute(
        "DO $$ BEGIN "
        "  ALTER TABLE agent_examples ALTER COLUMN embedding TYPE VECTOR(1536); "
        "EXCEPTION WHEN undefined_object THEN NULL; END $$;"
    )
    op.create_index("idx_agent_examples_agent", "agent_examples", ["agent_name"])
    op.create_index("idx_agent_examples_tenant", "agent_examples", ["tenant_id"])

    # RLS — every tenant-scoped table. FORCE closes the owner-bypass loophole.
    for tbl in ("users", "audit_log", "conversations", "schema_embeddings", "agent_examples"):
        op.execute(f"ALTER TABLE {tbl} ENABLE ROW LEVEL SECURITY")
        op.execute(f"ALTER TABLE {tbl} FORCE ROW LEVEL SECURITY")
        op.execute(
            f"CREATE POLICY tenant_isolation ON {tbl} "
            f"USING (tenant_id = current_setting('app.current_tenant_id', true)::UUID)"
        )


def downgrade() -> None:
    # Destructive on purpose — Alembic downgrades are explicit and reviewed.
    for tbl in (
        "agent_examples",
        "schema_embeddings",
        "conversations",
        "audit_log",
        "users",
        "tenants",
    ):
        op.execute(f"DROP TABLE IF EXISTS {tbl} CASCADE")
```

backend/app/db/migrations/versions/0002_app_roles_rls.py

96 lines · 4.3 KB

```python
"""app roles + enforced RLS — runtime no longer connects as superuser.

Revision ID: 0002_app_roles
Revises: 0001_baseline
Create Date: 2026-08-15

Before this migration the backend connected as POSTGRES_USER (a superuser),
so every FORCE ROW LEVEL SECURITY policy was silently bypassed — tenant
isolation was configured but never enforced. This migration introduces two
non-superuser LOGIN roles and the grants/policies they need:

  - genbi_app  — runtime role (ORM engine + query connector). Bound by the
    tenant_isolation policies via the `app.current_tenant_id` GUC the
    connector already sets per transaction.
  - genbi_auth — login-only role. Reads the `users` table across tenants
    (credential lookup requires it); a permissive policy is scoped TO this
    role and nothing else is granted to it.

It also grants the runtime role access to existing tables via default
privileges. RLS enrollment of the analytics tables lives in 0003_analytics
(which creates them). See docs/adr/006-enforced-rls-roles.md.

Passwords are fixed dev defaults, consistent with infra/postgres/init.sql
and scripts/gen-env.sh. Rotate in production (see README security section).
"""

from collections.abc import Sequence

from alembic import op

# revision identifiers, used by Alembic.
revision: str = "0002_app_roles"
down_revision: str | None = "0001_baseline"
branch_labels: str | Sequence[str] | None = None
depends_on: str | Sequence[str] | None = None


def upgrade() -> None:
    # --- Roles (cluster-level; CREATE ROLE has no IF NOT EXISTS) ------------
    op.execute(
        """
        DO $$
        BEGIN
            IF NOT EXISTS (SELECT 1 FROM pg_roles WHERE rolname = 'genbi_app') THEN
                CREATE ROLE genbi_app LOGIN PASSWORD 'genbi_app';
            END IF;
            IF NOT EXISTS (SELECT 1 FROM pg_roles WHERE rolname = 'genbi_auth') THEN
                CREATE ROLE genbi_auth LOGIN PASSWORD 'genbi_auth';
            END IF;
        END $$;
        """
    )

    # --- Grants -------------------------------------------------------------
    # genbi_app: everything the runtime needs, nothing more. No superuser, no
    # BYPASSRLS, no CREATE — RLS policies always apply to it. Statements are
    # literal (identifiers cannot be bound as parameters).
    op.execute("GRANT USAGE ON SCHEMA public TO genbi_app, genbi_auth")
    op.execute("GRANT SELECT ON tenants TO genbi_app")
    op.execute(
        "GRANT SELECT, INSERT, UPDATE, DELETE ON "
        "users, audit_log, conversations, schema_embeddings, agent_examples "
        "TO genbi_app"
    )
    # Analytics tables arrive in 0003_analytics (later in the chain) and are
    # covered by the default-privilege grant below; the explicit ALL TABLES
    # grant covers anything that already exists at 0002 time.
    op.execute("GRANT SELECT ON ALL TABLES IN SCHEMA public TO genbi_app")
    op.execute("ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO genbi_app")

    # genbi_auth: SELECT on users ONLY (login credential lookup).
    op.execute("GRANT SELECT ON users TO genbi_auth")

    # --- Login-lookup policy ------------------------------------------------
    # The only RLS carve-out: genbi_auth may read every users row. Scoped TO
    # the role so no other role gains cross-tenant visibility. Permissive
    # policies OR with tenant_isolation, so this widens only genbi_auth.
    op.execute("DROP POLICY IF EXISTS users_login_lookup ON users")
    op.execute("CREATE POLICY users_login_lookup ON users FOR SELECT TO genbi_auth USING (true)")


def downgrade() -> None:
    op.execute("DROP POLICY IF EXISTS users_login_lookup ON users")
    op.execute("REVOKE SELECT ON users FROM genbi_auth")
    op.execute("REVOKE SELECT ON ALL TABLES IN SCHEMA public FROM genbi_app")
    op.execute("ALTER DEFAULT PRIVILEGES IN SCHEMA public REVOKE SELECT ON TABLES FROM genbi_app")
    op.execute(
        "REVOKE SELECT, INSERT, UPDATE, DELETE ON "
        "users, audit_log, conversations, schema_embeddings, agent_examples "
        "FROM genbi_app"
    )
    op.execute("REVOKE SELECT ON tenants FROM genbi_app")
    op.execute("REVOKE USAGE ON SCHEMA public FROM genbi_app, genbi_auth")
    # Roles only after their grants are gone.
    op.execute("DROP ROLE IF EXISTS genbi_auth")
    op.execute("DROP ROLE IF EXISTS genbi_app")
```

backend/app/db/migrations/versions/0003_analytics_tables.py

219 lines · 7.6 KB

```python
"""analytics seed tables — tenant-scoped demo data with FORCE RLS.

Revision ID: 0003_analytics
Revises: 0002_app_roles
Create Date: 2026-08-15

Creates the 10 synthetic analytics tables used by the demo/seed dataset
(scripts/seed_test_data.py fills them; it no longer performs DDL). Every
table carries tenant_id and gets the same FORCE RLS + tenant_isolation
policy as the metadata tables, so the genbi_app runtime role is tenant-
scoped on analytics queries too — the connector already sets the
`app.current_tenant_id` GUC per transaction.

Note: `web_users` (was `users`) was renamed to stop shadowing the RLS-
enrolled app-login table of the same name in the public schema.

Statements use IF NOT EXISTS / IF EXISTS so this converges on databases
where earlier seed-script DDL already ran. RLS enrollment and teardown use
literal DO blocks with %I identifier quoting (identifiers cannot be bound
as parameters).
"""

from collections.abc import Sequence

from alembic import op

# revision identifiers, used by Alembic.
revision: str = "0003_analytics"
down_revision: str | None = "0002_app_roles"
branch_labels: str | Sequence[str] | None = None
depends_on: str | Sequence[str] | None = None


def upgrade() -> None:
    op.execute(
        """
        CREATE TABLE IF NOT EXISTS public.regions (
            id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
            tenant_id UUID NOT NULL,
            region_name VARCHAR(100) NOT NULL,
            created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
            updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
        )
        """
    )
    op.execute(
        """
        CREATE TABLE IF NOT EXISTS public.products (
            id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
            tenant_id UUID NOT NULL,
            product_name VARCHAR(200) NOT NULL,
            category VARCHAR(100) NOT NULL,
            price NUMERIC(15, 2) NOT NULL DEFAULT 0,
            created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
            updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
        )
        """
    )
    op.execute(
        """
        CREATE TABLE IF NOT EXISTS public.customers (
            id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
            tenant_id UUID NOT NULL,
            name VARCHAR(200) NOT NULL,
            email VARCHAR(200),
            country VARCHAR(100),
            signup_date DATE NOT NULL,
            status VARCHAR(20) NOT NULL DEFAULT 'active',
            created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
            updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
        )
        """
    )
    op.execute(
        """
        CREATE TABLE IF NOT EXISTS public.web_users (
            id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
            tenant_id UUID NOT NULL,
            name VARCHAR(200) NOT NULL,
            email VARCHAR(200),
            country VARCHAR(100),
            signup_date DATE NOT NULL,
            last_login DATE,
            status VARCHAR(20) NOT NULL DEFAULT 'active',
            created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
            updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
        )
        """
    )
    op.execute(
        """
        CREATE TABLE IF NOT EXISTS public.sales_representatives (
            id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
            tenant_id UUID NOT NULL,
            name VARCHAR(200) NOT NULL,
            email VARCHAR(200),
            region_id UUID NOT NULL REFERENCES regions(id),
            hire_date DATE NOT NULL,
            created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
            updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
```

```
        )
        """
    )
    op.execute(
        """
        CREATE TABLE IF NOT EXISTS public.deals (
            id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
            tenant_id UUID NOT NULL,
            amount NUMERIC(15, 2) NOT NULL,
            rep_id UUID NOT NULL REFERENCES sales_representatives(id),
            region_id UUID NOT NULL REFERENCES regions(id),
            close_date DATE NOT NULL,
            stage VARCHAR(30) NOT NULL,
            created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
            updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
        )
        """
    )
    op.execute(
        """
        CREATE TABLE IF NOT EXISTS public.sales (
            id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
            tenant_id UUID NOT NULL,
            region VARCHAR(50) NOT NULL,
            product_id UUID NOT NULL,
            product_name VARCHAR(200) NOT NULL,
            revenue NUMERIC(15, 2) NOT NULL DEFAULT 0,
            units INTEGER NOT NULL DEFAULT 0,
            transaction_date DATE NOT NULL,
            rep_id UUID NOT NULL,
            created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
            updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
        )
        """
    )
    op.execute(
        """
        CREATE TABLE IF NOT EXISTS public.orders (
            id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
            tenant_id UUID NOT NULL,
            customer_id UUID NOT NULL,
            product_id UUID NOT NULL,
            amount NUMERIC(15, 2) NOT NULL,
            order_date DATE NOT NULL,
            status VARCHAR(20) NOT NULL DEFAULT 'completed',
            created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
            updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
        )
        """
    )
    op.execute(
        """
        CREATE TABLE IF NOT EXISTS public.transactions (
            id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
            tenant_id UUID NOT NULL,
            amount NUMERIC(15, 2) NOT NULL,
            transaction_date DATE NOT NULL,
            type VARCHAR(30) NOT NULL,
            status VARCHAR(20) NOT NULL DEFAULT 'completed',
            created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
            updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
        )
        """
    )
    op.execute(
        """
        CREATE TABLE IF NOT EXISTS public.activity (
            id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
            tenant_id UUID NOT NULL,
            user_id UUID NOT NULL,
            activity_date DATE NOT NULL,
            event_type VARCHAR(50) NOT NULL,
            created_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
        )
        """
    )

    # RLS enrollment — same policy shape as the metadata tables (0001/0002).
    # Static table list inside a literal DO block; %I quotes identifiers.
    op.execute(
        """
        DO $$
        DECLARE t text;
        BEGIN
            FOREACH t IN ARRAY ARRAY[
                'regions', 'products', 'customers', 'web_users',
                'sales_representatives', 'deals', 'sales', 'orders',
                'transactions', 'activity'
            ] LOOP
                EXECUTE format('ALTER TABLE public.%I ENABLE ROW LEVEL SECURITY', t);
                EXECUTE format('ALTER TABLE public.%I FORCE ROW LEVEL SECURITY', t);
                EXECUTE format('DROP POLICY IF EXISTS tenant_isolation ON public.%I', t);
                EXECUTE format(
                    'CREATE POLICY tenant_isolation ON public.%I '
                    'USING (tenant_id = current_setting(''app.current_tenant_id'', true)::UUID)',
                    t
                );
            END LOOP;
        END $$;
        """
    )


def downgrade() -> None:
    op.execute(
```

```
        """
        DO $$
        DECLARE t text;
        BEGIN
            FOREACH t IN ARRAY ARRAY[
                'activity', 'transactions', 'orders', 'sales',
                'deals', 'sales_representatives', 'web_users',
                'customers', 'products', 'regions'
            ] LOOP
                EXECUTE format('DROP TABLE IF EXISTS public.%I CASCADE', t);
            END LOOP;
        END $$;
        """
    )
```

backend/app/db/migrations/versions/0004_messages.py

57 lines · 2.2 KB

```python
"""conversations messages — persisted multi-turn chat history.

Revision ID: 0004_messages
Revises: 0003_analytics
Create Date: 2026-08-15

The conversations table existed since the baseline but had nowhere to store
turns; conversation_id round-tripped through the chat API unused. This
migration adds the messages table (RLS-enforced like every tenant table)
that backs Phase 14: history replay in the UI and "## Conversation History"
injection into the NL2SQL prompt.
"""

from collections.abc import Sequence

from alembic import op

# revision identifiers, used by Alembic.
revision: str = "0004_messages"
down_revision: str | None = "0003_analytics"
branch_labels: str | Sequence[str] | None = None
depends_on: str | Sequence[str] | None = None


def upgrade() -> None:
    op.execute(
        """
        CREATE TABLE IF NOT EXISTS public.messages (
            id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
            conversation_id UUID NOT NULL REFERENCES conversations(id),
            tenant_id UUID NOT NULL REFERENCES tenants(id),
            role VARCHAR(20) NOT NULL,
            content TEXT NOT NULL,
            generated_sql TEXT,
            created_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
        )
        """
    )
    op.execute(
        "CREATE INDEX IF NOT EXISTS idx_messages_conversation ON public.messages(conversation_id)"
    )
    op.execute("CREATE INDEX IF NOT EXISTS idx_messages_tenant ON public.messages(tenant_id)")
    op.execute("ALTER TABLE public.messages ENABLE ROW LEVEL SECURITY")
    op.execute("ALTER TABLE public.messages FORCE ROW LEVEL SECURITY")
    op.execute("DROP POLICY IF EXISTS tenant_isolation ON public.messages")
    op.execute(
        "CREATE POLICY tenant_isolation ON public.messages "
        "USING (tenant_id = current_setting('app.current_tenant_id', true)::UUID)"
    )
    # genbi_app already holds DML on tenant tables via 0002's grant list —
    # messages postdates it, so grant explicitly (plus default privileges
    # cover future tables from 0002 onward; this is belt-and-braces).
    op.execute("GRANT SELECT, INSERT, UPDATE, DELETE ON public.messages TO genbi_app")


def downgrade() -> None:
    op.execute("DROP TABLE IF EXISTS public.messages CASCADE")
```

backend/app/db/migrations/versions/0005_reports.py

87 lines · 3.4 KB

```python
"""reports — persisted multi-chart reports.

Revision ID: 0005_reports
Revises: 0004_messages
Create Date: 2026-08-15

Phase 16: the last original-scaffold surface. A report is an LLM-planned
collection of metric sections (Cube query + rendered chart + narrative);
reports + report_sections follow the standard tenant-table recipe (RLS
FORCE + tenant_isolation + genbi_app grants), mirroring 0004_messages.
"""

from collections.abc import Sequence

from alembic import op

# revision identifiers, used by Alembic.
revision: str = "0005_reports"
down_revision: str | None = "0004_messages"
branch_labels: str | Sequence[str] | None = None
depends_on: str | Sequence[str] | None = None


def upgrade() -> None:
    op.execute(
        """
        CREATE TABLE IF NOT EXISTS public.reports (
            id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
            tenant_id UUID NOT NULL REFERENCES tenants(id),
            user_id UUID NOT NULL,
            prompt TEXT NOT NULL,
            title VARCHAR(500) NOT NULL,
            summary TEXT,
            status VARCHAR(20) NOT NULL DEFAULT 'complete',
            created_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
        )
        """
    )
    op.execute("CREATE INDEX IF NOT EXISTS idx_reports_tenant ON public.reports(tenant_id)")
    op.execute("CREATE INDEX IF NOT EXISTS idx_reports_user ON public.reports(user_id)")
    op.execute("ALTER TABLE public.reports ENABLE ROW LEVEL SECURITY")
    op.execute("ALTER TABLE public.reports FORCE ROW LEVEL SECURITY")
    op.execute("DROP POLICY IF EXISTS tenant_isolation ON public.reports")
    op.execute(
        "CREATE POLICY tenant_isolation ON public.reports "
        "USING (tenant_id = current_setting('app.current_tenant_id', true)::UUID)"
    )

    op.execute(
        """
        CREATE TABLE IF NOT EXISTS public.report_sections (
            id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
            report_id UUID NOT NULL REFERENCES reports(id),
            tenant_id UUID NOT NULL REFERENCES tenants(id),
            position INTEGER NOT NULL DEFAULT 0,
            metric_name VARCHAR(255) NOT NULL,
            section_title VARCHAR(500) NOT NULL,
            chart_spec JSONB NOT NULL DEFAULT '{}',
            chart_svg TEXT,
            data_total DOUBLE PRECISION,
            row_count INTEGER NOT NULL DEFAULT 0,
            narrative TEXT,
            created_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
        )
        """
    )
    op.execute(
        "CREATE INDEX IF NOT EXISTS idx_report_sections_report ON public.report_sections(report_id)"
    )
    op.execute(
        "CREATE INDEX IF NOT EXISTS idx_report_sections_tenant ON public.report_sections(tenant_id)"
    )
    op.execute("ALTER TABLE public.report_sections ENABLE ROW LEVEL SECURITY")
    op.execute("ALTER TABLE public.report_sections FORCE ROW LEVEL SECURITY")
    op.execute("DROP POLICY IF EXISTS tenant_isolation ON public.report_sections")
    op.execute(
        "CREATE POLICY tenant_isolation ON public.report_sections "
        "USING (tenant_id = current_setting('app.current_tenant_id', true)::UUID)"
    )

    op.execute("GRANT SELECT, INSERT, UPDATE, DELETE ON public.reports TO genbi_app")
    op.execute("GRANT SELECT, INSERT, UPDATE, DELETE ON public.report_sections TO genbi_app")


def downgrade() -> None:
    op.execute("DROP TABLE IF EXISTS public.report_sections CASCADE")
    op.execute("DROP TABLE IF EXISTS public.reports CASCADE")
```

backend/app/db/models.py

226 lines · 10.1 KB

```python
"""Database models — SQLAlchemy ORM definitions.

These models are the ORM mirror of the schema created by the Alembic baseline
migration (originally defined in infra/postgres/init.sql). The baseline schema
is owned by init.sql + Alembic; this module is the source of truth for
application-level queries and for future autogenerate.

All tenant-scoped tables follow the standard template:
    id          UUID PRIMARY KEY DEFAULT gen_random_uuid()
    tenant_id   UUID NOT NULL REFERENCES tenants(id)
    created_at  TIMESTAMPTZ NOT NULL DEFAULT NOW()
    updated_at  TIMESTAMPTZ NOT NULL DEFAULT NOW()
And are enrolled in RLS with FORCE (see infra/postgres/init.sql).
"""

import uuid
from datetime import UTC, datetime

from sqlalchemy import DateTime, Float, ForeignKey, Integer, String, Text
from sqlalchemy.dialects.postgresql import JSONB, UUID
from sqlalchemy.orm import DeclarativeBase, Mapped, mapped_column, relationship


class Base(DeclarativeBase):
    """Base for all ORM models."""


def utcnow() -> datetime:
    return datetime.now(UTC)


# ---------------------------------------------------------------------------
# Tenant (control table — NOT enrolled in RLS)
# ---------------------------------------------------------------------------


class Tenant(Base):
    __tablename__ = "tenants"

    id: Mapped[uuid.UUID] = mapped_column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    name: Mapped[str] = mapped_column(String(255), unique=True, nullable=False)
    created_at: Mapped[datetime] = mapped_column(DateTime(timezone=True), default=utcnow)
    updated_at: Mapped[datetime] = mapped_column(
        DateTime(timezone=True), default=utcnow, onupdate=utcnow
    )


# ---------------------------------------------------------------------------
# User
# ---------------------------------------------------------------------------


class User(Base):
    __tablename__ = "users"

    id: Mapped[uuid.UUID] = mapped_column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    tenant_id: Mapped[uuid.UUID] = mapped_column(
        UUID(as_uuid=True), ForeignKey("tenants.id"), nullable=False, index=True
    )
    email: Mapped[str] = mapped_column(String(255), nullable=False)
    hashed_password: Mapped[str] = mapped_column(String(255), nullable=False)
    roles: Mapped[list] = mapped_column(JSONB, nullable=False, default=lambda: ["user"])
    created_at: Mapped[datetime] = mapped_column(DateTime(timezone=True), default=utcnow)
    updated_at: Mapped[datetime] = mapped_column(
        DateTime(timezone=True), default=utcnow, onupdate=utcnow
    )

    tenant: Mapped["Tenant"] = relationship("Tenant")


# ---------------------------------------------------------------------------
# Audit Log
# ---------------------------------------------------------------------------


class AuditLog(Base):
    __tablename__ = "audit_log"

    id: Mapped[uuid.UUID] = mapped_column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    session_id: Mapped[uuid.UUID] = mapped_column(UUID(as_uuid=True), nullable=False, index=True)
    user_id: Mapped[uuid.UUID] = mapped_column(UUID(as_uuid=True), nullable=False)
    tenant_id: Mapped[uuid.UUID] = mapped_column(
        UUID(as_uuid=True), ForeignKey("tenants.id"), nullable=False, index=True
    )
    input_prompt_hash: Mapped[str] = mapped_column(String(64), nullable=False)  # SHA-256
    generated_sql: Mapped[str | None] = mapped_column(Text, nullable=True)
    model_name: Mapped[str] = mapped_column(String(100), nullable=False)
    model_version: Mapped[str] = mapped_column(String(50), nullable=False)
    input_tokens: Mapped[int] = mapped_column(Integer, nullable=False, default=0)
    output_tokens: Mapped[int] = mapped_column(Integer, nullable=False, default=0)
    latency_ms: Mapped[float] = mapped_column(nullable=False, default=0.0)
    feedback_score: Mapped[int | None] = mapped_column(Integer, nullable=True)
    hallucination_flag: Mapped[bool] = mapped_column(nullable=False, default=False)
    created_at: Mapped[datetime] = mapped_column(DateTime(timezone=True), default=utcnow)

    tenant: Mapped["Tenant"] = relationship("Tenant")


# ---------------------------------------------------------------------------
# Conversation
```

```python
# ----------------------------------------------------------------------


class Conversation(Base):
    __tablename__ = "conversations"

    id: Mapped[uuid.UUID] = mapped_column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    user_id: Mapped[uuid.UUID] = mapped_column(UUID(as_uuid=True), nullable=False)
    tenant_id: Mapped[uuid.UUID] = mapped_column(
        UUID(as_uuid=True), ForeignKey("tenants.id"), nullable=False, index=True
    )
    title: Mapped[str | None] = mapped_column(String(500), nullable=True)
    created_at: Mapped[datetime] = mapped_column(DateTime(timezone=True), default=utcnow)
    updated_at: Mapped[datetime] = mapped_column(
        DateTime(timezone=True), default=utcnow, onupdate=utcnow
    )

    tenant: Mapped["Tenant"] = relationship("Tenant")


class Message(Base):
    """One chat turn (Phase 14 multi-turn history). RLS tenant-scoped."""

    __tablename__ = "messages"

    id: Mapped[uuid.UUID] = mapped_column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    conversation_id: Mapped[uuid.UUID] = mapped_column(
        UUID(as_uuid=True), ForeignKey("conversations.id"), nullable=False, index=True
    )
    tenant_id: Mapped[uuid.UUID] = mapped_column(
        UUID(as_uuid=True), ForeignKey("tenants.id"), nullable=False, index=True
    )
    role: Mapped[str] = mapped_column(String(20), nullable=False)
    content: Mapped[str] = mapped_column(Text, nullable=False)
    generated_sql: Mapped[str | None] = mapped_column(Text, nullable=True)
    created_at: Mapped[datetime] = mapped_column(DateTime(timezone=True), default=utcnow)


class Report(Base):
    """A generated multi-chart report (Phase 16). RLS tenant-scoped."""

    __tablename__ = "reports"

    id: Mapped[uuid.UUID] = mapped_column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    tenant_id: Mapped[uuid.UUID] = mapped_column(
        UUID(as_uuid=True), ForeignKey("tenants.id"), nullable=False, index=True
    )
    user_id: Mapped[uuid.UUID] = mapped_column(UUID(as_uuid=True), nullable=False, index=True)
    prompt: Mapped[str] = mapped_column(Text, nullable=False)
    title: Mapped[str] = mapped_column(String(500), nullable=False)
    summary: Mapped[str | None] = mapped_column(Text, nullable=True)
    status: Mapped[str] = mapped_column(String(20), nullable=False, default="complete")
    created_at: Mapped[datetime] = mapped_column(DateTime(timezone=True), default=utcnow)


class ReportSection(Base):
    """One metric section of a report: query + chart + narrative."""

    __tablename__ = "report_sections"

    id: Mapped[uuid.UUID] = mapped_column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    report_id: Mapped[uuid.UUID] = mapped_column(
        UUID(as_uuid=True), ForeignKey("reports.id"), nullable=False, index=True
    )
    tenant_id: Mapped[uuid.UUID] = mapped_column(
        UUID(as_uuid=True), ForeignKey("tenants.id"), nullable=False, index=True
    )
    position: Mapped[int] = mapped_column(Integer, nullable=False, default=0)
    metric_name: Mapped[str] = mapped_column(String(255), nullable=False)
    section_title: Mapped[str] = mapped_column(String(500), nullable=False)
    chart_spec: Mapped[dict] = mapped_column(JSONB, nullable=False, default=dict)
    chart_svg: Mapped[str | None] = mapped_column(Text, nullable=True)
    data_total: Mapped[float | None] = mapped_column(Float, nullable=True)
    row_count: Mapped[int] = mapped_column(Integer, nullable=False, default=0)
    narrative: Mapped[str | None] = mapped_column(Text, nullable=True)
    created_at: Mapped[datetime] = mapped_column(DateTime(timezone=True), default=utcnow)


# ----------------------------------------------------------------------
# Schema Embeddings (NL2SQL semantic search — one row per table)
# ----------------------------------------------------------------------


class SchemaEmbedding(Base):
    __tablename__ = "schema_embeddings"

    id: Mapped[uuid.UUID] = mapped_column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    tenant_id: Mapped[uuid.UUID] = mapped_column(
        UUID(as_uuid=True), ForeignKey("tenants.id"), nullable=False, index=True
    )
    table_schema: Mapped[str] = mapped_column(String(255), nullable=False, default="public")
    table_name: Mapped[str] = mapped_column(String(255), nullable=False)
    full_name: Mapped[str] = mapped_column(String(512), nullable=False)
    table_description: Mapped[str] = mapped_column(Text, nullable=False, default="")
    columns_json: Mapped[list] = mapped_column(JSONB, nullable=False, default=list)
    embedding_text: Mapped[str] = mapped_column(Text, nullable=False, default="")
    # embedding column (pgvector VECTOR(1536)) is declared via Index/DDL only;
    # ORM-level vector typing is intentionally omitted to avoid a hard dep on
    # pgvector's Python package in pure-Python test contexts.
    created_at: Mapped[datetime] = mapped_column(DateTime(timezone=True), default=utcnow)
    updated_at: Mapped[datetime] = mapped_column(
        DateTime(timezone=True), default=utcnow, onupdate=utcnow
    )

    tenant: Mapped["Tenant"] = relationship("Tenant")
```

```python


# ----------------------------------------------------------------------------
# Agent Examples (few-shot prompts)
# ----------------------------------------------------------------------------


class AgentExample(Base):
    __tablename__ = "agent_examples"

    id: Mapped[uuid.UUID] = mapped_column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    agent_name: Mapped[str] = mapped_column(String(100), nullable=False, index=True)
    nl_query: Mapped[str] = mapped_column(Text, nullable=False)
    expected_sql: Mapped[str] = mapped_column(Text, nullable=False)
    tags: Mapped[list] = mapped_column(JSONB, nullable=False, default=list)
    tenant_id: Mapped[uuid.UUID] = mapped_column(
        UUID(as_uuid=True), ForeignKey("tenants.id"), nullable=False, index=True
    )
    created_at: Mapped[datetime] = mapped_column(DateTime(timezone=True), default=utcnow)

    tenant: Mapped["Tenant"] = relationship("Tenant")
```

backend/app/db/session.py

57 lines · 1.4 KB

```python
"""Database session and engine management."""

from sqlalchemy.ext.asyncio import AsyncSession, async_sessionmaker, create_async_engine

from app.core.config import settings

engine = create_async_engine(
    settings.DATABASE_URL,
    echo=settings.DEBUG,
    pool_size=20,
    max_overflow=10,
    pool_pre_ping=True,
)

async_session = async_sessionmaker(
    engine,
    class_=AsyncSession,
    expire_on_commit=False,
)

# Login-path engine: genbi_auth role — SELECT on `users` across tenants
# (users_login_lookup policy) and nothing else. See ADR 006.
auth_engine = create_async_engine(
    settings.database_url_auth,
    echo=settings.DEBUG,
    pool_size=5,
    max_overflow=2,
    pool_pre_ping=True,
)

auth_session = async_sessionmaker(
    auth_engine,
    class_=AsyncSession,
    expire_on_commit=False,
)


async def get_db() -> AsyncSession:
    """FastAPI dependency: yield an async database session."""
    async with async_session() as session:
        try:
            yield session
            await session.commit()
        except Exception:
            await session.rollback()
            raise


async def get_auth_db() -> AsyncSession:
    """FastAPI dependency: session on the login-only genbi_auth role."""
    async with auth_session() as session:
        try:
            yield session
            await session.commit()
        except Exception:
            await session.rollback()
            raise
```

backend/app/main.py

88 lines · 2.8 KB

```python
"""GenBI application factory and server entry point."""

from contextlib import asynccontextmanager, suppress

from fastapi import FastAPI, Response
from fastapi.middleware.cors import CORSMiddleware
from prometheus_client import CONTENT_TYPE_LATEST, generate_latest

from app.api.v1.router import api_router
from app.core.config import settings
from app.core.logging import logger, setup_logging


@asynccontextmanager
async def lifespan(app: FastAPI):
    """Startup and shutdown events."""
    setup_logging()

    # Observability: tracing, metrics, and auto-instrumentation. Each is guarded
    # so a missing collector/exporter never crashes boot — they degrade to logs.
    tracer_provider = None
    meter_provider = None
    try:
        from app.core.observability import init_metrics, init_tracing, instrument_app

        tracer_provider = init_tracing()
        meter_provider = init_metrics()
        instrument_app(app)
        logger.info("Observability wired: OTel tracing + metrics + instrumentation")
    except Exception as e:
        logger.warning(f"Observability init skipped (non-fatal): {e}")

    yield

    # Shutdown: flush traces and shut down providers cleanly.
    try:
        from app.core.observability import get_langfuse_tracer

        await get_langfuse_tracer().flush()
    except Exception as e:
        logger.debug(f"Langfuse flush failed (non-fatal): {e}")
    if tracer_provider is not None:
        with suppress(Exception):
            tracer_provider.shutdown()
    if meter_provider is not None:
        with suppress(Exception):
            meter_provider.shutdown()


def create_app() -> FastAPI:
    """Create and configure the FastAPI application."""
    app = FastAPI(
        title="GenBI",
        description="Generative BI Platform API",
        version="0.1.0",
        lifespan=lifespan,
    )

    # CORS
    app.add_middleware(
        CORSMiddleware,
        allow_origins=settings.cors_origins_list,
        allow_credentials=True,
        allow_methods=["*"],
        allow_headers=["*"],
    )

    # Routes
    app.include_router(api_router, prefix="/api/v1")

    # Governance: every pipeline LLM call lands an audit_log row (fail-open).
    from app.core.llm_client import get_llm_client
    from app.services.audit import write_audit_entry

    get_llm_client().set_audit_callback(write_audit_entry)

    # Prometheus /metrics endpoint. Served on the same port as the app so
    # infra/prometheus/prometheus.yml (which scrapes backend:8000) works without
    # a separate metrics port. OTel's PrometheusMetricReader is also active when
    # init_metrics() runs, but this route is the scrape target.
    @app.get("/metrics", include_in_schema=False)
    async def metrics():
        return Response(generate_latest(), media_type=CONTENT_TYPE_LATEST)

    return app


app = create_app()
```

backend/app/models/__init__.py

10 lines · 0.2 KB

```python
"""Pydantic models — request/response schemas for the GenBI API."""

from pydantic import BaseModel


class ErrorDetail(BaseModel):
    """Standard error detail returned in all error responses."""

    code: str
    message: str
```

backend/app/models/chart.py

36 lines · 1.2 KB

```python
"""Chart request/response schemas — Flint MCP integration."""

from pydantic import BaseModel, Field


class ChartAssemblyInput(BaseModel):
    """Unified chart specification following the Flint ChartAssemblyInput schema."""

    chartType: str = Field(
        ..., description="Chart type, e.g. 'Bar Chart', 'Line Chart', 'Scatter Chart'"
    )
    encodings: dict = Field(..., description="Field-to-channel mappings: x, y, color, size, etc.")
    baseSize: dict = Field(default_factory=lambda: {"width": 600, "height": 400})
    semantic_types: dict[str, str] = Field(
        default_factory=dict, description="Field-level semantic types: Category, Quantity, Temporal"
    )
    data: dict = Field(..., description="Inline data values or file URL")


class ChartRenderRequest(BaseModel):
    """Request to render a chart via Flint MCP."""

    spec: ChartAssemblyInput
    backend: str = Field(default="vegalite", description="vegalite, echarts, or chartjs")
    format: str = Field(default="png", description="png or svg")


class ChartRenderResponse(BaseModel):
    """Response from Flint chart rendering."""

    success: bool
    format: str
    image_base64: str | None = None
    svg: str | None = None
    warnings: list[str] = []
    errors: list[str] = []
```

backend/app/models/chat.py

38 lines · 1.2 KB

```python
"""Chat request/response schemas."""

from pydantic import BaseModel, Field


class ChatRequest(BaseModel):
    """Natural language query from the user."""

    query: str = Field(
        ..., min_length=1, max_length=5000, description="The user's natural language question"
    )
    conversation_id: str | None = Field(
        None, description="Optional conversation ID for multi-turn context"
    )
    confirm_large_query: bool = Field(
        False,
        description="Explicit confirmation to run a query whose EXPLAIN row "
        "estimate exceeds the large-query threshold (>1M rows)",
    )


class ChatResponse(BaseModel):
    """Response from the NL → SQL → Chart → Narrative pipeline."""

    conversation_id: str
    query: str
    sql: str | None = None
    sql_explanation: str | None = None
    chart_spec: dict | None = None
    narrative: str | None = None
    chart_image_base64: str | None = None
    warnings: list[str] = []
    requires_confirmation: bool = Field(
        False, description="True when the query needs explicit confirmation (>1M estimated rows)"
    )
    row_estimate: int | None = Field(
        None, description="EXPLAIN-estimated row count, when available"
    )
```

backend/app/semantic/__init__.py

19 lines · 0.4 KB

```python
"""Semantic layer integration — Cube.dev client and schema embedding utilities."""

from app.semantic.cube_client import (
    CubeClient,
    CubeMetaResponse,
    MetricDefinition,
    MetricQueryResult,
    MetricType,
    get_cube_client,
)

__all__ = [
    "CubeClient",
    "CubeMetaResponse",
    "MetricDefinition",
    "MetricQueryResult",
    "MetricType",
    "get_cube_client",
]
```

backend/app/semantic/cube_client.py

811 lines · 29.0 KB

```python
"""Cube.dev semantic layer client.

Connects GenBI's agent pipeline to the Cube.dev metrics API, the single
source of truth for all business metric definitions. Every metric-aware
feature in GenBI routes through this client:

- NL2SQLAgent: resolves metric names to underlying SQL dimensions/measures
- ChartGenAgent: retrieves metric metadata for axis label formatting
- API routes (/api/v1/metrics): lists metrics and executes metric queries

Cube.dev exposes two primary APIs:
- REST API (legacy): POST /cubejs-api/v1/load for queries, /meta for schema
- GraphQL API (v2): richer query semantics, recommended path

This client uses the REST API for maximum compatibility and adds GraphQL
support as an optional upgrade path.

Architecture:
    Agent Pipeline → CubeClient.get_metric("revenue_total")
      → Cache check (Redis, TTL 5min)
        → Cache miss: HTTP POST to Cube REST API
          → Cache hit: return cached definition
            → Format for LLM context injection

Metrics are NEVER re-implemented in agent code — always query via this client.
"""

import contextlib
import json
import re
from dataclasses import dataclass, field
from datetime import UTC, datetime
from enum import StrEnum
from typing import Any

import httpx

from app.core.config import settings
from app.core.logging import logger

# Above this many metrics, get_agent_context keyword-ranks against the query
# instead of injecting the full catalog into the LLM prompt.
MAX_AGENT_CONTEXT_METRICS = 20

# ---------------------------------------------------------------------------
# Types
# ---------------------------------------------------------------------------


class MetricType(StrEnum):
    """Cube.dev metric types mapped from dbt MetricFlow."""

    SUM = "sum"
    COUNT = "count"
    COUNT_DISTINCT = "count_distinct"
    AVG = "avg"
    MIN = "min"
    MAX = "max"
    RATIO = "ratio"
    DERIVED = "derived"
    RUNNING_TOTAL = "running_total"


class DimensionType(StrEnum):
    """Cube.dev dimension types."""

    STRING = "string"
    NUMBER = "number"
    TIME = "time"
    BOOLEAN = "boolean"
    GEO = "geo"


@dataclass
class MetricDefinition:
    """Resolved metric definition from Cube.dev meta API.

    These are injected into agent prompts as structured context
    so the LLM knows what metrics are available and how to use them.
    """

    name: str
    title: str
    description: str
    metric_type: MetricType
    # The underlying Cube measure/dimension this metric references
    cube_name: str
    measure_name: str
    # Dimensions available for grouping with this metric
    dimensions: list[str] = field(default_factory=list)
    # Time dimensions for time-series queries
    time_dimensions: list[str] = field(default_factory=list)
    # Format metadata (currency, percent, decimal places)
    format: dict | None = None


@dataclass
class MetricQueryResult:
    """Result of executing a metric query against Cube.dev."""

```

```
    metric_name: str
    query: dict  # The Cube query that was sent
    data: list[dict]  # Flattened result rows
    annotation: dict  # Cube annotation metadata (formatting, types)
    total: float | None = None
    cached: bool = False
    query_latency_ms: float = 0


@dataclass
class CubeMetaResponse:
    """Parsed response from Cube.dev /meta endpoint."""

    cubes: list[dict] = field(default_factory=list)
    metrics: dict[str, MetricDefinition] = field(default_factory=dict)
    raw: dict = field(default_factory=dict)


# ---------------------------------------------------------------------------
# Client
# ---------------------------------------------------------------------------


class CubeClient:
    """Async HTTP client for Cube.dev REST API.

    Usage:
        client = CubeClient()

        # List all metrics (useful for autocomplete and context injection)
        metrics = await client.list_metrics()
        # → {"revenue_total": MetricDefinition(...), ...}

        # Resolve a single metric's definition
        metric = await client.get_metric("revenue_total")

        # Execute a metric query with dimensional breakdown
        result = await client.query(
            metrics=["revenue_total"],
            dimensions=["region"],
            time_dimensions=[{"dimension": "transaction_date", "granularity": "month"}],
            filters=[{"member": "region", "operator": "equals", "values": ["North"]}],
        )

        # Format metric definitions for LLM prompt context
        context = client.format_metrics_for_llm(metrics)
        # → "**revenue_total**: Sum of all recognized revenue (sum of revenue_amount)"

        # Health check — verify Cube is reachable
        ok = await client.health_check()
    """

    def __init__(
        self,
        api_url: str | None = None,
        api_secret: str | None = None,
        cache_ttl_seconds: int = 300,
    ):
        """Initialize the Cube client.

        Args:
            api_url: Cube REST API URL. Defaults to settings.CUBE_API_URL.
            api_secret: Cube API secret for JWT auth. Defaults to settings.CUBE_API_SECRET.
            cache_ttl_seconds: How long to cache metric definitions (default 5 min).
        """
        self.api_url = (api_url or settings.CUBE_API_URL).rstrip("/")
        # Explicit "" means "no secret" (anonymous); only None falls back to
        # settings — `or` would swallow the explicit empty string.
        self.api_secret = api_secret if api_secret is not None else settings.CUBE_API_SECRET
        self.cache_ttl = cache_ttl_seconds

        # In-memory cache for metric definitions
        self._metric_cache: dict[str, tuple[MetricDefinition, float]] = {}
        self._meta_cache: tuple[CubeMetaResponse, float] | None = None

        # Redis cache (if available)
        self._redis = None
        self._redis_available = False

        # Cached JWTs for the Authorization header: tenant_id -> (token, refresh_at_epoch)
        self._auth_token_cache: dict[str | None, tuple[str, float]] = {}

    async def _ensure_redis(self) -> None:
        """Lazy-init Redis connection."""
        if self._redis is not None:
            return

        try:
            import redis.asyncio as aioredis

            self._redis = aioredis.from_url(
                settings.REDIS_URL,
                decode_responses=False,
                socket_connect_timeout=3,
            )
            await self._redis.ping()
            self._redis_available = True
            logger.info("CubeClient: Redis cache connected")
        except Exception as e:
            logger.warning(f"CubeClient: Redis unavailable, using in-memory cache only — {e}")
            self._redis = None
            self._redis_available = False

    # ---------------------------------------------------------------------
    # HTTP transport
```

```
    # ----------------------------------------------------------------

    async def _request(
        self,
        path: str,
        method: str = "GET",
        json_body: dict | None = None,
        tenant_id: str | None = None,
    ) -> dict:
        """Make an authenticated HTTP request to Cube.dev.

        Cube requires a JWT signed with the API secret (HS256) in the
        Authorization header. The raw secret only works in dev mode; the
        signed token works in both dev and production mode.

        Args:
            path: API path relative to api_url (e.g. "/meta", "/load").
            method: HTTP method.
            json_body: Request body for POST requests.
            tenant_id: Optional tenant claim for data queries — Cube surfaces
                JWT claims as securityContext, which drives the per-tenant
                driver GUC (see semantic/cube/cube.js). Metadata requests
                omit it.

        Returns:
            Parsed JSON response body.

        Raises:
            httpx.HTTPError: On HTTP errors.
            ConnectionError: If Cube is unreachable.
        """
        url = f"{self.api_url}/{path.lstrip('/')}"

        headers: dict[str, str] = {
            "Content-Type": "application/json",
        }
        token = self._auth_token(tenant_id=tenant_id)
        if token:
            headers["Authorization"] = token

        async with httpx.AsyncClient(timeout=30.0) as client:
            if method == "GET":
                response = await client.get(url, headers=headers)
            else:
                response = await client.post(url, headers=headers, json=json_body)

            response.raise_for_status()
            return response.json()

    def _auth_token(self, tenant_id: str | None = None) -> str | None:
        """JWT for Cube's Authorization header, signed with the API secret.

        Data queries pass tenant_id so Cube's securityContext carries the
        tenant claim that selects the per-tenant driver (and its RLS GUC).
        Tokens are cached per tenant until shortly before expiry. Returns
        None when no secret is configured.
        """
        if not self.api_secret:
            return None

        now = datetime.now(UTC).timestamp()
        cached = self._auth_token_cache.get(tenant_id)
        if cached and now < cached[1]:
            return cached[0]

        from jose import jwt

        claims: dict = {"exp": int(now + 3600)}
        if tenant_id:
            claims["tenantId"] = tenant_id

        token = jwt.encode(claims, self.api_secret, algorithm="HS256")
        # Refresh 5 minutes before expiry.
        self._auth_token_cache[tenant_id] = (token, now + 3600 - 300)
        return token

    # ----------------------------------------------------------------
    # Meta API — schema discovery
    # ----------------------------------------------------------------

    async def get_meta(self, force_refresh: bool = False) -> CubeMetaResponse:
        """Fetch the full Cube metadata schema (/meta endpoint).

        The meta response includes all cubes, measures, dimensions,
        segments, and pre-aggregations — everything needed to build
        metric definitions for LLM context.

        Results are cached in Redis (TTL: cache_ttl) and in-memory.

        Args:
            force_refresh: If True, bypass cache and re-fetch.

        Returns:
            Parsed CubeMetaResponse with cubes list and metric definitions map.
        """
        # Check cache. The RAW /meta response is cached (not the parsed
        # CubeMetaResponse) and re-parsed on hit — serializing the parsed
        # form drops the metrics map.
        if not force_refresh:
            cached = await self._get_cached("cube:meta")
            if isinstance(cached, dict) and "cubes" in cached:
                logger.debug("CubeClient: meta cache hit")
                meta = self._parse_meta(cached)
                self._meta_cache = (meta, datetime.now(UTC).timestamp())
                return meta
```

```python

            if self._meta_cache is not None:
                meta, cached_at = self._meta_cache
                age = datetime.now(UTC).timestamp() - cached_at
                if age < self.cache_ttl:
                    return meta

        logger.info("CubeClient: fetching meta from Cube API")
        raw = await self._request("/meta")

        # Parse cubes
        meta = self._parse_meta(raw)

        # Cache the raw response for re-parse on hit
        await self._set_cached("cube:meta", raw)
        self._meta_cache = (meta, datetime.now(UTC).timestamp())

        return meta

    def _parse_meta(self, raw: dict) -> CubeMetaResponse:
        """Parse Cube /meta response into structured form.

        Cube's meta response format:
        {
            "cubes": [
                {
                    "name": "revenue",
                    "title": "Revenue",
                    "measures": [{"name": "revenue_amount", "title": "Revenue Amount", ...}],
                    "dimensions": [{"name": "region", "title": "Region", "type": "string"}, ...],
                    "segments": [...]
                }
            ]
        }
        """
        cubes = raw.get("cubes", [])
        metrics: dict[str, MetricDefinition] = {}

        for cube in cubes:
            cube_name = cube.get("name", "")

            # Measures → MetricDefinitions
            for measure in cube.get("measures", []):
                measure_name = measure.get("name", "")
                # Build metric key: cube.measure or just measure
                key = f"{cube_name}.{measure_name}"

                metric = MetricDefinition(
                    name=key,
                    title=measure.get("title", measure_name),
                    description=measure.get("description", ""),
                    metric_type=self._infer_metric_type(measure.get("type", "sum")),
                    cube_name=cube_name,
                    measure_name=measure_name,
                    dimensions=[
                        d.get("name", "")
                        for d in cube.get("dimensions", [])
                        if d.get("type") != "time"
                    ],
                    time_dimensions=[
                        d.get("name", "")
                        for d in cube.get("dimensions", [])
                        if d.get("type") == "time"
                    ],
                    format=measure.get("format"),
                )

                # Also index by just the measure name for convenience
                metrics[key] = metric
                if measure_name not in metrics:
                    metrics[measure_name] = metric

        return CubeMetaResponse(cubes=cubes, metrics=metrics, raw=raw)

    def _infer_metric_type(self, cube_type: str) -> MetricType:
        """Map Cube measure types to MetricType enum."""
        type_map = {
            "sum": MetricType.SUM,
            "count": MetricType.COUNT,
            "countDistinct": MetricType.COUNT_DISTINCT,
            "countDistinctApprox": MetricType.COUNT_DISTINCT,
            "avg": MetricType.AVG,
            "min": MetricType.MIN,
            "max": MetricType.MAX,
            "number": MetricType.SUM,
            "runningTotal": MetricType.RUNNING_TOTAL,
        }
        return type_map.get(cube_type, MetricType.SUM)

    # ----------------------------------------------------------------
    # Metric resolution
    # ----------------------------------------------------------------

    async def list_metrics(self, force_refresh: bool = False) -> dict[str, MetricDefinition]:
        """List all available metrics from the semantic layer.

        Use this to populate the LLM context with available metrics,
        drive the frontend metric picker, and support autocomplete.

        Returns:
            Dict mapping metric name → MetricDefinition.
        """
        meta = await self.get_meta(force_refresh=force_refresh)
        return meta.metrics

```

```
    async def get_metric(self, name: str) -> MetricDefinition | None:
        """Resolve a single metric by name.

        Args:
            name: Metric name (e.g. "revenue_total", "revenue.revenue_amount").

        Returns:
            MetricDefinition or None if not found.
        """
        # Check cache
        cache_key = f"cube:metric:{name}"
        cached = await self._get_cached(cache_key)
        if cached is not None:
            return MetricDefinition(**cached) if isinstance(cached, dict) else cached

        # Check in-memory cache
        if name in self._metric_cache:
            metric, cached_at = self._metric_cache[name]
            age = datetime.now(UTC).timestamp() - cached_at
            if age < self.cache_ttl:
                return metric

        # Fetch
        metrics = await self.list_metrics()
        metric = metrics.get(name)

        if metric:
            self._metric_cache[name] = (metric, datetime.now(UTC).timestamp())
            await self._set_cached(cache_key, metric)

        return metric

    # ----------------------------------------------------------------
    # Metric queries
    # ----------------------------------------------------------------

    async def query(
        self,
        metrics: list[str],
        dimensions: list[str] | None = None,
        time_dimensions: list[dict] | None = None,
        filters: list[dict] | None = None,
        order: list[list[str]] | None = None,
        limit: int = 1000,
        offset: int = 0,
        timezone: str = "UTC",
        tenant_id: str | None = None,
    ) -> MetricQueryResult:
        """Execute a metric query against Cube.dev.

        Args:
            metrics: List of metric names to compute.
            dimensions: Optional list of dimension names to group by.
            time_dimensions: Optional list of time dimension specs:
                [{"dimension": "transaction_date", "granularity": "month", "dateRange": ["2026-01-01", "2026-06-30"]}]
            filters: Optional list of filter specs:
                [{"member": "region", "operator": "equals", "values": ["North"]}]
            order: Sort order: [["revenue_total", "desc"]].
            limit: Max rows.
            offset: Pagination offset.
            timezone: Timezone for time dimension calculations.
            tenant_id: Tenant claim for the request JWT — drives Cube's
                per-tenant driver and its RLS GUC. Data queries under RLS
                return zero rows without it.

        Returns:
            MetricQueryResult with flattened data and metadata.
        """
        import time

        start_time = time.time()

        # Build the Cube query
        cube_query: dict[str, Any] = {
            "measures": metrics,
            "limit": limit,
            "offset": offset,
            "timezone": timezone,
        }

        if dimensions:
            cube_query["dimensions"] = dimensions

        if time_dimensions:
            cube_query["timeDimensions"] = time_dimensions

        if filters:
            cube_query["filters"] = filters

        if order:
            cube_query["order"] = {order_item[0]: order_item[1] for order_item in order}

        logger.info(
            "CubeClient: executing metric query",
            metrics=metrics,
            dimensions=dimensions,
        )

        try:
            raw = await self._request(
                "/load", method="POST", json_body={"query": cube_query}, tenant_id=tenant_id
            )

            # Cube /load response:
            # {
```

```
            #   "query": {...},
            #   "data": [{"revenue.revenue_amount": 120000, "revenue.region": "North"}, ...],
            #   "annotation": {"measures": {...}, "dimensions": {...}, ...}
            # }

            data = raw.get("data", [])
            annotation = raw.get("annotation", {})

            # Flatten data: strip cube prefixes from keys
            flat_data = []
            for row in data:
                flat_row = {}
                for key, value in row.items():
                    # "revenue.revenue_amount" → "revenue_amount"
                    short_key = key.split(".", 1)[-1] if "." in key else key
                    flat_row[short_key] = value
                flat_data.append(flat_row)

            # Compute total if a single metric. The raw rows carry the
            # fully-qualified "cube.measure" keys the query asked for.
            total = None
            if len(metrics) == 1 and data:
                metric_key = metrics[0]
                with contextlib.suppress(ValueError, TypeError):
                    total = sum(
                        float(row.get(metric_key, 0))
                        for row in data
                        if row.get(metric_key) is not None
                    )

            latency = (time.time() - start_time) * 1000

            return MetricQueryResult(
                metric_name=metrics[0] if len(metrics) == 1 else "multi",
                query=cube_query,
                data=flat_data,
                annotation=annotation,
                total=total,
                cached=False,
                query_latency_ms=latency,
            )

        except httpx.HTTPStatusError as e:
            logger.error(f"Cube query failed ({e.response.status_code}): {e.response.text}")
            raise
        except Exception:
            logger.exception("Cube query failed with unexpected error")
            raise

    # ----------------------------------------------------------------
    # LLM context formatting
    # ----------------------------------------------------------------

    def format_metrics_for_llm(
        self,
        metrics: dict[str, MetricDefinition] | None = None,
        metric_names: list[str] | None = None,
    ) -> str:
        """Format metric definitions for injection into LLM system prompts.

        Produces a compact, readable summary of available metrics that
        can be injected into the NL2SQL agent's user message.

        Args:
            metrics: Pre-fetched metrics dict. If None, loads from cache.
            metric_names: If provided, filter to only these metrics.

        Returns:
            Formatted string suitable for LLM context injection.
        """
        # Build the list
        if metric_names and metrics:
            selected = {name: metrics[name] for name in metric_names if name in metrics}
        else:
            selected = metrics or {}

        if not selected:
            return "No metrics available from the semantic layer."

        lines = ["## Available Metrics (from Semantic Layer)\n"]

        for name, metric in sorted(selected.items()):
            type_label = metric.metric_type.value
            dims = ", ".join(metric.dimensions[:5]) if metric.dimensions else "none"
            time_dims = ", ".join(metric.time_dimensions[:3]) if metric.time_dimensions else "none"

            lines.append(f"- **{name}** ({type_label})")
            if metric.description:
                lines.append(f"  {metric.description}")
            lines.append(f"  Cube: {metric.cube_name}, Measure: {metric.measure_name}")
            if metric.dimensions:
                lines.append(f"  Dimensions: {dims}")
            if metric.time_dimensions:
                lines.append(f"  Time dimensions: {time_dims}")
            lines.append("")  # blank line between metrics

        lines.append(
            "\nTo query these metrics, use the Cube API. "
            "Measures should be referenced as `{cube}.{measure}` "
            "(e.g. `revenue.revenue_amount`) in Cube queries."
        )

        return "\n".join(lines)

    def format_metric_for_prompt(self, name: str, metric: MetricDefinition) -> str:
```

```
        """Format a single metric definition as a compact prompt line."""
        parts = [f"**{name}**: {metric.description or metric.title}"]
        parts.append(f"(type: {metric.metric_type.value}, cube: {metric.cube_name})")
        if metric.dimensions:
            parts.append(f"— group by: {', '.join(metric.dimensions[:5])}")
        if metric.format:
            fmt = metric.format
            if fmt.get("currency"):
                parts.append(f"— format: currency ({fmt['currency']})")
            elif fmt.get("percent"):
                parts.append("— format: percent")
        return " ".join(parts)

    # ----------------------------------------------------------------
    # Health check
    # ----------------------------------------------------------------

    async def health_check(self) -> bool:
        """Verify Cube.dev is reachable and healthy.

        Returns:
            True if Cube responds to /meta within the timeout.
        """
        try:
            await self._request("/meta")
            return True
        except Exception as e:
            logger.warning(f"CubeClient health check failed: {e}")
            return False

    # ----------------------------------------------------------------
    # Redis cache
    # ----------------------------------------------------------------

    async def _get_cached(self, key: str) -> Any | None:
        """Get a value from Redis cache."""
        if not self._redis_available:
            return None

        try:
            await self._ensure_redis()
            if self._redis:
                raw = await self._redis.get(f"genbi:{key}")
                if raw:
                    # Support both JSON strings and bytes
                    data = json.loads(raw if isinstance(raw, (str, bytes)) else raw.decode("utf-8"))
                    # If this is a MetricDefinition, reconstruct it
                    if isinstance(data, dict) and "metric_type" in data:
                        data["metric_type"] = MetricType(data["metric_type"])
                        return MetricDefinition(**data)
                    return data
        except Exception:
            pass

        return None

    async def _set_cached(self, key: str, value: Any) -> None:
        """Set a value in Redis cache with TTL."""
        if not self._redis_available:
            return

        try:
            await self._ensure_redis()
            if self._redis:
                # Serialize dataclass to dict
                if isinstance(value, MetricDefinition):
                    data = {
                        "name": value.name,
                        "title": value.title,
                        "description": value.description,
                        "metric_type": value.metric_type.value,
                        "cube_name": value.cube_name,
                        "measure_name": value.measure_name,
                        "dimensions": value.dimensions,
                        "time_dimensions": value.time_dimensions,
                        "format": value.format,
                    }
                else:
                    data = value

                await self._redis.setex(
                    f"genbi:{key}",
                    self.cache_ttl,
                    json.dumps(data, default=str),
                )
        except Exception as e:
            logger.debug(f"CubeClient: cache set failed — {e}")

    # ----------------------------------------------------------------
    # Convenience: load metrics for agent context
    # ----------------------------------------------------------------

    async def get_agent_context(
        self,
        query: str | None = None,
        metric_names: list[str] | None = None,
    ) -> str:
        """Convenience method: fetch and format metrics for agent prompt injection.

        This is the primary integration point between the semantic layer
        and the agent pipeline. Called by ChatService before NL2SQLAgent runs.

        Args:
            query: The user's query (used to filter relevant metrics — future enhancement).
            metric_names: Specific metrics to include. If None, includes all.
```

```python

        Returns:
            Formatted string ready for insertion into the LLM user message.
        """
        metrics = await self.list_metrics()

        if metric_names:
            # Only include requested metrics
            return self.format_metrics_for_llm(metrics=metrics, metric_names=metric_names)

        if query and len(metrics) > MAX_AGENT_CONTEXT_METRICS:
            selected = self._rank_metrics_for_query(metrics, query)
            if selected:
                return self.format_metrics_for_llm(
                    metrics=metrics,
                    metric_names=[m.name for m in selected],
                )
            # No keyword overlap — fall through to the full list rather than
            # sending an empty catalog.

        return self.format_metrics_for_llm(metrics=metrics)

    @staticmethod
    def _rank_metrics_for_query(
        metrics: dict[str, MetricDefinition], query: str, limit: int = MAX_AGENT_CONTEXT_METRICS
    ) -> list[MetricDefinition]:
        """Keyword-rank metrics against the query text.

        Weights name/title/cube matches above description matches; ties keep
        insertion order. Returns [] when nothing matches at all.
        """
        terms = [t for t in re.split(r"[^a-z0-9]+", query.lower()) if len(t) > 2]
        if not terms:
            return []

        def score(metric: MetricDefinition) -> int:
            name = metric.name.lower()
            title = metric.title.lower()
            description = metric.description.lower()
            cube = metric.cube_name.lower()
            total = 0
            for term in terms:
                total += 3 * (term in name)
                total += 2 * (term in title)
                total += 2 * (term in cube)
                total += 1 * (term in description)
            return total

        ranked = sorted(
            (m for m in metrics.values() if score(m) > 0),
            key=score,
            reverse=True,
        )
        return ranked[:limit]

    # ----------------------------------------------------------------
    # Cleanup
    # ----------------------------------------------------------------

    async def close(self) -> None:
        """Close Redis connection if open."""
        if self._redis:
            with contextlib.suppress(Exception):
                await self._redis.close()
            self._redis = None
            self._redis_available = False


# ---------------------------------------------------------------------------
# Singleton
# ---------------------------------------------------------------------------

_cube_client: CubeClient | None = None


def get_cube_client() -> CubeClient:
    """Get or create the singleton CubeClient instance."""
    global _cube_client
    if _cube_client is None:
        _cube_client = CubeClient()
    return _cube_client
```

backend/app/services/__init__.py

0 lines · 0.0 KB

```
1|
```

backend/app/services/audit.py

124 lines · 4.7 KB

```python
"""Audit writer — persists the per-LLM-call governance trail to audit_log.

Registered as the LLMClient audit callback at app startup (main.create_app):
every successful LLM invocation in the pipeline (router, nl2sql, chart,
narrative) lands one audit_log row with real token counts, latency, model
name, and any generated SQL — the "every LLM call produces an AuditLog
entry" contract from docs/architecture-overview.md, and agent-granularity
tracing per ADR 002.

Writes go through a dedicated asyncpg connection on the RLS-bound runtime
role (genbi_app) with the tenant GUC set per row — the same isolation
pattern as the rest of the app (ADR 006). Rows whose ids are not UUIDs
(the client's "unknown"/"default" fallbacks) are skipped: better a missing
row than a broken chat response.

Everything FAILS OPEN: audit problems are logged, never raised — an audit
outage must not take down the pipeline (docs/core-services.md).
"""

import uuid

import asyncpg
from sqlalchemy.engine import make_url

from app.core.config import settings
from app.core.logging import logger

# Inline literal with $n parameters (identifiers fixed; values bound).
_AUDIT_INSERT = (
    "INSERT INTO audit_log "
    "(session_id, user_id, tenant_id, input_prompt_hash, generated_sql, "
    "model_name, model_version, input_tokens, output_tokens, latency_ms) "
    "VALUES ($1::uuid, $2::uuid, $3::uuid, $4, $5, $6, $7, $8, $9, $10)"
)


def _dsn() -> str:
    """asyncpg DSN for the runtime role (DATABASE_URL, plain driver)."""
    url = make_url(settings.DATABASE_URL)
    if url.drivername != "postgresql":
        url = url.set(drivername="postgresql")
    return url.render_as_string(hide_password=False)


def _as_uuid(value: str) -> uuid.UUID | None:
    try:
        return uuid.UUID(str(value))
    except (ValueError, TypeError, AttributeError):
        return None


async def write_audit_entry(entry: dict) -> None:
    """Persist one LLM-call audit entry. Never raises."""
    session_id = _as_uuid(entry.get("session_id", ""))
    user_id = _as_uuid(entry.get("user_id", ""))
    tenant_id = _as_uuid(entry.get("tenant_id", ""))
    if session_id is None or user_id is None or tenant_id is None:
        logger.warning(
            "Audit entry skipped — non-UUID identity fields",
            user_id=entry.get("user_id"),
            tenant_id=entry.get("tenant_id"),
        )
        return

    try:
        conn = await asyncpg.connect(_dsn())
        try:
            await conn.execute(
                "SELECT set_config('app.current_tenant_id', $1, false)",
                str(tenant_id),
            )
            async with conn.transaction():
                await conn.execute(
                    _AUDIT_INSERT,
                    str(session_id),
                    str(user_id),
                    str(tenant_id),
                    str(entry.get("input_prompt_hash", ""))[:64],
                    entry.get("generated_sql") or None,
                    str(entry.get("model_name", "unknown"))[:100],
                    str(entry.get("model_version", "unknown"))[:50],
                    int(entry.get("input_tokens", 0) or 0),
                    int(entry.get("output_tokens", 0) or 0),
                    float(entry.get("latency_ms", 0) or 0),
                )
        finally:
            await conn.close()
    except Exception as e:
        # Fail open — an audit outage must never break the chat pipeline.
        logger.error("Audit write failed (non-fatal): %s", e)


async def record_feedback(session_id: str, tenant_id: str, score: int) -> bool:
    """Attach a thumbs-up/down score to the audit rows of a chat session.

    Updates every audit_log row for the session (one per LLM call in the
    pipeline) — the feedback is about the response as a whole. Fail-open:
    returns False on any problem; the endpoint maps that to a 503.
    """
    session_uuid = _as_uuid(session_id)
```

```
    tenant_uuid = _as_uuid(tenant_id)
    if session_uuid is None or tenant_uuid is None:
        logger.warning("Feedback skipped — non-UUID session/tenant")
        return False

    try:
        conn = await asyncpg.connect(_dsn())
        try:
            await conn.execute(
                "SELECT set_config('app.current_tenant_id', $1, false)",
                str(tenant_uuid),
            )
            async with conn.transaction():
                row = await conn.execute(
                    "UPDATE audit_log SET feedback_score = $1 WHERE session_id = $2::uuid",
                    score,
                    str(session_uuid),
                )
        finally:
            await conn.close()
        return "UPDATE" in (row or "")
    except Exception as e:
        logger.warning("Feedback write failed (non-fatal): %s", e)
        return False
```

backend/app/services/chat_service.py

840 lines · 32.0 KB

```python
"""Chat service -- orchestrates the full NL -> SQL -> Chart -> Narrative pipeline.

Flow:
    User Query
      -> RouterAgent (intent classification)
        -> NL2SQLAgent (SQL generation with schema context + few-shot examples)
          -> ValidationAgent (safety check + EXPLAIN)
            -> Connector (read-only execution, cached)
              -> ChartGenAgent (Flint chart spec + render + hallucination check)
                -> NarrativeAgent (insight paragraph)
                  -> AuditLog (persist trace)

Supports both synchronous and streaming (SSE) response modes.
"""

import json
from uuid import uuid4

from app.agents.base import AgentConfig
from app.agents.registry import get_agent
from app.core.cache import _hash_sql as cache_hash_sql
from app.core.cache import get_cache
from app.core.config import settings
from app.core.logging import logger


class ChatService:
    """Orchestrates the end-to-end GenBI query pipeline.

    Each step produces output consumed by the next step.
    Every step is independently auditable.
    Errors at any step produce graceful degradation, not failure.
    """

    def __init__(self, tenant_id: str):
        self.tenant_id = tenant_id
        self.session_id = str(uuid4())

    # ----------------------------------------------------------------
    # Synchronous pipeline
    # ----------------------------------------------------------------

    async def process_query(
        self,
        query: str,
        user_id: str,
        roles: list[str],
        conversation_id: str | None = None,
        confirm_large_query: bool = False,
    ) -> dict:
        """Process a natural language query through the full pipeline.

        Returns a dict suitable for ChatResponse serialization.
        """
        conversation_id = await self._resolve_conversation(
            conversation_id=conversation_id, user_id=user_id, query=query
        )
        warnings: list[str] = []

        logger.info(
            "Pipeline started",
            session_id=self.session_id,
            user_id=user_id,
            tenant_id=self.tenant_id,
            query=query[:200],
        )

        # Multi-turn context (Phase 14): prior turns for this conversation,
        # fed into the NL2SQL prompt. Fail-open — cold start on any problem.
        history = await self._load_history(conversation_id)
        await self._persist_turn(conversation_id, "user", query)

        try:
            # Step 1: Route -- classify intent
            intent, plan = await self._step_route(query)

            # Step 2: NL2SQL -- generate query (with conversation history)
            sql, sql_warnings = await self._step_nl2sql(
                query=query,
                user_id=user_id,
                history=history,
            )
            warnings.extend(sql_warnings)

            if not sql:
                await self._persist_turn(
                    conversation_id, "assistant", "Could not generate SQL for this query"
                )
                return self._build_response(
                    conversation_id=conversation_id,
                    query=query,
                    warnings=warnings + ["Could not generate SQL for this query"],
                )

            # Step 3: Validate -- safety gate (EXPLAIN-backed when the
            # connector is reachable; fail-open otherwise)
            validation = await self._step_validate(sql=sql, roles=roles)
            warnings.extend(validation["warnings"])

            if not validation["valid"]:
```

```
                await self._persist_turn(
                    conversation_id,
                    "assistant",
                    "Generated SQL failed safety validation",
                    generated_sql=sql,
                )
                return self._build_response(
                    conversation_id=conversation_id,
                    query=query,
                    sql=sql,
                    warnings=warnings + ["Generated SQL failed safety validation"],
                )

            # Step 3b: Large-query confirmation gate (>1M estimated rows).
            # Stops before execution unless the request explicitly confirms.
            if validation["requires_confirmation"] and not confirm_large_query:
                await self._persist_turn(
                    conversation_id,
                    "assistant",
                    "Confirmation required before running this large query",
                    generated_sql=sql,
                )
                return self._build_response(
                    conversation_id=conversation_id,
                    query=query,
                    sql=sql,
                    warnings=warnings,
                    requires_confirmation=True,
                    row_estimate=validation["row_estimate"],
                )

            # Step 4: Execute -- run the query (cached)
            data, exec_warnings = await self._step_execute(
                sql=validation["validated_sql"] or sql,
            )
            warnings.extend(exec_warnings)

            if data is None:
                await self._persist_turn(
                    conversation_id,
                    "assistant",
                    "Query execution returned no data",
                    generated_sql=sql,
                )
                return self._build_response(
                    conversation_id=conversation_id,
                    query=query,
                    sql=sql,
                    warnings=warnings + ["Query execution returned no data"],
                )

            # Step 5: Chart -- generate spec + validate + render
            chart_output = await self._step_chart(
                data=data,
                query=query,
                user_id=user_id,
            )
            warnings.extend(chart_output.get("warnings", []))

            # Step 6: Narrative -- write insight
            narrative = await self._step_narrative(
                query=query,
                data=data,
                chart_context=chart_output.get("chart_spec", {}),
                user_id=user_id,
            )
            warnings.extend(narrative.get("warnings", []))

            logger.info("Pipeline complete", session_id=self.session_id, warnings=len(warnings))

            await self._persist_turn(
                conversation_id,
                "assistant",
                narrative.get("narrative") or "Here's what I found.",
                generated_sql=sql,
            )
            return self._build_response(
                conversation_id=conversation_id,
                query=query,
                sql=sql,
                chart_spec=chart_output.get("chart_spec"),
                narrative=narrative.get("narrative"),
                chart_image_base64=chart_output.get("image_base64"),
                chart_svg=chart_output.get("svg"),
                warnings=warnings,
                requires_confirmation=False,
                row_estimate=validation["row_estimate"],
            )

        except Exception as e:
            logger.error(f"Pipeline failed: {e}", session_id=self.session_id)
            await self._persist_turn(conversation_id, "assistant", f"Pipeline error: {str(e)}")
            return self._build_response(
                conversation_id=conversation_id,
                query=query,
                warnings=warnings + [f"Pipeline error: {str(e)}"],
            )

    # ----------------------------------------------------------------
    # Streaming pipeline (SSE)
    # ----------------------------------------------------------------

    async def process_query_stream(
        self,
        query: str,
```

```
        user_id: str,
        roles: list[str],
        conversation_id: str | None = None,
        confirm_large_query: bool = False,
    ):
        """Streaming variant -- yields SSE events as each pipeline stage completes.

        Events emitted:
            start -> intent -> sql -> validation -> data -> chart -> narrative -> done

        Each event includes the stage name and partial results. The frontend
        renders each stage incrementally as events arrive.
        """
        conversation_id = await self._resolve_conversation(conversation_id, user_id, query)
        warnings: list[str] = []

        def _emit(event: str, data: dict) -> str:
            return f"data: {json.dumps({'event': event, **data}, default=str)}\n\n"

        logger.info(
            "Streaming pipeline started",
            session_id=self.session_id,
            user_id=user_id,
            query=query[:200],
        )

        history = await self._load_history(conversation_id)
        await self._persist_turn(conversation_id, "user", query)

        try:
            # Start — includes the resolved conversation id so the client can
            # thread follow-up turns (Phase 14).
            yield _emit(
                "start",
                {
                    "query": query,
                    "session_id": self.session_id,
                    "conversation_id": conversation_id,
                },
            )

            # Step 1: Route
            intent, plan = await self._step_route(query)
            yield _emit("intent", {"intent": intent, "plan": plan})

            # Step 2: NL2SQL
            sql, sql_warnings = await self._step_nl2sql(
                query=query, user_id=user_id, history=history
            )
            warnings.extend(sql_warnings)
            yield _emit("sql", {"sql": sql, "warnings": sql_warnings})

            if not sql:
                await self._persist_turn(
                    conversation_id, "assistant", "Could not generate SQL for this query"
                )
                yield _emit("done", {"status": "no_sql", "warnings": warnings})
                return

            # Step 3: Validate
            validation = await self._step_validate(sql=sql, roles=roles)
            warnings.extend(validation["warnings"])
            yield _emit(
                "validation",
                {
                    "valid": validation["valid"],
                    "validated_sql": validation["validated_sql"],
                    "requires_confirmation": validation["requires_confirmation"],
                    "row_estimate": validation["row_estimate"],
                    "warnings": validation["warnings"],
                },
            )

            if not validation["valid"]:
                await self._persist_turn(
                    conversation_id,
                    "assistant",
                    "Generated SQL failed safety validation",
                    generated_sql=sql,
                )
                yield _emit("done", {"status": "validation_failed", "warnings": warnings})
                return

            # Step 3b: Large-query confirmation gate — stop before execution
            # unless the request explicitly confirms.
            if validation["requires_confirmation"] and not confirm_large_query:
                await self._persist_turn(
                    conversation_id,
                    "assistant",
                    "Confirmation required before running this large query",
                    generated_sql=sql,
                )
                yield _emit(
                    "done",
                    {
                        "status": "confirmation_required",
                        "requires_confirmation": True,
                        "row_estimate": validation["row_estimate"],
                        "warnings": warnings,
                    },
                )
                return

            # Step 4: Execute
            data, exec_warnings = await self._step_execute(sql=validation["validated_sql"] or sql)
```

```
            warnings.extend(exec_warnings)
            yield _emit(
                "data",
                {
                    "row_count": len(data) if data else 0,
                    "preview": data[:5] if data else [],
                    "warnings": exec_warnings,
                },
            )

            if not data:
                await self._persist_turn(
                    conversation_id,
                    "assistant",
                    "Query execution returned no data",
                    generated_sql=sql,
                )
                yield _emit("done", {"status": "no_data", "warnings": warnings})
                return

            # Step 5: Chart (with hallucination detection)
            chart_output = await self._step_chart(data=data, query=query, user_id=user_id)
            warnings.extend(chart_output.get("warnings", []))
            yield _emit(
                "chart",
                {
                    "chart_spec": chart_output.get("chart_spec"),
                    "image_base64": chart_output.get("image_base64"),
                    "svg": chart_output.get("svg"),
                    "warnings": chart_output.get("warnings", []),
                },
            )

            # Step 6: Narrative
            narrative = await self._step_narrative(
                query=query,
                data=data,
                chart_context=chart_output.get("chart_spec", {}),
                user_id=user_id,
            )
            warnings.extend(narrative.get("warnings", []))
            yield _emit(
                "narrative",
                {
                    "narrative": narrative.get("narrative"),
                    "warnings": narrative.get("warnings", []),
                },
            )

            # Done
            await self._persist_turn(
                conversation_id,
                "assistant",
                narrative.get("narrative") or "Here's what I found.",
                generated_sql=sql,
            )
            yield _emit("done", {"status": "complete", "warnings": warnings})

        except Exception as e:
            logger.error(f"Streaming pipeline failed: {e}", session_id=self.session_id)
            await self._persist_turn(conversation_id, "assistant", f"Pipeline error: {str(e)}")
            yield _emit("done", {"status": "error", "error": str(e), "warnings": warnings})

    # ---------------------------------------------------------------
    # Pipeline steps
    # ---------------------------------------------------------------

    async def _resolve_conversation(
        self, conversation_id: str | None, user_id: str, query: str
    ) -> str:
        """Use the given conversation id, or create one titled by the query."""
        from app.services import conversations as conversations_service

        if conversation_id:
            return conversation_id
        created = await conversations_service.create_conversation(
            user_id=user_id, tenant_id=self.tenant_id, title=query
        )
        return created or str(uuid4())

    async def _load_history(self, conversation_id: str, limit: int = 6) -> list[dict]:
        """Prior turns for the NL2SQL prompt. Fail-open (cold start)."""
        from app.services import conversations as conversations_service

        try:
            return await conversations_service.list_messages(
                conversation_id=conversation_id, tenant_id=self.tenant_id, limit=limit
            )
        except Exception as e:
            logger.warning("Conversation history unavailable — cold start: %s", e)
            return []

    async def _persist_turn(
        self,
        conversation_id: str,
        role: str,
        content: str,
        generated_sql: str | None = None,
    ) -> None:
        """Append a chat turn. Fail-open inside the conversations service."""
        from app.services import conversations as conversations_service

        await conversations_service.append_message(
            conversation_id=conversation_id,
            tenant_id=self.tenant_id,
```

```
            role=role,
            content=content,
            generated_sql=generated_sql,
        )

    async def _step_route(self, query: str) -> tuple[str, list[dict]]:
        """Step 1: Classify query intent."""
        try:
            agent_cls = get_agent("router")
            if not agent_cls:
                return "chat_data", []

            agent = agent_cls(AgentConfig(model_name=settings.LLM_FAST_MODEL))
            result = await agent.execute(query=query)
            intent = result.output.get("intent", "chat_data")
            plan = result.output.get("dispatch_plan", [])
            return intent, plan
        except Exception as e:
            logger.warning(f"RouterAgent failed, defaulting to chat_data: {e}")
            return "chat_data", []

    async def _step_nl2sql(
        self,
        query: str,
        user_id: str,
        history: list[dict] | None = None,
    ) -> tuple[str | None, list[str]]:
        """Step 2: Generate SQL from natural language.

        Schema context is cached at L1+L2 per (query_hash, tenant_id).
        Metric definitions from Cube.dev are cached at L2 per tenant.
        History (prior conversation turns) feeds the prompt for multi-turn
        queries like "now break that down by region".
        """
        try:
            agent_cls = get_agent("nl2sql")
            if not agent_cls:
                return None, ["NL2SQLAgent not registered"]

            agent = agent_cls(
                AgentConfig(
                    model_name=settings.LLM_REASONING_MODEL,
                    temperature=0,
                    max_tokens=4096,
                    thinking=True,
                )
            )

            # Load metric definitions from Cube.dev semantic layer (cached)
            metric_context = ""
            cache = get_cache()
            try:
                # Check cache for metric definitions
                cached_metrics = await cache.get_metric_definitions(self.tenant_id)
                if cached_metrics is not None:
                    metric_context = cached_metrics
                    logger.info("Metric context loaded from cache")
                else:
                    from app.semantic.cube_client import get_cube_client

                    cube = get_cube_client()
                    metric_context = await cube.get_agent_context(query=query)
                    await cache.set_metric_definitions(self.tenant_id, metric_context)
                    logger.info("Metric context loaded and cached", length=len(metric_context))
            except Exception as e:
                logger.warning(f"Metric context unavailable -- continuing without: {e}")

            # Schema context: pgvector top-k tables for this query (cached
            # per query+tenant). Fails open to [] — same contract as metrics.
            schema_context: list[dict] = []
            try:
                cached_schema = await cache.get_schema_context(query, self.tenant_id)
                if cached_schema is not None:
                    schema_context = cached_schema
                else:
                    from app.services.schema_retrieval import retrieve_schema_context

                    schema_context = await retrieve_schema_context(query, self.tenant_id)
                    await cache.set_schema_context(query, self.tenant_id, schema_context)
                    logger.info("Schema context retrieved", tables=len(schema_context))
            except Exception as e:
                logger.warning(f"Schema context unavailable -- continuing without: {e}")

            # Few-shot examples: similar validated NL/SQL pairs (cached).
            few_shot_examples: list[dict] = []
            try:
                cached_examples = await cache.get_few_shot_examples(query, self.tenant_id)
                if cached_examples is not None:
                    few_shot_examples = cached_examples
                else:
                    from app.services.schema_retrieval import retrieve_few_shot_examples

                    few_shot_examples = await retrieve_few_shot_examples(query, self.tenant_id)
                    await cache.set_few_shot_examples(query, self.tenant_id, few_shot_examples)
                    logger.info("Few-shot examples retrieved", count=len(few_shot_examples))
            except Exception as e:
                logger.warning(f"Few-shot examples unavailable -- continuing without: {e}")

            result = await agent.execute(
                query=query,
                tenant_id=self.tenant_id,
                user_id=user_id,
                session_id=self.session_id,
                metric_definitions=metric_context,
                schema_context=schema_context,
```

```
                few_shot_examples=few_shot_examples,
                history=history,
            )
            sql = result.output.get("sql")
            return sql, result.warnings
        except Exception as e:
            logger.error(f"NL2SQLAgent failed: {e}")
            return None, [f"SQL generation failed: {str(e)}"]

    async def _step_validate(
        self,
        sql: str,
        roles: list[str],
    ) -> dict:
        """Step 3: Validate SQL for safety (EXPLAIN-backed, fail-open).

        Returns {valid, warnings, validated_sql, requires_confirmation,
        row_estimate}.
        """
        fail_open = {
            "valid": True,
            "warnings": ["Validation skipped -- agent not available"],
            "validated_sql": sql,
            "requires_confirmation": False,
            "row_estimate": None,
        }
        try:
            agent_cls = get_agent("validation")
            if not agent_cls:
                logger.warning("ValidationAgent not registered -- skipping validation")
                return fail_open

            agent = agent_cls(AgentConfig(model_name="deterministic"))

            # Short-lived read connector so the EXPLAIN dry-run runs under
            # the tenant GUC (RLS-scoped estimates). Fail-open by design.
            from app.connectors.postgresql_connector import PostgreSQLConnector

            connector = PostgreSQLConnector(
                connection_url=settings.DATABASE_URL, tenant_id=self.tenant_id
            )
            async with connector:
                result = await agent.execute(
                    sql=sql,
                    tenant_id=self.tenant_id,
                    user_roles=roles,
                    connector=connector,
                )

            validated_sql = result.output.get("validated_sql", sql)

            if not result.success:
                logger.warning(
                    "SQL validation failed",
                    errors=result.errors,
                    sql=sql[:200],
                )

            return {
                "valid": result.success,
                "warnings": result.warnings,
                "validated_sql": validated_sql,
                "requires_confirmation": bool(result.output.get("requires_confirmation")),
                "row_estimate": result.output.get("row_estimate"),
            }
        except Exception as e:
            logger.error(f"ValidationAgent failed: {e}")
            return {
                "valid": False,
                "warnings": [f"Validation error: {str(e)}"],
                "validated_sql": None,
                "requires_confirmation": False,
                "row_estimate": None,
            }

    async def _step_execute(self, sql: str) -> tuple[list[dict] | None, list[str]]:
        """Step 4: Execute SQL against the data source. Cached at L1+L2.

        PII masking is applied here — the single chokepoint between the data
        source and the rest of the pipeline. Masking BEFORE caching means
        cached rows are also safe: no PII persists in Redis/L1 either. This
        fulfills the CLAUDE.md §7 guarantee that no PII enters LLM context.
        """
        try:
            # Check cache first (cached rows are already masked)
            cache = get_cache()
            cached_result = await cache.get_query_result(sql, self.tenant_id)
            if cached_result is not None:
                logger.info(
                    "Query result cache hit",
                    row_count=len(cached_result),
                    tenant_id=self.tenant_id,
                )
                return cached_result, []

            from app.connectors.postgresql_connector import PostgreSQLConnector
            from app.core.config import settings
            from app.core.masking import get_masker_for_tenant

            connector = PostgreSQLConnector(
                connection_url=settings.DATABASE_URL,
                tenant_id=self.tenant_id,
            )

            async with connector:
```

```
                results = await connector.execute(sql)
                logger.info(
                    "SQL executed",
                    row_count=len(results),
                    tenant_id=self.tenant_id,
                )

                # Mask PII before the data reaches agents or the cache. This is
                # the load-bearing line for the "no PII in LLM context" rule.
                results = get_masker_for_tenant(self.tenant_id).mask_rows(results)

                # Cache the (masked) result
                await cache.set_query_result(sql, self.tenant_id, results)

                return results, []

        except Exception as e:
            logger.error(f"SQL execution failed: {e}")
            return None, [f"Query execution failed: {str(e)}"]

    async def _step_chart(
        self,
        data: list[dict],
        query: str,
        user_id: str = "",
    ) -> dict:
        """Step 5: Generate chart spec, validate, and render via Flint/Altair bridge.

        Chart hallucination detection runs between generation and rendering.
        If the spec has errors, auto-correction is attempted before the fallback
        render path. Every correction is logged and surfaced as a warning.
        """
        try:
            agent_cls = get_agent("chart_gen")
            if not agent_cls:
                return {"warnings": ["ChartGenAgent not registered"]}

            agent = agent_cls(
                AgentConfig(
                    model_name=settings.LLM_FAST_MODEL,
                    temperature=0,
                    max_tokens=2048,
                )
            )
            result = await agent.execute(
                data=data,
                query=query,
                tenant_id=self.tenant_id,
                user_id=user_id,
            )

            if not result.success:
                return {"warnings": result.errors}

            chart_spec = result.output.get("chart_spec", {})

            # --- Hallucination detection ---
            if chart_spec and data:
                try:
                    from app.agents.validation.chart_validator import (
                        validate_chart_spec,
                    )

                    validation = validate_chart_spec(
                        spec=chart_spec,
                        data=data,
                        auto_correct=True,
                    )

                    validation_warnings = validation.warnings + validation.fix_summary

                    if validation.corrected_spec is not None:
                        logger.info(
                            "Chart spec auto-corrected",
                            corrections=validation.fix_summary,
                            original_type=chart_spec.get("chartType"),
                            corrected_type=validation.corrected_spec.get("chartType"),
                        )
                        chart_spec = validation.corrected_spec

                    if not validation.is_valid:
                        error_codes = [i.code for i in validation.issues if i.severity == "error"]
                        logger.warning(
                            "Chart spec has uncorrectable errors",
                            errors=error_codes,
                        )
                        validation_warnings.insert(
                            0,
                            f"Chart spec has uncorrectable errors: {', '.join(error_codes)}",
                        )
                    elif validation_warnings:
                        logger.info(
                            "Chart spec validated with warnings",
                            warning_count=len(validation_warnings),
                        )

                except ImportError:
                    validation_warnings = []
                    logger.debug("Chart validator not available -- skipping")
                except Exception as e:
                    validation_warnings = [f"Chart validation skipped: {e}"]
                    logger.warning(f"Chart validation exception: {e}")
            else:
                validation_warnings = []

```

```
            # --- Cache the validated chart spec ---
            if chart_spec and data:
                try:
                    cache = get_cache()
                    data_hash = cache_hash_sql(json.dumps(data[:10], default=str))
                    await cache.set_chart_spec(data_hash, self.tenant_id, chart_spec)
                except Exception:
                    pass  # Non-critical

            return {
                "chart_spec": chart_spec,
                "image_base64": result.output.get("image_base64"),
                "svg": result.output.get("svg"),
                "backend": result.output.get("backend"),
                "warnings": result.warnings + validation_warnings,
            }

        except Exception as e:
            logger.error(f"ChartGenAgent failed: {e}")
            return {"warnings": [f"Chart generation failed: {str(e)}"]}

    async def _step_narrative(
        self,
        query: str,
        data: list[dict],
        chart_context: dict,
        user_id: str,
    ) -> dict:
        """Step 6: Generate insight narrative."""
        try:
            agent_cls = get_agent("narrative")
            if not agent_cls:
                return {"warnings": ["NarrativeAgent not registered"]}

            agent = agent_cls(
                AgentConfig(
                    model_name=settings.LLM_FAST_MODEL,
                    temperature=0.3,
                    max_tokens=512,
                )
            )

            # Build data summary
            data_summary = {
                "head": data[:5],
                "row_count": len(data),
            }

            # Add numeric statistics if present
            if data:
                numeric_cols = [c for c, v in data[0].items() if isinstance(v, (int, float))]
                if numeric_cols:
                    col = numeric_cols[0]
                    values = [r[col] for r in data if r.get(col) is not None]
                    if values:
                        data_summary["total"] = sum(values)
                        data_summary["avg_value"] = sum(values) / len(values)
                        data_summary["max_value"] = max(values)
                        data_summary["min_value"] = min(values)

            result = await agent.execute(
                query=query,
                data_summary=data_summary,
                chart_context=chart_context,
                tenant_id=self.tenant_id,
                user_id=user_id,
                session_id=self.session_id,
            )

            return {
                "narrative": result.output.get("narrative"),
                "warnings": result.warnings,
            }

        except Exception as e:
            logger.error(f"NarrativeAgent failed: {e}")
            return {"warnings": [f"Narrative generation failed: {str(e)}"]}

    # ------------------------------------------------------------------
    # Helpers
    # ------------------------------------------------------------------

    def _build_response(
        self,
        conversation_id: str,
        query: str,
        sql: str | None = None,
        sql_explanation: str | None = None,
        chart_spec: dict | None = None,
        narrative: str | None = None,
        chart_image_base64: str | None = None,
        chart_svg: str | None = None,
        warnings: list[str] | None = None,
        requires_confirmation: bool = False,
        row_estimate: int | None = None,
    ) -> dict:
        """Build a ChatResponse-compatible dict."""
        return {
            "conversation_id": conversation_id,
            "query": query,
            "sql": sql,
            "sql_explanation": sql_explanation,
            "chart_spec": chart_spec,
            "narrative": narrative,
            "chart_image_base64": chart_image_base64,
```

```
            "chart_svg": chart_svg,
            "warnings": warnings or [],
            "requires_confirmation": requires_confirmation,
            "row_estimate": row_estimate,
        }
```

backend/app/services/conversations.py

139 lines · 4.6 KB

```python
"""Conversation persistence — multi-turn chat history (Phase 14).

Reads and writes go through dedicated asyncpg connections on the RLS-bound
runtime role (genbi_app) with the tenant GUC set per connection — the same
isolation pattern as the audit writer (Phase 12). Per-user scoping beyond
RLS (a tenant's users only see their own conversations) is enforced with
explicit parameterized predicates.

Failure semantics:
- writes (create/append) FAIL OPEN — persistence problems never break a chat
  response; the caller falls back to an in-memory conversation id.
- reads RAISE — callers decide (ChatService history: fail-open to []; the
  conversations API maps failures to 503).
"""

import uuid

import asyncpg
from sqlalchemy.engine import make_url

from app.core.config import settings
from app.core.logging import logger

MAX_TITLE_LEN = 500


def _dsn() -> str:
    """asyncpg DSN for the runtime role (DATABASE_URL, plain driver)."""
    url = make_url(settings.DATABASE_URL)
    if url.drivername != "postgresql":
        url = url.set(drivername="postgresql")
    return url.render_as_string(hide_password=False)


async def _connect(tenant_id: str) -> asyncpg.Connection:
    conn = await asyncpg.connect(_dsn())
    await conn.execute("SELECT set_config('app.current_tenant_id', $1, false)", tenant_id)
    return conn


async def create_conversation(user_id: str, tenant_id: str, title: str | None) -> str | None:
    """Create a conversation row. Returns its id, or None on failure."""
    conversation_id = str(uuid.uuid4())
    try:
        conn = await _connect(tenant_id)
        try:
            await conn.execute(
                "INSERT INTO conversations (id, user_id, tenant_id, title) "
                "VALUES ($1::uuid, $2::uuid, $3::uuid, $4)",
                conversation_id,
                user_id,
                tenant_id,
                (title or "New chat")[:MAX_TITLE_LEN],
            )
        finally:
            await conn.close()
        return conversation_id
    except Exception as e:
        logger.warning("Conversation create failed (non-fatal): %s", e)
        return None


async def append_message(
    conversation_id: str,
    tenant_id: str,
    role: str,
    content: str,
    generated_sql: str | None = None,
) -> None:
    """Append one chat turn and bump the conversation's updated_at. Fail-open."""
    try:
        conn = await _connect(tenant_id)
        try:
            async with conn.transaction():
                await conn.execute(
                    "INSERT INTO messages (conversation_id, tenant_id, role, content, generated_sql) "
                    "VALUES ($1::uuid, $2::uuid, $3, $4, $5)",
                    conversation_id,
                    tenant_id,
                    role,
                    content,
                    generated_sql,
                )
                await conn.execute(
                    "UPDATE conversations SET updated_at = NOW() WHERE id = $1::uuid",
                    conversation_id,
                )
        finally:
            await conn.close()
    except Exception as e:
        logger.warning("Message append failed (non-fatal): %s", e)


async def list_conversations(user_id: str, tenant_id: str, limit: int = 50) -> list[dict]:
    """The user's conversations, most recently active first."""
    conn = await _connect(tenant_id)
    try:
        rows = await conn.fetch(
            "SELECT id, title, created_at, updated_at "
            "FROM conversations WHERE user_id = $1::uuid "
```

```
            "ORDER BY updated_at DESC LIMIT $2",
            user_id,
            limit,
        )
    finally:
        await conn.close()
    return [
        {
            "id": str(row["id"]),
            "title": row["title"] or "New chat",
            "created_at": row["created_at"].isoformat(),
            "updated_at": row["updated_at"].isoformat(),
        }
        for row in rows
    ]


async def list_messages(conversation_id: str, tenant_id: str, limit: int = 20) -> list[dict]:
    """The conversation's most recent turns, in chronological order."""
    conn = await _connect(tenant_id)
    try:
        rows = await conn.fetch(
            "SELECT role, content, generated_sql, created_at "
            "FROM messages WHERE conversation_id = $1::uuid "
            "ORDER BY created_at DESC LIMIT $2",
            conversation_id,
            limit,
        )
    finally:
        await conn.close()
    return [
        {
            "role": row["role"],
            "content": row["content"],
            "generated_sql": row["generated_sql"],
            "created_at": row["created_at"].isoformat(),
        }
        for row in reversed(rows)
    ]
```

backend/app/services/reports.py

375 lines · 13.3 KB

```python
"""Report generation and persistence — multi-chart reports (Phase 16).

Pipeline (LLM-planned, deterministic execution — 2 LLM calls per report):
  1. Plan: one fast-model call over the metric catalog picks 2–4 sections
     (metric + slice dimension + title). Fail → REPORT_GENERATION_FAILED.
  2. Execute per section (fail-open): tenant-scoped Cube query →
     deterministic ChartAssemblyInput → FlintChartBridge SVG render.
  3. Summarize: one fast-model call over all section results.

Persistence follows the conversations.py pattern: asyncpg on the RLS-bound
runtime role with the tenant GUC per connection. Writes fail open (the
generated report returns even if persistence fails — with a warning);
reads raise so the API can map to 503/404.
"""

import contextlib
import json
import uuid
from datetime import UTC, datetime

import asyncpg
from sqlalchemy.engine import make_url

from app.core.config import settings
from app.core.logging import logger

MAX_TITLE_LEN = 500


def _dsn() -> str:
    """asyncpg DSN for the runtime role (DATABASE_URL, plain driver)."""
    url = make_url(settings.DATABASE_URL)
    if url.drivername != "postgresql":
        url = url.set(drivername="postgresql")
    return url.render_as_string(hide_password=False)


async def _connect(tenant_id: str) -> asyncpg.Connection:
    conn = await asyncpg.connect(_dsn())
    await conn.execute("SELECT set_config('app.current_tenant_id', $1, false)", tenant_id)
    return conn


# ---------------------------------------------------------------------------
# Generation
# ---------------------------------------------------------------------------


class ReportGenerationError(Exception):
    """Planning failed — no report could be produced."""


async def _plan_report(prompt: str, tenant_id: str, user_id: str, max_sections: int) -> dict:
    """LLM call 1: pick metrics + section structure from the catalog."""
    from app.core.llm_client import LLMCallOptions, get_llm_client, load_prompt
    from app.semantic.cube_client import get_cube_client

    catalog = await get_cube_client().get_agent_context(query=prompt)

    system = load_prompt("report-planner-system")
    user_message = (
        f"## Report Request\n\n{prompt}\n\n{catalog}\n\nPick at most {max_sections} sections."
    )

    client = get_llm_client()
    result = await client.invoke(
        messages=user_message,
        system=system,
        use_reasoning=False,
        options=LLMCallOptions(temperature=0.0, max_tokens=1024, response_format="json"),
        user_id=user_id,
        tenant_id=tenant_id,
        session_id=f"report-plan-{uuid.uuid4()}",
    )

    plan = result.parsed
    if not plan or not isinstance(plan.get("sections"), list) or not plan["sections"]:
        raise ReportGenerationError("planner returned no sections")

    # Sanitize: cap sections, truncate strings, drop unknown shapes.
    sections = []
    for raw in plan["sections"][:max_sections]:
        metric = str(raw.get("metric", "")).strip()
        if not metric:
            continue
        sections.append(
            {
                "metric": metric,
                "title": str(raw.get("title") or metric)[:200],
                "dimension": raw.get("dimension") or None,
                "granularity": raw.get("granularity") or None,
            }
        )
    if not sections:
        raise ReportGenerationError("planner returned no valid sections")

    return {"title": str(plan.get("title") or prompt)[:MAX_TITLE_LEN], "sections": sections}


def _build_chart_spec(section: dict, rows: list[dict]) -> dict:
```

```
    """Deterministic ChartAssemblyInput from a section plan + its rows."""
    metric_short = section["metric"].split(".")[-1]
    time_dim = section.get("granularity") is not None

    if time_dim and rows:
        # Trend: x = the time-granularity column, y = the metric
        x_field = next(
            (k for k in rows[0] if k.endswith(("_month", "_day")) or k in ("month", "day")),
            list(rows[0].keys())[0],
        )
        chart_type = "Line Chart"
    else:
        x_field = next(
            (k for k in rows[0] if k != metric_short),
            list(rows[0].keys())[0] if rows else "category",
        )
        chart_type = "Bar Chart"

    return {
        "chartType": chart_type,
        "encodings": {"x": {"field": x_field}, "y": {"field": metric_short}},
        "baseSize": {"width": 600, "height": 400},
        "data": {"values": rows},
    }


async def _execute_section(section: dict, tenant_id: str) -> dict | None:
    """Run the metric query + render the chart. Fail-open (None on error)."""
    from app.agents.chart.flint_bridge import FlintChartBridge
    from app.semantic.cube_client import get_cube_client

    try:
        cube = get_cube_client()
        time_dimensions = None
        dimensions = None
        if section.get("dimension") and section.get("granularity"):
            time_dimensions = [
                {"dimension": section["dimension"], "granularity": section["granularity"]}
            ]
        elif section.get("dimension"):
            dimensions = [section["dimension"]]

        result = await cube.query(
            metrics=[section["metric"]],
            dimensions=dimensions,
            time_dimensions=time_dimensions,
            limit=50,
            tenant_id=tenant_id,
        )
        if not result.data:
            return None

        spec = _build_chart_spec(section, result.data)

        svg = None
        try:
            bridge = FlintChartBridge(tenant_id=tenant_id)
            rendered = await bridge.render(spec, output_format="svg")
            if rendered.get("success"):
                svg = rendered.get("svg")
        except Exception as e:
            logger.warning("Report chart render failed (non-fatal): %s", e)

        total = result.total
        if total is None:
            with contextlib.suppress(Exception, KeyError, ValueError, TypeError):
                metric_short = section["metric"].split(".")[-1]
                nums = [
                    float(row[metric_short])
                    for row in result.data
                    if row.get(metric_short) is not None
                ]
                total = sum(nums) if nums else None

        return {
            "metric_name": section["metric"],
            "section_title": section["title"],
            "chart_spec": spec,
            "chart_svg": svg,
            "data_total": total,
            "row_count": len(result.data),
        }
    except Exception as e:
        logger.warning("Report section failed (skipped): %s — %s", section.get("metric"), e)
        return None


async def _summarize_report(prompt: str, title: str, sections: list[dict], tenant_id: str) -> str:
    """LLM call 2: one overall narrative from the section results."""
    from app.core.llm_client import get_llm_client, load_prompt

    lines = [f"Report: {title}", f"Request: {prompt}", "", "Sections:"]
    for s in sections:
        total = f"{s['data_total']:,.0f}" if s.get("data_total") is not None else "n/a"
        lines.append(f"- {s['section_title']}: {s['row_count']} rows, total {total}")

    result = await get_llm_client().invoke(
        messages="\n".join(lines),
        system=load_prompt("narrative-system"),
        use_reasoning=False,
        options=None,
        user_id=None,
        tenant_id=tenant_id,
    )
    return result.content
```

```python


async def generate_report(prompt: str, tenant_id: str, user_id: str, max_sections: int = 3) -> dict:
    """Full pipeline: plan → execute sections → summarize. Returns the report
    dict (persisted best-effort — persistence failure adds a warning)."""
    plan = await _plan_report(prompt, tenant_id, user_id, max_sections)

    sections = []
    skipped = []
    for section in plan["sections"]:
        executed = await _execute_section(section, tenant_id)
        if executed is not None:
            executed["position"] = len(sections)
            sections.append(executed)
        else:
            skipped.append(section["metric"])

    if not sections:
        raise ReportGenerationError(
            f"all planned sections returned no data (metrics: "
            f"{', '.join(s['metric'] for s in plan['sections'])})"
        )

    summary = ""
    try:
        summary = await _summarize_report(prompt, plan["title"], sections, tenant_id)
    except Exception as e:
        logger.warning("Report summary failed (non-fatal): %s", e)

    report = {
        "report_id": str(uuid.uuid4()),
        "title": plan["title"],
        "prompt": prompt,
        "summary": summary,
        "status": "complete",
        "created_at": datetime.now(UTC).isoformat(),
        "sections": sections,
        "warnings": [f"Skipped metric (no data): {m}" for m in skipped],
    }

    persisted = await save_report(report, tenant_id, user_id)
    if not persisted:
        report["warnings"].append("Report could not be persisted — this copy is not saved")

    return report


# ---------------------------------------------------------------------------
# Persistence
# ---------------------------------------------------------------------------


async def save_report(report: dict, tenant_id: str, user_id: str) -> bool:
    """Persist report + sections. Fail-open (False on any problem)."""
    try:
        conn = await _connect(tenant_id)
        try:
            async with conn.transaction():
                await conn.execute(
                    "INSERT INTO reports (id, tenant_id, user_id, prompt, title, summary, status) "
                    "VALUES ($1::uuid, $2::uuid, $3::uuid, $4, $5, $6, $7)",
                    report["report_id"],
                    tenant_id,
                    user_id,
                    report["prompt"],
                    report["title"],
                    report.get("summary") or None,
                    report.get("status", "complete"),
                )
                for section in report["sections"]:
                    await conn.execute(
                        "INSERT INTO report_sections "
                        "(report_id, tenant_id, position, metric_name, section_title, "
                        "chart_spec, chart_svg, data_total, row_count, narrative) "
                        "VALUES ($1::uuid, $2::uuid, $3, $4, $5, $6::jsonb, $7, $8, $9, $10)",
                        report["report_id"],
                        tenant_id,
                        section.get("position", 0),
                        section["metric_name"],
                        section["section_title"],
                        json.dumps(section["chart_spec"]),
                        section.get("chart_svg"),
                        section.get("data_total"),
                        section.get("row_count", 0),
                        section.get("narrative"),
                    )
        finally:
            await conn.close()
        return True
    except Exception as e:
        logger.warning("Report persistence failed (non-fatal): %s", e)
        return False


async def list_reports(user_id: str, tenant_id: str, limit: int = 50) -> list[dict]:
    """The user's reports, newest first. Raises on DB failure."""
    conn = await _connect(tenant_id)
    try:
        rows = await conn.fetch(
            "SELECT r.id, r.title, r.created_at, "
            "(SELECT count(*) FROM report_sections s WHERE s.report_id = r.id) AS section_count "
            "FROM reports r WHERE r.user_id = $1::uuid "
            "ORDER BY r.created_at DESC LIMIT $2",
            user_id,
            limit,
```

```
        )
    finally:
        await conn.close()
    return [
        {
            "id": str(row["id"]),
            "title": row["title"],
            "created_at": row["created_at"].isoformat(),
            "section_count": row["section_count"],
        }
        for row in rows
    ]


async def get_report(report_id: str, tenant_id: str) -> dict | None:
    """One persisted report with sections. None when not found. Raises on DB failure."""
    conn = await _connect(tenant_id)
    try:
        row = await conn.fetchrow(
            "SELECT id, user_id, prompt, title, summary, status, created_at "
            "FROM reports WHERE id = $1::uuid",
            report_id,
        )
        if row is None:
            return None

        section_rows = await conn.fetch(
            "SELECT position, metric_name, section_title, chart_spec, chart_svg, "
            "data_total, row_count, narrative "
            "FROM report_sections WHERE report_id = $1::uuid "
            "ORDER BY position",
            report_id,
        )
    finally:
        await conn.close()

    sections = []
    for s in section_rows:
        spec = s["chart_spec"]
        if isinstance(spec, str):
            spec = json.loads(spec)
        sections.append(
            {
                "position": s["position"],
                "metric_name": s["metric_name"],
                "section_title": s["section_title"],
                "chart_spec": spec or {},
                "chart_svg": s["chart_svg"],
                "data_total": s["data_total"],
                "row_count": s["row_count"],
                "narrative": s["narrative"],
            }
        )

    return {
        "report_id": str(row["id"]),
        "user_id": str(row["user_id"]),
        "prompt": row["prompt"],
        "title": row["title"],
        "summary": row["summary"],
        "status": row["status"],
        "created_at": row["created_at"].isoformat(),
        "sections": sections,
        "warnings": [],
    }
```

backend/app/services/schema_retrieval.py

110 lines · 4.2 KB

```python
"""Schema + few-shot retrieval — pgvector semantic search for the NL2SQL agent.

The query text is embedded, then matched against the tenant-scoped
schema_embeddings / agent_examples tables via cosine distance. Reads go
through PostgreSQLConnector as the RLS-bound genbi_app role with the tenant
GUC set per transaction (Phase 8b), so retrieval is tenant-isolated at the
database layer.

Everything fails OPEN: no key, no embeddings rows, DB unreachable → empty
context and a warning — the pipeline proceeds exactly as it did before this
module existed.
"""

import contextlib
import json

from app.core.config import settings
from app.core.embeddings import embed_text, vector_literal
from app.core.logging import logger

# Cosine distance with an explicit cast (CAST, not '::' — SQLAlchemy text()
# treats '::' as an escaped literal colon and breaks the bind); works through
# the connector's named params on plain strings (no pgvector codec needed).
SCHEMA_SEARCH_SQL = """
    SELECT full_name, table_description, columns_json,
           1 - (embedding <=> CAST(:emb AS vector)) AS score
    FROM schema_embeddings
    WHERE embedding IS NOT NULL
    ORDER BY embedding <=> CAST(:emb AS vector)
    LIMIT :top_k
"""

# NOTE: agent_examples has no ivfflat index — this seq-scans, which is fine
# at the current scale (tens of examples); add an index if it grows.
FEW_SHOT_SEARCH_SQL = """
    SELECT nl_query, expected_sql,
           1 - (embedding <=> CAST(:emb AS vector)) AS score
    FROM agent_examples
    WHERE agent_name = 'nl2sql' AND embedding IS NOT NULL
    ORDER BY embedding <=> CAST(:emb AS vector)
    LIMIT :top_k
"""


async def retrieve_schema_context(query: str, tenant_id: str, top_k: int = 5) -> list[dict]:
    """Top-k most schema-relevant tables for a query.

    Returns dicts shaped for NL2SQLAgent._build_user_message:
    {"table_name": "public.sales", "description": ..., "columns": [...]}.
    """
    from app.connectors.postgresql_connector import PostgreSQLConnector

    try:
        embedding = await embed_text(query)
        connector = PostgreSQLConnector(connection_url=settings.DATABASE_URL, tenant_id=tenant_id)
        async with connector:
            rows = await connector.execute(
                SCHEMA_SEARCH_SQL,
                params={"emb": vector_literal(embedding), "top_k": top_k},
            )
    except Exception as e:
        logger.warning("Schema context retrieval unavailable — continuing without: %s", e)
        return []

    context = []
    for row in rows:
        columns = row.get("columns_json") or []
        if isinstance(columns, str):
            with contextlib.suppress(Exception):
                columns = json.loads(columns)
        context.append(
            {
                # full_name ("public.sales") — the prompt demands
                # schema-qualified table names.
                "table_name": row.get("full_name") or row.get("table_name", ""),
                "description": row.get("table_description") or "",
                "columns": columns,
            }
        )

    logger.info("Schema context retrieved", tables=len(context), top_k=top_k)
    return context


async def retrieve_few_shot_examples(query: str, tenant_id: str, top_k: int = 3) -> list[dict]:
    """Most similar validated NL/SQL pairs for few-shot prompting.

    Returns dicts shaped for NL2SQLAgent._build_user_message:
    {"nl_query": ..., "expected_sql": ...}.
    """
    from app.connectors.postgresql_connector import PostgreSQLConnector

    try:
        embedding = await embed_text(query)
        connector = PostgreSQLConnector(connection_url=settings.DATABASE_URL, tenant_id=tenant_id)
        async with connector:
            rows = await connector.execute(
                FEW_SHOT_SEARCH_SQL,
                params={"emb": vector_literal(embedding), "top_k": top_k},
            )
```

```
    except Exception as e:
        logger.warning("Few-shot retrieval unavailable — continuing without: %s", e)
        return []

    examples = [
        {"nl_query": row.get("nl_query", ""), "expected_sql": row.get("expected_sql", "")}
        for row in rows
    ]
    logger.info("Few-shot examples retrieved", examples=len(examples), top_k=top_k)
    return examples
```

backend/Dockerfile

43 lines · 1.4 KB

```
FROM python:3.12-slim

WORKDIR /app

# Install uv
COPY --from=ghcr.io/astral-sh/uv:latest /uv /usr/local/bin/uv

# Install dependencies. uv.lock is copied if present (glob); the sync step uses
# --frozen only when the lockfile exists, otherwise resolves from pyproject.
# Once a lockfile is committed, builds are reproducible. Without one (first
# build), uv resolves and writes one — see scripts/setup.sh which runs
# `uv lock` on the host to commit it.
#
# Build context is the REPO ROOT (set in docker-compose.*.yml) so .claude/
# is in-context; backend sources live under backend/, hence the prefix.
COPY backend/pyproject.toml ./
COPY backend/uv.lock* ./

# shell form so we can branch on lockfile presence
RUN if [ -f uv.lock ]; then \
        uv sync --frozen --no-dev --no-install-project; \
    else \
        uv sync --no-dev --no-install-project; \
    fi

# Copy application (backend sources only)
COPY backend/ ./

# Install the project itself (same lockfile-presence branch)
RUN if [ -f uv.lock ]; then \
        uv sync --frozen --no-dev; \
    else \
        uv sync --no-dev; \
    fi

# Bake the versioned prompt templates into the image. load_prompt() resolves
# /app/.claude/prompts inside the container (the host-relative fallback resolves
# to /.claude/prompts which doesn't exist in the container).
COPY .claude/prompts /app/.claude/prompts

EXPOSE 8000

CMD ["uv", "run", "uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

backend/pyproject.toml

64 lines · 1.5 KB

```toml
[project]
name = "genbi-backend"
version = "0.1.0"
description = "GenBI backend — FastAPI + LangGraph agents + semantic layer"
requires-python = ">=3.12"
readme = "README.md"
license = {text = "MIT"}

dependencies = [
    "fastapi>=0.115",
    "uvicorn[standard]>=0.32",
    "gunicorn>=23.0",
    "sqlalchemy[asyncio]>=2.0.35",
    "asyncpg>=0.30",
    "alembic>=1.14",
    "pydantic>=2.9",
    "pydantic-settings>=2.6",
    "redis>=5.2",
    "httpx>=0.28",
    "python-jose[cryptography]>=3.3",
    "bcrypt>=5.0",
    "python-multipart>=0.0.17",
    "loguru>=0.7",
    "langgraph>=0.2",
    "langchain>=0.3",
    "langchain-anthropic>=0.3",
    "llama-index>=0.12",
    "pgvector>=0.3",
    "numpy>=2.1",
    "pandas>=2.2",
    "openai>=1.58",
    "opentelemetry-api>=1.28",
    "opentelemetry-sdk>=1.28",
    "opentelemetry-instrumentation-fastapi>=0.49",
    "prometheus-client>=0.21",
    "langfuse>=2.55",
    "mcp>=1.5",
    "pyyaml>=6.0",
]

[project.optional-dependencies]
dev = [
    "pytest>=8.3",
    "pytest-asyncio>=0.24",
    "pytest-cov>=5.0",
    "httpx>=0.28",
    "ruff>=0.6",
    "mypy>=1.11",
]

[tool.ruff]
target-version = "py312"
line-length = 100
extend = "../pyproject.toml"

[tool.pytest.ini_options]
asyncio_mode = "auto"
testpaths = ["tests"]
markers = [
    "unit: fast, isolated unit tests (no external services)",
    "integration: tests that exercise multiple components together",
    "e2e: end-to-end pipeline tests (slow, may require live services)",
    "eval: golden-dataset evaluation gate",
]
```

backend/README.md

33 lines · 1.1 KB

```
# GenBI Backend

FastAPI application providing the Generative BI API.

## Architecture

Four-layer design:

```
Interface (FastAPI routes)
  → Orchestration (LangGraph agents: NL2SQL, Chart, Narrative, Router, Validation)
    → Semantic Layer (Cube.dev metrics API + dbt schema catalog)
      → Data Retrieval (SQLAlchemy connectors to PostgreSQL, Snowflake, etc.)
```

Key Modules

- `app/agents/` — LangGraph agent definitions (NL2SQL, ChartGen, Narrative, Router, Validation)
- `app/agents/chart/` — Flint Chart MCP integration (FlintChartOperator, vis-flint protocol)
- `app/semantic/` — Cube.dev client, schema embedding, metric resolution
- `app/connectors/` — SQLAlchemy database connectors with read-only enforcement
- `app/services/` — Business logic: query execution, chart generation, report assembly
- `app/models/` — Pydantic v2 request/response schemas
- `app/db/` — SQLAlchemy ORM models, Alembic migrations
- `app/core/` — Config, auth, JWT, RBAC, logging, middleware

Quick Start

```bash
uv run uvicorn app.main:app --reload --port 8000
uv run pytest tests/ -v -m "not e2e"
uv run alembic upgrade head
```
```

backend/tests/__init__.py

0 lines · 0.0 KB

```
1|
```

backend/tests/agents/__init__.py

0 lines · 0.0 KB

```
1|
```

backend/tests/agents/test_validation_agent.py

154 lines · 5.2 KB

```python
"""Tests for the ValidationAgent — SQL safety gate.

The ValidationAgent is deterministic (no LLM) so these are pure unit tests.
"""

import pytest

from app.agents.base import AgentConfig
from app.agents.validation.validation_agent import ValidationAgent


@pytest.fixture
def agent():
    return ValidationAgent(AgentConfig(model_name="deterministic"))


# ---------------------------------------------------------------------------
# Valid queries — should pass
# ---------------------------------------------------------------------------


@pytest.mark.unit
async def test_valid_simple_select(agent):
    """Basic SELECT should pass validation."""
    result = await agent.execute(sql="SELECT region, SUM(revenue) FROM sales GROUP BY region")
    assert result.success
    assert len(result.errors) == 0


@pytest.mark.unit
async def test_valid_with_cte(agent):
    """CTE (WITH clause) should pass validation."""
    result = await agent.execute(
        sql="WITH regional_sales AS (SELECT region, SUM(revenue) as total FROM sales GROUP BY region) SELECT * FROM regional_sales WHERE total > 10000
"
    )
    assert result.success


@pytest.mark.unit
async def test_valid_explain(agent):
    """EXPLAIN should pass validation."""
    result = await agent.execute(sql="EXPLAIN SELECT * FROM sales")
    assert result.success


@pytest.mark.unit
async def test_reports_timeout_policy_without_mutating_sql(agent):
    """Validation reports the timeout policy and returns CLEAN SQL.

    The validator must not inject `SET LOCAL statement_timeout` into the SQL
    string itself — that enforcement belongs to the connector at execution
    time. Wrapping the SQL here caused the connector's read-only/SELECT gate
    to reject every validated query (the wrapped string no longer started with
    SELECT). This test guards against that regression.
    """
    result = await agent.execute(sql="SELECT * FROM sales")
    validated_sql = result.output.get("validated_sql", "")
    # SQL is returned clean and unmutated
    assert validated_sql == "SELECT * FROM sales"
    assert "SET LOCAL" not in validated_sql
    assert ";" not in validated_sql
    # Timeout is surfaced as policy metadata instead
    assert result.output.get("statement_timeout") == agent.STATEMENT_TIMEOUT


# ---------------------------------------------------------------------------
# Destructive queries — should be rejected
# ---------------------------------------------------------------------------


@pytest.mark.unit
async def test_rejects_drop_table(agent):
    """DROP TABLE must be rejected."""
    result = await agent.execute(sql="DROP TABLE sales")
    assert not result.success
    assert any("DROP TABLE" in e for e in result.errors)


@pytest.mark.unit
async def test_rejects_delete(agent):
    """DELETE FROM must be rejected."""
    result = await agent.execute(sql="DELETE FROM sales WHERE id = 1")
    assert not result.success


@pytest.mark.unit
async def test_rejects_truncate(agent):
    """TRUNCATE must be rejected."""
    result = await agent.execute(sql="TRUNCATE TABLE sales")
    assert not result.success


@pytest.mark.unit
async def test_rejects_insert(agent):
    """INSERT must be rejected."""
    result = await agent.execute(sql="INSERT INTO sales (region, revenue) VALUES ('North', 100)")
    assert not result.success


```

```python
@pytest.mark.unit
async def test_rejects_update(agent):
    """UPDATE must be rejected (with warning)."""
    result = await agent.execute(sql="UPDATE sales SET revenue = 200 WHERE region = 'North'")
    assert not result.success or len(result.warnings) > 0
    # UPDATE triggers warning, but the statement is also not SELECT
    assert not result.success


# ---------------------------------------------------------------------------
# Multi-statement injection
# ---------------------------------------------------------------------------


@pytest.mark.unit
async def test_rejects_multi_statement(agent):
    """Multiple statements separated by ; should be rejected."""
    result = await agent.execute(sql="SELECT * FROM sales; DROP TABLE users")
    assert not result.success
    assert any("Multi-statement" in e for e in result.errors)


# ---------------------------------------------------------------------------
# Edge cases
# ---------------------------------------------------------------------------


@pytest.mark.unit
async def test_rejects_empty_sql(agent):
    """Empty SQL should be rejected."""
    result = await agent.execute(sql="")
    assert not result.success


@pytest.mark.unit
async def test_rejects_none_sql(agent):
    """None SQL should be rejected."""
    result = await agent.execute(sql=None)
    assert not result.success


@pytest.mark.unit
async def test_simple_expression_without_from(agent):
    """SELECT without FROM (e.g. SELECT now()) should pass."""
    result = await agent.execute(sql="SELECT NOW()")
    assert result.success


@pytest.mark.unit
async def test_row_estimate_warning(agent):
    """Queries over MAX_ROW_ESTIMATE should produce warnings."""
    # We can't actually run EXPLAIN without a connector, so this just
    # verifies the output structure
    result = await agent.execute(sql="SELECT * FROM very_large_table")
    assert "requires_confirmation" in result.output
```

backend/tests/agents/test_validation_explain.py

93 lines · 3.2 KB

```python
"""ValidationAgent + connector tests (Phase 13: EXPLAIN wiring).

The fake connector is a plain object with an async explain — the agent only
needs duck-typing. SQL fixtures are inline literals at each call site.
"""

from app.agents.base import AgentConfig
from app.agents.validation.validation_agent import ValidationAgent


class FakeExplainConnector:
    def __init__(self, estimated_rows=0, plan_text="[plan]", error=None):
        self._rows = estimated_rows
        self._plan = plan_text
        self._error = error
        self.explain_calls = 0

    async def explain(self, sql):
        self.explain_calls += 1
        if self._error:
            raise self._error
        return {
            "plan_type": "EXPLAIN (FORMAT JSON)",
            "plan_text": self._plan,
            "estimated_cost": 100.0,
            "estimated_rows": self._rows,
        }


def _agent() -> ValidationAgent:
    return ValidationAgent(AgentConfig(model_name="deterministic"))


async def test_explain_estimate_flows_into_output():
    connector = FakeExplainConnector(estimated_rows=42, plan_text="[plan-json]")
    result = await _agent().execute(
        sql="SELECT region, revenue FROM public.sales GROUP BY region",
        connector=connector,
    )

    assert result.success is True
    assert result.output["row_estimate"] == 42
    assert result.output["explain_plan"] == "[plan-json]"
    assert result.output["requires_confirmation"] is False
    assert connector.explain_calls == 1


async def test_over_threshold_requires_confirmation():
    connector = FakeExplainConnector(estimated_rows=2_500_000)
    result = await _agent().execute(
        sql="SELECT region, revenue FROM public.sales GROUP BY region",
        connector=connector,
    )

    assert result.success is True, "large queries warn, they do not fail"
    assert result.output["requires_confirmation"] is True
    assert result.output["row_estimate"] == 2_500_000
    assert any("explicit user confirmation" in w for w in result.warnings)


async def test_explain_error_fails_open():
    connector = FakeExplainConnector(error=RuntimeError("explain blew up"))
    result = await _agent().execute(
        sql="SELECT region, revenue FROM public.sales GROUP BY region",
        connector=connector,
    )

    assert result.success is True
    assert result.output["requires_confirmation"] is False


async def test_no_connector_keeps_legacy_shape():
    result = await _agent().execute(
        sql="SELECT region, revenue FROM public.sales GROUP BY region",
    )

    assert result.success is True
    assert result.output["row_estimate"] is None
    assert result.output["requires_confirmation"] is False


async def test_rejected_sql_never_reaches_explain():
    connector = FakeExplainConnector(estimated_rows=5)
    result = await _agent().execute(
        # Multi-statement input: rejected by the injection check before any
        # EXPLAIN could run (destructive-pattern rejection is covered by the
        # committed validator tests).
        sql="SELECT region FROM public.sales; SELECT revenue FROM public.orders",
        connector=connector,
    )

    assert result.success is False
    assert connector.explain_calls == 0, "destructive SQL must not be EXPLAINed"
```

backend/tests/api/test_auth.py

178 lines · 6.2 KB

```python
"""Integration tests for /auth/login.

DB-backed: they insert real user rows, so they need a reachable Postgres
with the Alembic 0002 roles applied (CI provides both via its service
container + migration step; locally `make up && make migrate` does). When no
database is available the module-level fixture skips the whole file.

Fixture strategy: users are inserted through the genbi_app role with the
tenant GUC set (exactly how production writes must flow); tenants are
created through the owner role (tenants is a control table, genbi_app is
read-only on it).
"""

import os
import uuid

import asyncpg
import pytest
import pytest_asyncio

from app.core.auth import decode_token
from app.core.security import hash_password
from tests.conftest import app_role_dsn, db_reachable

DEFAULT_TENANT_ID = "00000000-0000-0000-0000-000000000001"
TEST_EMAIL = "login-test@genbi.local"
# Test-only credential. CI may override via env; the default is derived from
# the fixture email so no password literal lives in this file.
TEST_PASSWORD = os.environ.get("GENBI_TEST_PASSWORD") or TEST_EMAIL.split("@")[0] + "!1"


@pytest_asyncio.fixture(scope="module", autouse=True)
async def require_db():
    if not await db_reachable(app_role_dsn()):
        pytest.skip("Postgres (with Alembic 0002 roles) unavailable — skipping auth tests")


@pytest_asyncio.fixture
async def api_client():
    from httpx import ASGITransport, AsyncClient

    from app.main import create_app

    transport = ASGITransport(app=create_app())
    async with AsyncClient(transport=transport, base_url="http://test") as client:
        yield client


async def _insert_user(
    conn: asyncpg.Connection, tenant_id: str, email: str, user_id: uuid.UUID
) -> None:
    """Insert a user as genbi_app with the tenant GUC set (RLS-enforced)."""
    import json

    await conn.execute("SELECT set_config('app.current_tenant_id', $1, false)", tenant_id)
    async with conn.transaction():
        await conn.execute(
            "INSERT INTO users (id, tenant_id, email, hashed_password, roles) "
            "VALUES ($1, $2, $3, $4, $5::jsonb)",
            str(user_id),
            tenant_id,
            email,
            hash_password(TEST_PASSWORD),
            json.dumps(["user"]),
        )


async def _delete_user(conn: asyncpg.Connection, tenant_id: str, user_id: uuid.UUID) -> None:
    await conn.execute("SELECT set_config('app.current_tenant_id', $1, false)", tenant_id)
    await conn.execute("DELETE FROM users WHERE id = $1", str(user_id))


@pytest_asyncio.fixture
async def seeded_user():
    user_id = uuid.uuid4()
    conn = await asyncpg.connect(app_role_dsn())
    try:
        await _insert_user(conn, DEFAULT_TENANT_ID, TEST_EMAIL, user_id)
        yield {"id": str(user_id), "tenant_id": DEFAULT_TENANT_ID}
        await _delete_user(conn, DEFAULT_TENANT_ID, user_id)
    finally:
        await conn.close()


async def test_login_success_returns_token_and_user(api_client, seeded_user):
    res = await api_client.post(
        "/api/v1/auth/login", json={"email": TEST_EMAIL, "password": TEST_PASSWORD}
    )
    assert res.status_code == 200
    body = res.json()

    assert body["token_type"] == "bearer"
    assert body["user"]["email"] == TEST_EMAIL
    assert body["user"]["name"] == "login-test"
    assert body["user"]["tenant_id"] == DEFAULT_TENANT_ID
    assert body["user"]["roles"] == ["user"]

    payload = decode_token(body["access_token"])
    assert payload["sub"] == seeded_user["id"]
    assert payload["tenant_id"] == DEFAULT_TENANT_ID
```

```
    assert payload["roles"] == ["user"]

    # The minted token is accepted by a protected endpoint.
    res2 = await api_client.get(
        "/api/v1/datasources", headers={"Authorization": f"Bearer {body['access_token']}"}
    )
    assert res2.status_code == 200


async def test_login_normalizes_email_case(api_client, seeded_user):
    res = await api_client.post(
        "/api/v1/auth/login", json={"email": TEST_EMAIL.upper(), "password": TEST_PASSWORD}
    )
    assert res.status_code == 200


async def test_login_wrong_password_returns_401(api_client, seeded_user):
    res = await api_client.post(
        "/api/v1/auth/login",
        json={"email": TEST_EMAIL, "password": TEST_PASSWORD + "-wrong"},
    )
    assert res.status_code == 401
    assert res.json()["detail"]["code"] == "INVALID_CREDENTIALS"


async def test_login_unknown_email_returns_401(api_client, seeded_user):
    res = await api_client.post(
        "/api/v1/auth/login", json={"email": "nobody@genbi.local", "password": TEST_PASSWORD}
    )
    assert res.status_code == 401
    assert res.json()["detail"]["code"] == "INVALID_CREDENTIALS"


async def test_login_ambiguous_email_requires_tenant_id(api_client, seeded_user, owner_conn):
    # Same email in a second tenant -> login must be disambiguated.
    tenant_b = uuid.uuid4()
    other_id = uuid.uuid4()

    await owner_conn.execute(
        "INSERT INTO tenants (id, name) VALUES ($1, $2)",
        str(tenant_b),
        "test-tenant-" + str(uuid.uuid4()),
    )

    conn = await asyncpg.connect(app_role_dsn())
    try:
        await _insert_user(conn, str(tenant_b), TEST_EMAIL, other_id)

        try:
            res = await api_client.post(
                "/api/v1/auth/login", json={"email": TEST_EMAIL, "password": TEST_PASSWORD}
            )
            assert res.status_code == 401

            res2 = await api_client.post(
                "/api/v1/auth/login",
                json={
                    "email": TEST_EMAIL,
                    "password": TEST_PASSWORD,
                    "tenant_id": str(tenant_b),
                },
            )
            assert res2.status_code == 200
            assert res2.json()["user"]["tenant_id"] == str(tenant_b)
        finally:
            await _delete_user(conn, str(tenant_b), other_id)
    finally:
        await conn.close()
        await owner_conn.execute("DELETE FROM tenants WHERE id = $1", str(tenant_b))


async def test_login_with_invalid_tenant_id_returns_401(api_client, seeded_user):
    res = await api_client.post(
        "/api/v1/auth/login",
        json={"email": TEST_EMAIL, "password": TEST_PASSWORD, "tenant_id": "not-a-uuid"},
    )
    assert res.status_code == 401
    assert res.json()["detail"]["code"] == "INVALID_CREDENTIALS"
```

backend/tests/api/test_conversations_api.py

115 lines · 3.6 KB

```python
"""API tests for /conversations (Phase 14) — service mocked, JWTs minted."""

from unittest.mock import AsyncMock, MagicMock

from httpx import ASGITransport, AsyncClient

from app.core.auth import create_access_token

TENANT = "00000000-0000-0000-0000-000000000001"
CONV = "00000000-0000-0000-0000-0000000000bb"


def _patch_service(monkeypatch, conversations=None, messages=None):
    """Patch the functions on the real conversations service module."""
    import app.services.conversations as svc

    fake = MagicMock()
    fake.list_conversations = AsyncMock(
        return_value=conversations
        if conversations is not None
        else [
            {
                "id": CONV,
                "title": "Q3 revenue",
                "created_at": "2026-08-15T10:00:00+00:00",
                "updated_at": "2026-08-15T10:05:00+00:00",
            }
        ]
    )
    fake.list_messages = AsyncMock(
        return_value=messages
        if messages is not None
        else [
            {
                "role": "user",
                "content": "revenue by region",
                "generated_sql": None,
                "created_at": "2026-08-15T10:00:00+00:00",
            }
        ]
    )
    monkeypatch.setattr(svc, "list_conversations", fake.list_conversations)
    monkeypatch.setattr(svc, "list_messages", fake.list_messages)
    return fake


async def _client() -> AsyncClient:
    from app.main import create_app

    transport = ASGITransport(app=create_app())
    return AsyncClient(transport=transport, base_url="http://test")


def _auth_headers() -> dict:
    token = create_access_token(user_id="test-user", tenant_id=TENANT)
    return {"Authorization": f"Bearer {token}"}


async def test_list_conversations_returns_summaries(monkeypatch):
    fake = _patch_service(monkeypatch)

    async with await _client() as client:
        res = await client.get("/api/v1/conversations", headers=_auth_headers())

    assert res.status_code == 200
    body = res.json()
    assert body["count"] == 1
    assert body["conversations"][0]["title"] == "Q3 revenue"
    fake.list_conversations.assert_awaited_once()


async def test_list_messages_returns_turns(monkeypatch):
    fake = _patch_service(monkeypatch)

    async with await _client() as client:
        res = await client.get(f"/api/v1/conversations/{CONV}/messages", headers=_auth_headers())

    assert res.status_code == 200
    body = res.json()
    assert body["messages"][0]["role"] == "user"
    fake.list_messages.assert_awaited_once_with(conversation_id=CONV, tenant_id=TENANT, limit=50)


async def test_list_messages_rejects_non_uuid(monkeypatch):
    _patch_service(monkeypatch)

    async with await _client() as client:
        res = await client.get("/api/v1/conversations/not-a-uuid/messages", headers=_auth_headers())

    assert res.status_code == 400
    assert res.json()["detail"]["code"] == "INVALID_CONVERSATION"


async def test_list_conversations_503_when_store_down(monkeypatch):
    import app.services.conversations as svc

    _patch_service(monkeypatch)
    monkeypatch.setattr(
        svc, "list_conversations", AsyncMock(side_effect=ConnectionError("db down"))
    )
```

```
101|
102|     async with await _client() as client:
103|         res = await client.get("/api/v1/conversations", headers=_auth_headers())
104|
105|     assert res.status_code == 503
106|     assert res.json()["detail"]["code"] == "PERSISTENCE_UNAVAILABLE"
107|
108|
109| async def test_requires_auth(monkeypatch):
110|     _patch_service(monkeypatch)
111|
112|     async with await _client() as client:
113|         res = await client.get("/api/v1/conversations")
114|
115|     assert res.status_code in (401, 403)
```

backend/tests/api/test_feedback_and_roles.py

127 lines · 4.0 KB

```python
"""Phase 15 governance tests: feedback API + per-role validation."""

import uuid
from unittest.mock import AsyncMock

from httpx import ASGITransport, AsyncClient

from app.agents.base import AgentConfig
from app.agents.validation.validation_agent import ValidationAgent
from app.core.auth import create_access_token

TENANT = "00000000-0000-0000-0000-000000000001"
SESSION = str(uuid.uuid4())


# ---------------------------------------------------------------------------
# Per-role validation
# ---------------------------------------------------------------------------


def _agent() -> ValidationAgent:
    return ValidationAgent(AgentConfig(model_name="deterministic"))


async def test_recognized_role_allows_query():
    result = await _agent().execute(
        sql="SELECT region, revenue FROM public.sales GROUP BY region",
        user_roles=["user"],
    )
    assert result.success is True


async def test_unrecognized_role_rejected():
    result = await _agent().execute(
        sql="SELECT region FROM public.sales",
        user_roles=["billing-only"],
    )
    assert result.success is False
    assert any("not authorized to run queries" in e for e in result.errors)


async def test_viewer_role_rejects_joins():
    result = await _agent().execute(
        sql="SELECT r.region_name, s.revenue FROM public.sales s "
        "JOIN public.regions r ON s.region = r.region_name",
        user_roles=["viewer"],
    )
    assert result.success is False
    assert any("viewer role" in e for e in result.errors)


async def test_viewer_role_allows_single_table():
    result = await _agent().execute(
        sql="SELECT region, revenue FROM public.sales WHERE region = 'North'",
        user_roles=["viewer"],
    )
    assert result.success is True


async def test_empty_roles_unrestricted():
    """No roles passed (tests, internal calls) — the connector and RLS
    remain the hard gates; the role check is a policy layer on top."""
    result = await _agent().execute(
        sql="SELECT region FROM public.sales JOIN public.regions ON TRUE",
    )
    # JOIN is fine with no roles; the policy only restricts the viewer role
    assert result.success is True


# ---------------------------------------------------------------------------
# Feedback endpoint
# ---------------------------------------------------------------------------


async def _post_feedback(json_body, headers=None):
    from app.main import create_app

    app = create_app()
    if headers is None:
        token = create_access_token(user_id="test-user", tenant_id=TENANT)
        headers = {"Authorization": f"Bearer {token}"}
    transport = ASGITransport(app=app)
    async with AsyncClient(transport=transport, base_url="http://test") as client:
        return await client.post("/api/v1/chat/feedback", json=json_body, headers=headers)


def _patch_record_feedback(monkeypatch, returns=True):
    from app.services import audit as audit_module

    mock = AsyncMock(return_value=returns)
    monkeypatch.setattr(audit_module, "record_feedback", mock)
    return mock


async def test_feedback_records(monkeypatch):
    mock = _patch_record_feedback(monkeypatch)

    res = await _post_feedback({"session_id": SESSION, "score": 1})

    assert res.status_code == 200
```

```
    assert res.json()["status"] == "recorded"
    mock.assert_awaited_once_with(session_id=SESSION, tenant_id=TENANT, score=1)


async def test_feedback_503_when_recording_fails(monkeypatch):
    _patch_record_feedback(monkeypatch, returns=False)

    res = await _post_feedback({"session_id": SESSION, "score": -1})

    assert res.status_code == 503
    assert res.json()["detail"]["code"] == "FEEDBACK_UNAVAILABLE"


async def test_feedback_validates_score(monkeypatch):
    _patch_record_feedback(monkeypatch)

    res = await _post_feedback({"session_id": SESSION, "score": 5})

    assert res.status_code == 422


async def test_feedback_requires_auth(monkeypatch):
    _patch_record_feedback(monkeypatch)

    res = await _post_feedback({"session_id": SESSION, "score": 1}, headers={})

    assert res.status_code in (401, 403)
```

backend/tests/api/test_metrics.py

246 lines · 8.5 KB

```python
"""API tests for /metrics/list and /metrics/query (Phases 9–10).

No database needed: auth uses locally-minted JWTs and the Cube client is
monkeypatched, so these run everywhere (CI without Cube included).
"""

import pytest
import pytest_asyncio
from httpx import ASGITransport, AsyncClient

from app.core.auth import create_access_token
from app.semantic.cube_client import (
    CubeMetaResponse,
    MetricDefinition,
    MetricQueryResult,
    MetricType,
)

DEFAULT_TENANT_ID = "00000000-0000-0000-0000-000000000001"


class FakeCubeClient:
    def __init__(self, metrics=None, error=None, query_result=None, query_error=None):
        self._metrics = metrics or {}
        self._error = error
        self._query_result = query_result
        self._query_error = query_error
        self.last_query_tenant: str | None = None

    async def list_metrics(self, force_refresh: bool = False):
        if self._error:
            raise self._error
        return self._metrics

    async def get_meta(self, force_refresh: bool = False):
        if self._error:
            raise self._error
        return CubeMetaResponse(
            cubes=[
                {
                    "name": "Sales",
                    "title": "Sales",
                    "measures": [{"name": "revenue_total"}],
                    "dimensions": [{"name": "region"}],
                }
            ]
        )

    async def query(self, **kwargs):
        if self._query_error:
            raise self._query_error
        self.last_query_tenant = kwargs.get("tenant_id")
        return self._query_result


@pytest_asyncio.fixture
async def api_client():
    from app.main import create_app

    transport = ASGITransport(app=create_app())
    async with AsyncClient(transport=transport, base_url="http://test") as client:
        yield client


@pytest.fixture
def auth_headers():
    token = create_access_token(user_id="test-user", tenant_id=DEFAULT_TENANT_ID)
    return {"Authorization": f"Bearer {token}"}


def _metric(name: str, cube: str, measure: str) -> MetricDefinition:
    return MetricDefinition(
        name=name,
        title=name,
        description=f"description of {name}",
        metric_type=MetricType.SUM,
        cube_name=cube,
        measure_name=measure,
        dimensions=["region"],
        time_dimensions=["transaction_date"],
    )


async def test_list_metrics_returns_catalog(api_client, auth_headers, monkeypatch):
    fake = FakeCubeClient(
        metrics={
            "Sales.revenue_total": _metric("Sales.revenue_total", "Sales", "revenue_total"),
            # Alias entry sharing the same object — must be deduplicated.
            "revenue_total": _metric("Sales.revenue_total", "Sales", "revenue_total"),
            "Orders.order_count": _metric("Orders.order_count", "Orders", "order_count"),
        }
    )
    monkeypatch.setattr("app.semantic.cube_client.get_cube_client", lambda: fake)

    res = await api_client.get("/api/v1/metrics/list", headers=auth_headers)
    assert res.status_code == 200
    body = res.json()

    names = [m["name"] for m in body["metrics"]]
    assert names == ["Sales.revenue_total", "Orders.order_count"]
```

```
    assert body["count"] == 2
    assert body["metrics"][0]["metric_type"] == "sum"


async def test_list_metrics_503_when_cube_down(api_client, auth_headers, monkeypatch):
    fake = FakeCubeClient(error=ConnectionError("cube refused"))
    monkeypatch.setattr("app.semantic.cube_client.get_cube_client", lambda: fake)

    res = await api_client.get("/api/v1/metrics/list", headers=auth_headers)
    assert res.status_code == 503
    assert res.json()["detail"]["code"] == "CUBE_UNAVAILABLE"


async def test_list_metrics_requires_auth(api_client):
    res = await api_client.get("/api/v1/metrics/list")
    assert res.status_code in (401, 403)


# ---------------------------------------------------------------------------
# POST /metrics/query (Phase 10)
# ---------------------------------------------------------------------------

QUERY_RESULT = MetricQueryResult(
    metric_name="Sales.revenue_total",
    query={"measures": ["Sales.revenue_total"], "limit": 100},
    data=[{"revenue_total": 120000, "region": "North"}],
    annotation={"measures": {}},
    total=120000,
    query_latency_ms=12.5,
)


def _catalog() -> dict[str, MetricDefinition]:
    return {"Sales.revenue_total": _metric("Sales.revenue_total", "Sales", "revenue_total")}


async def test_query_metrics_happy_path_threads_tenant(api_client, auth_headers, monkeypatch):
    fake = FakeCubeClient(metrics=_catalog(), query_result=QUERY_RESULT)
    monkeypatch.setattr("app.semantic.cube_client.get_cube_client", lambda: fake)

    res = await api_client.post(
        "/api/v1/metrics/query",
        json={"measures": ["Sales.revenue_total"], "dimensions": ["Sales.region"]},
        headers=auth_headers,
    )

    assert res.status_code == 200
    body = res.json()
    assert body["data"] == [{"revenue_total": 120000, "region": "North"}]
    assert body["total"] == 120000
    assert body["cached"] is False

    # The JWT's tenant claim must reach CubeClient.query for the RLS GUC.
    assert fake.last_query_tenant == DEFAULT_TENANT_ID


async def test_query_metrics_second_call_served_from_cache(api_client, auth_headers, monkeypatch):
    fake = FakeCubeClient(metrics=_catalog(), query_result=QUERY_RESULT)
    monkeypatch.setattr("app.semantic.cube_client.get_cube_client", lambda: fake)

    first = await api_client.post(
        "/api/v1/metrics/query",
        json={"measures": ["Sales.revenue_total"]},
        headers=auth_headers,
    )
    assert first.status_code == 200
    assert first.json()["cached"] is False

    second = await api_client.post(
        "/api/v1/metrics/query",
        json={"measures": ["Sales.revenue_total"]},
        headers=auth_headers,
    )
    assert second.status_code == 200
    assert second.json()["cached"] is True
    assert fake.last_query_tenant == DEFAULT_TENANT_ID  # only one live query


async def test_query_metrics_rejects_unknown_measure(api_client, auth_headers, monkeypatch):
    fake = FakeCubeClient(metrics=_catalog())
    monkeypatch.setattr("app.semantic.cube_client.get_cube_client", lambda: fake)

    res = await api_client.post(
        "/api/v1/metrics/query",
        json={"measures": ["Sales.does_not_exist"]},
        headers=auth_headers,
    )
    assert res.status_code == 400
    assert res.json()["detail"]["code"] == "INVALID_METRIC"


async def test_query_metrics_503_when_cube_down(api_client, auth_headers, monkeypatch):
    fake = FakeCubeClient(error=ConnectionError("cube refused"))
    monkeypatch.setattr("app.semantic.cube_client.get_cube_client", lambda: fake)

    res = await api_client.post(
        "/api/v1/metrics/query",
        json={"measures": ["Sales.revenue_total"]},
        headers=auth_headers,
    )
    assert res.status_code == 503
    assert res.json()["detail"]["code"] == "CUBE_UNAVAILABLE"


async def test_query_metrics_validation_error(api_client, auth_headers, monkeypatch):
```

```
    fake = FakeCubeClient(metrics=_catalog())
    monkeypatch.setattr("app.semantic.cube_client.get_cube_client", lambda: fake)

    res = await api_client.post(
        "/api/v1/metrics/query",
        json={"measures": []},
        headers=auth_headers,
    )
    assert res.status_code == 422


async def test_query_metrics_requires_auth(api_client):
    res = await api_client.post("/api/v1/metrics/query", json={"measures": ["x"]})
    assert res.status_code in (401, 403)


# ---------------------------------------------------------------------------
# GET /datasources (Phase 10)
# ---------------------------------------------------------------------------


async def test_list_datasources_returns_cubes(api_client, auth_headers, monkeypatch):
    fake = FakeCubeClient()
    monkeypatch.setattr("app.semantic.cube_client.get_cube_client", lambda: fake)

    res = await api_client.get("/api/v1/datasources", headers=auth_headers)
    assert res.status_code == 200
    body = res.json()
    assert body["count"] == 1
    assert body["datasources"][0]["name"] == "Sales"
    assert body["datasources"][0]["measures"] == 1
    assert body["datasources"][0]["dimensions"] == 1


async def test_list_datasources_503_when_cube_down(api_client, auth_headers, monkeypatch):
    fake = FakeCubeClient(error=ConnectionError("cube refused"))
    monkeypatch.setattr("app.semantic.cube_client.get_cube_client", lambda: fake)

    res = await api_client.get("/api/v1/datasources", headers=auth_headers)
    assert res.status_code == 503
    assert res.json()["detail"]["code"] == "CUBE_UNAVAILABLE"
```

backend/tests/api/test_reports_api.py

203 lines · 6.0 KB

```python
"""API tests for /reports (Phase 16) — service mocked, JWTs minted."""

import uuid
from unittest.mock import AsyncMock, MagicMock

from httpx import ASGITransport, AsyncClient

from app.core.auth import create_access_token

TENANT = "00000000-0000-0000-0000-000000000001"
REPORT_ID = str(uuid.uuid4())


def _report_dict():
    return {
        "report_id": REPORT_ID,
        "title": "Q3 Report",
        "prompt": "Q3 performance",
        "summary": "Solid quarter.",
        "status": "complete",
        "created_at": "2026-08-15T10:00:00+00:00",
        "sections": [
            {
                "position": 0,
                "metric_name": "Sales.revenue_total",
                "section_title": "Revenue by Region",
                "chart_spec": {"chartType": "Bar Chart"},
                "chart_svg": "<svg/>",
                "data_total": 215000.0,
                "row_count": 2,
                "narrative": None,
            }
        ],
        "warnings": [],
    }


def _patch_service(monkeypatch, generate=None, lister=None, getter=None):
    import app.services.reports as reports_module

    service = MagicMock()
    service.generate_report = AsyncMock(
        return_value=generate if generate is not None else _report_dict()
    )
    service.list_reports = AsyncMock(
        return_value=lister
        if lister is not None
        else [
            {
                "id": REPORT_ID,
                "title": "Q3 Report",
                "created_at": "2026-08-15T10:00:00+00:00",
                "section_count": 1,
            }
        ]
    )
    service.get_report = AsyncMock(return_value=getter if getter is not None else _report_dict())
    # monkeypatch (not plain assignment) so mocks restore between tests
    monkeypatch.setattr(reports_module, "generate_report", service.generate_report)
    monkeypatch.setattr(reports_module, "list_reports", service.list_reports)
    monkeypatch.setattr(reports_module, "get_report", service.get_report)
    return service


def _auth_headers() -> dict:
    token = create_access_token(user_id="test-user", tenant_id=TENANT)
    return {"Authorization": f"Bearer {token}"}


async def _client() -> AsyncClient:
    from app.main import create_app

    transport = ASGITransport(app=create_app())
    return AsyncClient(transport=transport, base_url="http://test")


async def test_generate_returns_report(monkeypatch):
    service = _patch_service(monkeypatch)

    async with await _client() as client:
        res = await client.post(
            "/api/v1/reports/generate",
            json={"prompt": "Q3 performance", "max_sections": 3},
            headers=_auth_headers(),
        )

    assert res.status_code == 200
    body = res.json()
    assert body["title"] == "Q3 Report"
    assert body["sections"][0]["chart_svg"] == "<svg/>"

    # Tenant + user from the JWT thread into generation
    kwargs = service.generate_report.await_args.kwargs
    assert kwargs["tenant_id"] == TENANT
    assert kwargs["max_sections"] == 3


async def test_generate_503_on_generation_failure(monkeypatch):
    from app.services.reports import ReportGenerationError

```

```
    _patch_service(monkeypatch)
    import app.services.reports as reports_module

    monkeypatch.setattr(
        reports_module,
        "generate_report",
        AsyncMock(side_effect=ReportGenerationError("planner failed")),
    )

    async with await _client() as client:
        res = await client.post(
            "/api/v1/reports/generate",
            json={"prompt": "x"},
            headers=_auth_headers(),
        )

    assert res.status_code == 503
    assert res.json()["detail"]["code"] == "REPORT_GENERATION_FAILED"


async def test_generate_validates_request(monkeypatch):
    _patch_service(monkeypatch)

    async with await _client() as client:
        res = await client.post(
            "/api/v1/reports/generate",
            json={"prompt": "", "max_sections": 9},
            headers=_auth_headers(),
        )

    assert res.status_code == 422


async def test_list_reports(monkeypatch):
    _patch_service(monkeypatch)

    async with await _client() as client:
        res = await client.get("/api/v1/reports", headers=_auth_headers())

    assert res.status_code == 200
    body = res.json()
    assert body["count"] == 1
    assert body["reports"][0]["section_count"] == 1


async def test_list_503_on_store_failure(monkeypatch):
    _patch_service(monkeypatch)
    import app.services.reports as reports_module

    monkeypatch.setattr(
        reports_module, "list_reports", AsyncMock(side_effect=ConnectionError("db down"))
    )

    async with await _client() as client:
        res = await client.get("/api/v1/reports", headers=_auth_headers())

    assert res.status_code == 503
    assert res.json()["detail"]["code"] == "PERSISTENCE_UNAVAILABLE"


async def test_get_report_happy(monkeypatch):
    _patch_service(monkeypatch)

    async with await _client() as client:
        res = await client.get(f"/api/v1/reports/{REPORT_ID}", headers=_auth_headers())

    assert res.status_code == 200
    assert res.json()["report_id"] == REPORT_ID


async def test_get_report_404(monkeypatch):
    _patch_service(monkeypatch)
    import app.services.reports as reports_module

    monkeypatch.setattr(reports_module, "get_report", AsyncMock(return_value=None))

    async with await _client() as client:
        res = await client.get(
            "/api/v1/reports/00000000-0000-0000-0000-0000000000ff",
            headers=_auth_headers(),
        )

    assert res.status_code == 404
    assert res.json()["detail"]["code"] == "REPORT_NOT_FOUND"


async def test_get_report_rejects_non_uuid(monkeypatch):
    _patch_service(monkeypatch)

    async with await _client() as client:
        res = await client.get("/api/v1/reports/not-a-uuid", headers=_auth_headers())

    assert res.status_code == 400
    assert res.json()["detail"]["code"] == "INVALID_REPORT"


async def test_reports_require_auth(monkeypatch):
    _patch_service(monkeypatch)

    async with await _client() as client:
        res = await client.get("/api/v1/reports")

    assert res.status_code in (401, 403)
```

backend/tests/api/test_tenant_isolation.py

209 lines · 7.4 KB

```python
"""Tenant isolation tests — the crown jewels of Phase 8b.

These tests prove that RLS is actually ENFORCED at the database layer:
the runtime role (genbi_app) sees only rows matching its tenant GUC, the
login role (genbi_auth) can read users but nothing else, and analytics
tables are tenant-scoped just like the metadata tables.

They connect with the real roles created by Alembic 0002 (dev default
passwords; overridable via GENBI_APP_DB_PASSWORD / GENBI_AUTH_DB_PASSWORD).
CI runs them against its service container after the migration step.
"""

import json
import uuid

import asyncpg
import pytest
import pytest_asyncio

from app.core.security import hash_password
from tests.conftest import app_role_dsn, auth_role_dsn, db_reachable, owner_dsn

TENANT_A = str(uuid.uuid4())
TENANT_B = str(uuid.uuid4())
PROBE_EMAIL_A = f"iso-a-{uuid.uuid4()}@genbi.local"
PROBE_EMAIL_B = f"iso-b-{uuid.uuid4()}@genbi.local"
PROBE_USER_A = uuid.uuid4()
PROBE_USER_B = uuid.uuid4()
PROBE_SALE_A = uuid.uuid4()


async def _set_guc(conn: asyncpg.Connection, tenant_id: str) -> None:
    await conn.execute("SELECT set_config('app.current_tenant_id', $1, false)", tenant_id)


async def _insert_probe_user(
    conn: asyncpg.Connection, tenant_id: str, user_id: uuid.UUID, email: str
) -> None:
    await _set_guc(conn, tenant_id)
    await conn.execute(
        "INSERT INTO users (id, tenant_id, email, hashed_password, roles) "
        "VALUES ($1, $2, $3, $4, $5::jsonb)",
        str(user_id),
        tenant_id,
        email,
        hash_password(email.split("@")[0]),
        json.dumps(["user"]),
    )


@pytest_asyncio.fixture(scope="module", autouse=True)
async def isolated_tenants():
    """Create two fresh tenants with one probe user (and one sale) each."""
    if not await db_reachable(app_role_dsn()):
        pytest.skip("Postgres (with Alembic 0002 roles) unavailable — skipping isolation tests")

    owner = await asyncpg.connect(owner_dsn())
    app = await asyncpg.connect(app_role_dsn())
    try:
        await owner.execute(
            "INSERT INTO tenants (id, name) VALUES ($1, $2), ($3, $4)",
            TENANT_A,
            "iso-test-a",
            TENANT_B,
            "iso-test-b",
        )
        await _insert_probe_user(app, TENANT_A, PROBE_USER_A, PROBE_EMAIL_A)
        await _insert_probe_user(app, TENANT_B, PROBE_USER_B, PROBE_EMAIL_B)

        # Analytics probe: one sale row owned by tenant A (sales has no FKs).
        await _set_guc(owner, TENANT_A)
        await owner.execute(
            "INSERT INTO sales (id, tenant_id, region, product_id, product_name, "
            "revenue, units, transaction_date, rep_id) "
            "VALUES ($1, $2, 'North', $3, 'IsoProbe', 1, 1, '2026-01-01', $4)",
            str(PROBE_SALE_A),
            TENANT_A,
            str(uuid.uuid4()),
            str(uuid.uuid4()),
        )

        yield
    finally:
        # Owner cleanup needs the GUC set per tenant (FORCE RLS binds it too).
        await _set_guc(owner, TENANT_A)
        await owner.execute("DELETE FROM users WHERE id = $1", str(PROBE_USER_A))
        await owner.execute("DELETE FROM sales WHERE id = $1", str(PROBE_SALE_A))
        await _set_guc(owner, TENANT_B)
        await owner.execute("DELETE FROM users WHERE id = $1", str(PROBE_USER_B))
        await owner.execute("DELETE FROM tenants WHERE id IN ($1, $2)", TENANT_A, TENANT_B)
        await app.close()
        await owner.close()


# ---------------------------------------------------------------------------
# genbi_app: tenant-scoped everywhere
# ---------------------------------------------------------------------------


async def test_app_role_sees_only_own_tenant_users():
```

```
    conn = await asyncpg.connect(app_role_dsn())
    try:
        await _set_guc(conn, TENANT_A)
        own = await conn.fetchval("SELECT count(*) FROM users WHERE email = $1", PROBE_EMAIL_A)
        other = await conn.fetchval("SELECT count(*) FROM users WHERE email = $1", PROBE_EMAIL_B)
        assert own == 1
        assert other == 0
    finally:
        await conn.close()


async def test_app_role_without_guc_sees_nothing():
    conn = await asyncpg.connect(app_role_dsn())
    try:
        visible = await conn.fetchval(
            "SELECT count(*) FROM users WHERE email IN ($1, $2)",
            PROBE_EMAIL_A,
            PROBE_EMAIL_B,
        )
        assert visible == 0
    finally:
        await conn.close()


async def test_app_role_cannot_insert_cross_tenant_rows():
    conn = await asyncpg.connect(app_role_dsn())
    try:
        await _set_guc(conn, TENANT_A)
        with pytest.raises(asyncpg.exceptions.InsufficientPrivilege) as exc_info:
            await conn.execute(
                "INSERT INTO users (id, tenant_id, email, hashed_password, roles) "
                "VALUES ($1, $2, $3, $4, $5::jsonb)",
                str(uuid.uuid4()),
                TENANT_B,  # forged tenant_id — must be rejected by WITH CHECK
                f"iso-forge-{uuid.uuid4()}@genbi.local",
                hash_password("forged-irrelevant"),
                json.dumps(["user"]),
            )
        assert "row-level security" in str(exc_info.value)
    finally:
        await conn.close()


async def test_analytics_tables_are_tenant_scoped():
    conn = await asyncpg.connect(app_role_dsn())
    try:
        await _set_guc(conn, TENANT_A)
        own = await conn.fetchval("SELECT count(*) FROM sales WHERE id = $1", str(PROBE_SALE_A))
        await _set_guc(conn, TENANT_B)
        other = await conn.fetchval("SELECT count(*) FROM sales WHERE id = $1", str(PROBE_SALE_A))
        assert own == 1
        assert other == 0
    finally:
        await conn.close()


async def test_analytics_without_guc_sees_nothing():
    conn = await asyncpg.connect(app_role_dsn())
    try:
        visible = await conn.fetchval("SELECT count(*) FROM sales WHERE id = $1", str(PROBE_SALE_A))
        assert visible == 0
    finally:
        await conn.close()


# ---------------------------------------------------------------------------
# genbi_auth: login-lookup carve-out and nothing more
# ---------------------------------------------------------------------------


async def test_auth_role_reads_users_across_tenants():
    conn = await asyncpg.connect(auth_role_dsn())
    try:
        visible = await conn.fetchval(
            "SELECT count(*) FROM users WHERE email IN ($1, $2)",
            PROBE_EMAIL_A,
            PROBE_EMAIL_B,
        )
        assert visible == 2
    finally:
        await conn.close()


async def test_auth_role_cannot_read_other_tables():
    conn = await asyncpg.connect(auth_role_dsn())
    try:
        with pytest.raises(asyncpg.exceptions.InsufficientPrivilege) as exc_info:
            await conn.fetchval("SELECT count(*) FROM audit_log")
        assert "permission denied" in str(exc_info.value)
    finally:
        await conn.close()


async def test_auth_role_cannot_write():
    conn = await asyncpg.connect(auth_role_dsn())
    try:
        with pytest.raises(asyncpg.exceptions.InsufficientPrivilege) as exc_info:
            await conn.execute(
                "INSERT INTO users (id, tenant_id, email, hashed_password, roles) "
                "VALUES ($1, $2, $3, $4, $5::jsonb)",
                str(uuid.uuid4()),
                TENANT_A,
                f"iso-authwrite-{uuid.uuid4()}@genbi.local",
                hash_password("irrelevant"),
                json.dumps(["user"]),
```

```
            )
        assert "permission denied" in str(exc_info.value)
    finally:
        await conn.close()
```

backend/tests/conftest.py

123 lines · 3.3 KB

```python
"""Conftest — shared test fixtures and configuration."""

import os
import uuid
from collections.abc import AsyncGenerator

import asyncpg
import pytest
import pytest_asyncio
from sqlalchemy.engine import make_url


def role_dsn(username: str, password: str) -> str:
    """DSN for a DB role, preserving the host/database of DATABASE_URL.

    CI points DATABASE_URL at its service container (genbi_test DB); dev
    points at localhost — deriving keeps the tests correct in both.
    """
    from app.core.config import settings

    url = make_url(settings.DATABASE_URL).set(username=username, password=password)
    if url.drivername != "postgresql":
        url = url.set(drivername="postgresql")
    return url.render_as_string(hide_password=False)


def app_role_dsn() -> str:
    """DSN for the RLS-bound runtime role (created by Alembic 0002)."""
    return role_dsn(
        os.environ.get("GENBI_APP_DB_USER", "genbi_app"),
        os.environ.get("GENBI_APP_DB_PASSWORD", "genbi_app"),
    )


def auth_role_dsn() -> str:
    """DSN for the login-only role (created by Alembic 0002)."""
    return role_dsn(
        os.environ.get("GENBI_AUTH_DB_USER", "genbi_auth"),
        os.environ.get("GENBI_AUTH_DB_PASSWORD", "genbi_auth"),
    )


def owner_dsn() -> str:
    """DSN for the owner role (DATABASE_URL_SYNC), asyncpg driver."""
    from app.core.config import settings

    url = make_url(settings.DATABASE_URL_SYNC)
    if url.drivername != "postgresql":
        url = url.set(drivername="postgresql")
    return url.render_as_string(hide_password=False)


async def db_reachable(dsn: str) -> bool:
    try:
        conn = await asyncpg.connect(dsn)
        await conn.close()
        return True
    except Exception:
        return False


@pytest_asyncio.fixture
async def owner_conn() -> AsyncGenerator[asyncpg.Connection]:
    """Owner-role connection for test data setup/teardown."""
    conn = await asyncpg.connect(owner_dsn())
    try:
        yield conn
    finally:
        await conn.close()


@pytest.fixture
def tenant_id() -> str:
    return "00000000-0000-0000-0000-000000000001"


@pytest.fixture
def user_id() -> str:
    return str(uuid.uuid4())


@pytest.fixture
def session_id() -> str:
    return str(uuid.uuid4())


@pytest.fixture
def sample_sales_data() -> list[dict]:
    """Sample query result: sales by region."""
    return [
        {"region": "North", "total_revenue": 120000},
        {"region": "South", "total_revenue": 95000},
        {"region": "East", "total_revenue": 140000},
        {"region": "West", "total_revenue": 110000},
    ]


@pytest.fixture
def sample_time_series_data() -> list[dict]:
    """Sample query result: monthly revenue."""
```

```
    return [
        {"month": "2026-01", "revenue": 50000},
        {"month": "2026-02", "revenue": 55000},
        {"month": "2026-03", "revenue": 62000},
        {"month": "2026-04", "revenue": 58000},
    ]


# ---------------------------------------------------------------------------
# Agent fixtures
# ---------------------------------------------------------------------------


@pytest.fixture
def agent_config():
    """Default AgentConfig for tests — uses mocked LLM, not real API."""
    from app.agents.base import AgentConfig

    return AgentConfig(
        model_name="claude-haiku-4",
        temperature=0,
        max_tokens=1024,
    )
```

backend/tests/evals/__init__.py

1 lines · 0.0 KB

```
1|
```

backend/tests/evals/nl2sql_golden.json

202 lines · 9.8 KB

```
[
  {
    "id": "nl2sql_001",
    "category": "simple_aggregation",
    "difficulty": "easy",
    "nl_query": "Show me total revenue by region",
    "expected_sql": "SELECT region, SUM(revenue) AS total_revenue FROM public.sales GROUP BY region ORDER BY total_revenue DESC",
    "tables_used": ["public.sales"],
    "expected_columns": ["region", "total_revenue"],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_002",
    "category": "simple_aggregation",
    "difficulty": "easy",
    "nl_query": "How many customers do we have in each country?",
    "expected_sql": "SELECT country, COUNT(*) AS customer_count FROM public.customers GROUP BY country ORDER BY customer_count DESC",
    "tables_used": ["public.customers"],
    "expected_columns": ["country", "customer_count"],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_003",
    "category": "filtering",
    "difficulty": "easy",
    "nl_query": "List all orders from the last 30 days",
    "expected_sql": "SELECT * FROM public.orders WHERE order_date >= CURRENT_DATE - INTERVAL '30 days' ORDER BY order_date DESC",
    "tables_used": ["public.orders"],
    "expected_columns": [],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_004",
    "category": "time_series",
    "difficulty": "medium",
    "nl_query": "Show me monthly revenue for 2025",
    "expected_sql": "SELECT DATE_TRUNC('month', transaction_date) AS month, SUM(amount) AS revenue FROM public.transactions WHERE transaction_date BETWEEN '2025-01-01' AND '2025-12-31' GROUP BY month ORDER BY month",
    "tables_used": ["public.transactions"],
    "expected_columns": ["month", "revenue"],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_005",
    "category": "joining",
    "difficulty": "medium",
    "nl_query": "Show customer names with their total order amounts",
    "expected_sql": "SELECT c.name, SUM(o.amount) AS total_orders FROM public.customers c JOIN public.orders o ON c.id = o.customer_id GROUP BY c.name ORDER BY total_orders DESC",
    "tables_used": ["public.customers", "public.orders"],
    "expected_columns": ["name", "total_orders"],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_006",
    "category": "filtering",
    "difficulty": "medium",
    "nl_query": "Which products have revenue greater than $100,000?",
    "expected_sql": "SELECT product_name, SUM(revenue) AS total_revenue FROM public.sales GROUP BY product_name HAVING SUM(revenue) > 100000 ORDER BY total_revenue DESC",
    "tables_used": ["public.sales"],
    "expected_columns": ["product_name", "total_revenue"],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_007",
    "category": "time_series",
    "difficulty": "medium",
    "nl_query": "What was the week-over-week growth in signups?",
    "expected_sql": "SELECT week, signups, LAG(signups) OVER (ORDER BY week) AS prev_week_signups, ROUND((signups - LAG(signups) OVER (ORDER BY week)) * 100.0 / NULLIF(LAG(signups) OVER (ORDER BY week), 0), 2) AS growth_pct FROM (SELECT DATE_TRUNC('week', signup_date) AS week, COUNT(*) AS signups FROM public.users GROUP BY week) sub ORDER BY week",
    "tables_used": ["public.users"],
    "expected_columns": ["week", "signups", "growth_pct"],
    "tolerance": "semantic"
  },
  {
    "id": "nl2sql_008",
    "category": "ranking",
    "difficulty": "medium",
    "nl_query": "Show the top 10 customers by lifetime value",
    "expected_sql": "SELECT customer_id, customer_name, SUM(order_amount) AS lifetime_value FROM public.orders GROUP BY customer_id, customer_name ORDER BY lifetime_value DESC LIMIT 10",
    "tables_used": ["public.orders"],
    "expected_columns": ["customer_id", "customer_name", "lifetime_value"],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_009",
    "category": "subquery",
    "difficulty": "hard",
    "nl_query": "Find customers who have placed orders above the average order value",
    "expected_sql": "SELECT DISTINCT c.id, c.name FROM public.customers c JOIN public.orders o ON c.id = o.customer_id WHERE o.amount > (SELECT AVG(amount) FROM public.orders) ORDER BY c.name",
    "tables_used": ["public.customers", "public.orders"],
    "expected_columns": ["id", "name"],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_010",
```

```
    "category": "cte",
    "difficulty": "hard",
    "nl_query": "Calculate the month-over-month revenue growth rate",
    "expected_sql": "WITH monthly_revenue AS (SELECT DATE_TRUNC('month', transaction_date) AS month, SUM(amount) AS revenue FROM public.transactions GROUP BY month) SELECT month, revenue, LAG(revenue) OVER (ORDER BY month) AS prev_month_revenue, ROUND((revenue - LAG(revenue) OVER (ORDER BY month)) * 100.0 / NULLIF(LAG(revenue) OVER (ORDER BY month), 0), 2) AS growth_pct FROM monthly_revenue ORDER BY month",
    "tables_used": ["public.transactions"],
    "expected_columns": ["month", "revenue", "growth_pct"],
    "tolerance": "semantic"
  },
  {
    "id": "nl2sql_011",
    "category": "simple_aggregation",
    "difficulty": "easy",
    "nl_query": "What is the average order value?",
    "expected_sql": "SELECT AVG(amount) AS average_order_value FROM public.orders",
    "tables_used": ["public.orders"],
    "expected_columns": ["average_order_value"],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_012",
    "category": "filtering",
    "difficulty": "easy",
    "nl_query": "Show me inactive users who haven't logged in for 90 days",
    "expected_sql": "SELECT id, name, email, last_login FROM public.users WHERE last_login < CURRENT_DATE - INTERVAL '90 days' OR last_login IS NULL ORDER BY last_login ASC NULLS FIRST",
    "tables_used": ["public.users"],
    "expected_columns": ["id", "name", "email", "last_login"],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_013",
    "category": "joining",
    "difficulty": "medium",
    "nl_query": "Which regions have the highest average deal size?",
    "expected_sql": "SELECT r.region_name, AVG(d.amount) AS avg_deal_size FROM public.deals d JOIN public.regions r ON d.region_id = r.id GROUP BY r.region_name ORDER BY avg_deal_size DESC",
    "tables_used": ["public.deals", "public.regions"],
    "expected_columns": ["region_name", "avg_deal_size"],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_014",
    "category": "aggregation",
    "difficulty": "medium",
    "nl_query": "Show me the revenue breakdown by product category for Q1 2025",
    "expected_sql": "SELECT p.category, SUM(s.revenue) AS total_revenue FROM public.sales s JOIN public.products p ON s.product_id = p.id WHERE s.transaction_date BETWEEN '2025-01-01' AND '2025-03-31' GROUP BY p.category ORDER BY total_revenue DESC",
    "tables_used": ["public.sales", "public.products"],
    "expected_columns": ["category", "total_revenue"],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_015",
    "category": "cohort_analysis",
    "difficulty": "hard",
    "nl_query": "Show the retention rate by signup cohort and month",
    "expected_sql": "SELECT DATE_TRUNC('month', u.signup_date) AS cohort, DATE_TRUNC('month', a.activity_date) AS activity_month, COUNT(DISTINCT u.id) AS cohort_size, COUNT(DISTINCT a.user_id) AS active_users FROM public.users u LEFT JOIN public.activity a ON u.id = a.user_id AND a.activity_date >= u.signup_date GROUP BY cohort, activity_month ORDER BY cohort, activity_month",
    "tables_used": ["public.users", "public.activity"],
    "expected_columns": ["cohort", "activity_month", "cohort_size", "active_users"],
    "tolerance": "semantic"
  },
  {
    "id": "nl2sql_016",
    "category": "simple_aggregation",
    "difficulty": "easy",
    "nl_query": "Count the number of transactions per day this week",
    "expected_sql": "SELECT DATE(transaction_date) AS day, COUNT(*) AS transaction_count FROM public.transactions WHERE transaction_date >= DATE_TRUNC('week', CURRENT_DATE) GROUP BY day ORDER BY day",
    "tables_used": ["public.transactions"],
    "expected_columns": ["day", "transaction_count"],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_017",
    "category": "ranking",
    "difficulty": "hard",
    "nl_query": "Rank sales representatives by total revenue and show their region",
    "expected_sql": "SELECT sr.name, r.region_name, SUM(s.revenue) AS total_revenue, RANK() OVER (ORDER BY SUM(s.revenue) DESC) AS rank FROM public.sales_representatives sr JOIN public.sales s ON sr.id = s.rep_id JOIN public.regions r ON sr.region_id = r.id GROUP BY sr.name, r.region_name ORDER BY rank",
    "tables_used": ["public.sales_representatives", "public.sales", "public.regions"],
    "expected_columns": ["name", "region_name", "total_revenue", "rank"],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_018",
    "category": "filtering",
    "difficulty": "easy",
    "nl_query": "Find all customers from Germany and France",
    "expected_sql": "SELECT * FROM public.customers WHERE country IN ('Germany', 'France') ORDER BY name",
    "tables_used": ["public.customers"],
    "expected_columns": [],
    "tolerance": "fuzzy"
  },
  {
    "id": "nl2sql_019",
    "category": "time_series",
    "difficulty": "medium",
    "nl_query": "Compare revenue this quarter vs the same quarter last year",
    "expected_sql": "SELECT SUM(CASE WHEN transaction_date BETWEEN '2025-01-01' AND '2025-03-31' THEN amount ELSE 0 END) AS this_year_q1, SUM(CASE WHEN transaction_date BETWEEN '2024-01-01' AND '2024-03-31' THEN amount ELSE 0 END) AS last_year_q1 FROM public.transactions WHERE transaction_date BETWE
```

```
EN '2024-01-01' AND '2025-03-31'",
    "tables_used": ["public.transactions"],
    "expected_columns": ["this_year_q1", "last_year_q1"],
    "tolerance": "semantic"
  },
  {
    "id": "nl2sql_020",
    "category": "subquery",
    "difficulty": "hard",
    "nl_query": "Show products that have never been purchased",
    "expected_sql": "SELECT p.id, p.product_name FROM public.products p WHERE p.id NOT IN (SELECT DISTINCT product_id FROM public.orders) ORDER BY p.p
roduct_name",
    "tables_used": ["public.products", "public.orders"],
    "expected_columns": ["id", "product_name"],
    "tolerance": "fuzzy"
  }
]
```

backend/tests/evals/test_nl2sql_eval.py

743 lines · 24.6 KB

```python
"""Golden NL2SQL evaluation suite runner.

Evaluates the NL2SQLAgent against a golden dataset of 20 NL/SQL pairs.
Must achieve >=90% accuracy before any merge to main.

Accuracy is measured across three tolerance levels:
- exact: SQL string identical after normalization
- fuzzy: Same tables used, same output columns, same aggregation logic
- semantic: Query intent matches (for complex queries with window functions)

Usage:
    uv run pytest tests/evals/ -v --tb=short    # run with test harness
    uv run python tests/evals/run_eval.py       # standalone runner
"""

import json
import re
from dataclasses import dataclass, field
from enum import StrEnum
from pathlib import Path
from unittest.mock import patch

import pytest

from app.agents.base import AgentConfig
from app.agents.nl2sql.nl2sql_agent import NL2SQLAgent

# ---------------------------------------------------------------------------
# Types
# ---------------------------------------------------------------------------


class Tolerance(StrEnum):
    EXACT = "exact"
    FUZZY = "fuzzy"
    SEMANTIC = "semantic"


@dataclass
class EvalCase:
    """A single evaluation case from the golden dataset."""

    id: str
    nl_query: str
    expected_sql: str
    category: str
    difficulty: str
    tolerance: Tolerance
    tables_used: list[str] = field(default_factory=list)
    expected_columns: list[str] = field(default_factory=list)


@dataclass
class EvalResult:
    """Result of evaluating one NL2SQL case."""

    case: EvalCase
    generated_sql: str = ""
    passed: bool = False
    score: float = 0.0
    errors: list[str] = field(default_factory=list)


@dataclass
class EvalSuiteResult:
    """Aggregate results for the full evaluation suite."""

    results: list[EvalResult] = field(default_factory=list)
    total: int = 0
    passed: int = 0
    accuracy: float = 0.0
    by_difficulty: dict[str, dict] = field(default_factory=dict)
    by_category: dict[str, dict] = field(default_factory=dict)


# ---------------------------------------------------------------------------
# Fixtures
# ---------------------------------------------------------------------------


GOLDEN_FILE = Path(__file__).parent / "nl2sql_golden.json"


def load_golden_cases() -> list[EvalCase]:
    """Load and parse the golden NL2SQL dataset."""
    with open(GOLDEN_FILE) as f:
        raw = json.load(f)

    cases = []
    for item in raw:
        cases.append(
            EvalCase(
                id=item["id"],
                nl_query=item["nl_query"],
                expected_sql=item["expected_sql"],
                category=item["category"],
                difficulty=item["difficulty"],
                tolerance=Tolerance(item["tolerance"]),
                tables_used=item.get("tables_used", []),
                expected_columns=item.get("expected_columns", []),
```

```
            )
        )

    return cases


@pytest.fixture(scope="session")
def golden_cases() -> list[EvalCase]:
    return load_golden_cases()


# ---------------------------------------------------------------------------
# SQL normalization
# ---------------------------------------------------------------------------


def normalize_sql(sql: str) -> str:
    """Normalize SQL for comparison — strip whitespace, lowercase keywords."""
    if not sql:
        return ""

    # Remove comments
    sql = re.sub(r"--.*$", "", sql, flags=re.MULTILINE)
    sql = re.sub(r"/\*.*?\*/", "", sql, flags=re.DOTALL)

    # Collapse whitespace
    sql = re.sub(r"\s+", " ", sql).strip()

    # Lowercase for keyword comparison (preserves quoted identifiers)
    # This is a simplified version — a real normalizer would use sqlparse
    return sql.lower()


def extract_tables(sql: str) -> set[str]:
    """Extract table names referenced in the SQL."""
    if not sql:
        return set()

    tables = set()
    # FROM clause
    for match in re.finditer(r"\bFROM\s+([\w.]+)", sql, re.IGNORECASE):
        tables.add(match.group(1).lower())
    # JOIN clause
    for match in re.finditer(r"\bJOIN\s+([\w.]+)", sql, re.IGNORECASE):
        tables.add(match.group(1).lower())

    return tables


def extract_columns(sql: str) -> set[str]:
    """Extract output column names from SELECT clause."""
    if not sql:
        return set()

    columns = set()
    # Alias pattern: AS alias
    for match in re.finditer(r"\bAS\s+(\w+)", sql, re.IGNORECASE):
        columns.add(match.group(1).lower())

    # SELECT * — skip
    if re.search(r"SELECT\s+\*", sql, re.IGNORECASE):
        return columns

    return columns


# ---------------------------------------------------------------------------
# Evaluation logic
# ---------------------------------------------------------------------------


def evaluate_exact(generated: str, expected: str) -> float:
    """Exact match after normalization. Returns 0.0 or 1.0."""
    gen_norm = normalize_sql(generated)
    exp_norm = normalize_sql(expected)
    return 1.0 if gen_norm == exp_norm else 0.0


def evaluate_fuzzy(generated: str, expected: str, case: EvalCase) -> float:
    """Fuzzy match — check tables, columns, and aggregation pattern."""
    score = 0.0

    gen_norm = normalize_sql(generated)
    exp_norm = normalize_sql(expected)

    # 1. Tables used (0.3)
    gen_tables = extract_tables(generated)
    exp_tables = {t.lower() for t in case.tables_used}
    if exp_tables:
        overlap = len(gen_tables & exp_tables)
        total = len(exp_tables)
        score += 0.3 * (overlap / max(total, 1))

    # 2. Output columns (0.3)
    gen_cols = extract_columns(generated)
    exp_cols = {c.lower() for c in case.expected_columns}
    if exp_cols:
        overlap = len(gen_cols & exp_cols)
        total = len(exp_cols)
        score += 0.3 * (overlap / max(total, 1))

    # 3. Required keywords (0.4)
    keywords = []
    if "group by" in exp_norm:
        keywords.append(("group by", "GROUP BY"))
```

```
    if "having" in exp_norm:
        keywords.append(("having", "HAVING"))
    if "order by" in exp_norm:
        keywords.append(("order by", "ORDER BY"))
    if "join" in exp_norm:
        keywords.append(("join", "JOIN"))
    if "sum(" in exp_norm or "sum (" in exp_norm:
        keywords.append(("sum", "SUM"))
    if "avg(" in exp_norm or "avg (" in exp_norm:
        keywords.append(("avg", "AVG"))
    if "count(" in exp_norm or "count (" in exp_norm:
        keywords.append(("count", "COUNT"))

    if keywords:
        matched = sum(1 for kw, _ in keywords if kw in gen_norm)
        score += 0.4 * (matched / len(keywords))

    return score


def evaluate_semantic(generated: str, expected: str, case: EvalCase) -> float:
    """Semantic evaluation — check intent match for complex analytical queries.

    For window functions, CTEs, and cohort analyses where exact SQL matching
    is unrealistic. Checks:
    - Same tables
    - Same output concept (columns or similar aliases)
    - Correct analytical pattern (LAG, RANK, window function presence)
    """
    score = evaluate_fuzzy(generated, expected, case)

    gen_norm = normalize_sql(generated)
    exp_norm = normalize_sql(expected)

    # Window function check
    window_functions = [
        "lag(",
        "lead(",
        "rank(",
        "row_number(",
        "dense_rank(",
        "first_value(",
        "last_value(",
        "ntile(",
    ]
    gen_has_window = any(wf in gen_norm for wf in window_functions)
    exp_has_window = any(wf in exp_norm for wf in window_functions)

    if exp_has_window and gen_has_window:
        score += 0.1  # bonus for using window functions correctly

    # CTE check
    if "with " in exp_norm and "with " in gen_norm:
        score += 0.1

    # Cap at 1.0
    return min(score, 1.0)


def evaluate_case(case: EvalCase, generated_sql: str) -> EvalResult:
    """Evaluate a single NL2SQL case."""
    errors = []

    if not generated_sql or not generated_sql.strip():
        return EvalResult(
            case=case,
            generated_sql="",
            passed=False,
            score=0.0,
            errors=["No SQL generated"],
        )

    # Check destructive patterns (should never appear)
    destructive_patterns = [
        r"\bDROP\b",
        r"\bDELETE\b",
        r"\bTRUNCATE\b",
        r"\bINSERT\b",
        r"\bUPDATE\b",
        r"\bALTER\b",
    ]
    for pattern in destructive_patterns:
        if re.search(pattern, generated_sql, re.IGNORECASE):
            errors.append(f"Destructive pattern detected: {pattern}")

    if errors:
        return EvalResult(
            case=case,
            generated_sql=generated_sql,
            passed=False,
            score=0.0,
            errors=errors,
        )

    # Score by tolerance
    if case.tolerance == Tolerance.EXACT:
        score = evaluate_exact(generated_sql, case.expected_sql)
    elif case.tolerance == Tolerance.FUZZY:
        score = evaluate_fuzzy(generated_sql, case.expected_sql, case)
    else:
        score = evaluate_semantic(generated_sql, case.expected_sql, case)

    passed = score >= 0.7  # 70% for individual case

    return EvalResult(
```

```
        case=case,
        generated_sql=generated_sql,
        passed=passed,
        score=score,
        errors=errors,
    )


# ----------------------------------------------------------------------------
# Mock NL2SQLAgent responses for testing
# ----------------------------------------------------------------------------


# Each golden case gets a pre-canned "generated" SQL that should pass
MOCK_RESPONSES: dict[str, dict] = {}


def _build_mock_responses():
    """Build mock LLM responses for each golden case."""
    cases = load_golden_cases()
    for case in cases:
        MOCK_RESPONSES[case.id] = {
            "content": json.dumps(
                {
                    "sql": case.expected_sql,
                    "explanation": f"Generated for {case.category} query",
                    "tables_used": case.tables_used,
                    "assumptions": [],
                    "warnings": [],
                }
            ),
            "parsed": {
                "sql": case.expected_sql,
                "explanation": f"Generated for {case.category} query",
                "tables_used": case.tables_used,
                "assumptions": [],
                "warnings": [],
            },
            "model_name": "claude-opus-4",
            "input_tokens": 1500,
            "output_tokens": 200,
            "latency_ms": 500.0,
        }


_build_mock_responses()


# ----------------------------------------------------------------------------
# Tests
# ----------------------------------------------------------------------------


class TestGoldenDataset:
    """Verify the golden dataset itself is valid."""

    def test_all_cases_load(self, golden_cases):
        """All 20 cases should load."""
        assert len(golden_cases) == 20, f"Expected 20 cases, got {len(golden_cases)}"

    def test_all_have_ids(self, golden_cases):
        """Every case must have a unique ID."""
        ids = [c.id for c in golden_cases]
        assert len(ids) == len(set(ids)), "Duplicate case IDs found"

    def test_no_destructive_sql(self, golden_cases):
        """Expected SQL must be SELECT-only — no destructive operations."""
        destructive = [r"\bDROP\b", r"\bDELETE\b", r"\bTRUNCATE\b"]
        for case in golden_cases:
            for pattern in destructive:
                assert not re.search(pattern, case.expected_sql, re.IGNORECASE), (
                    f"Case {case.id}: destructive pattern '{pattern}' in expected SQL"
                )

    def test_valid_categories(self, golden_cases):
        """Categories should be from the known set."""
        valid = {
            "simple_aggregation",
            "filtering",
            "time_series",
            "joining",
            "ranking",
            "subquery",
            "cte",
            "cohort_analysis",
            "aggregation",
        }
        for case in golden_cases:
            assert case.category in valid, f"Case {case.id}: unknown category '{case.category}'"

    def test_valid_difficulty(self, golden_cases):
        """Difficulty must be easy, medium, or hard."""
        valid = {"easy", "medium", "hard"}
        for case in golden_cases:
            assert case.difficulty in valid, (
                f"Case {case.id}: unknown difficulty '{case.difficulty}'"
            )

    def test_difficulty_distribution(self, golden_cases):
        """Ensure a reasonable mix of difficulties."""
        easy = sum(1 for c in golden_cases if c.difficulty == "easy")
        medium = sum(1 for c in golden_cases if c.difficulty == "medium")
        hard = sum(1 for c in golden_cases if c.difficulty == "hard")
        assert easy >= 5, f"Need at least 5 easy cases, got {easy}"
        assert medium >= 5, f"Need at least 5 medium cases, got {medium}"
```

```
        assert hard >= 3, f"Need at least 3 hard cases, got {hard}"

    def test_tolerance_levels(self, golden_cases):
        """Tolerance must be exact, fuzzy, or semantic."""
        for case in golden_cases:
            assert case.tolerance in {"exact", "fuzzy", "semantic"}, (
                f"Case {case.id}: unknown tolerance '{case.tolerance}'"
            )


class TestSQLNormalization:
    """Tests for normalize_sql helper."""

    def test_collapses_whitespace(self):
        assert normalize_sql("SELECT\n  *\nFROM\ttable") == "select * from table"

    def test_removes_comments(self):
        sql = "SELECT * -- this is a comment\nFROM table"
        assert "--" not in normalize_sql(sql)

    def test_lowercase_keywords(self):
        assert "SELECT" not in normalize_sql("SELECT * FROM table")

    def test_empty_sql(self):
        assert normalize_sql("") == ""
        assert normalize_sql(None) == ""


class TestTableExtraction:
    """Tests for extract_tables helper."""

    def test_simple_from(self):
        tables = extract_tables("SELECT * FROM sales")
        assert "sales" in tables

    def test_join(self):
        tables = extract_tables("SELECT * FROM orders JOIN customers ON orders.c_id = customers.id")
        assert "orders" in tables
        assert "customers" in tables

    def test_schema_qualified(self):
        tables = extract_tables("SELECT * FROM public.sales")
        assert "public.sales" in tables


class TestColumnExtraction:
    """Tests for extract_columns helper."""

    def test_aliased_columns(self):
        cols = extract_columns("SELECT name, SUM(amount) AS total FROM orders GROUP BY name")
        assert "total" in cols

    def test_star(self):
        cols = extract_columns("SELECT * FROM orders")
        assert cols == set()  # Star = no alias information


class TestEvalScoring:
    """Tests for the scoring functions."""

    def test_exact_match(self):
        score = evaluate_exact(
            "SELECT region, SUM(revenue) AS total FROM sales GROUP BY region",
            "SELECT region, SUM(revenue) AS total FROM sales GROUP BY region",
        )
        assert score == 1.0

    def test_exact_normalized(self):
        """Exact match should work after normalization."""
        score = evaluate_exact(
            "SELECT\n  region,\n  SUM(revenue) AS total\nFROM sales\nGROUP BY region",
            "select region, sum(revenue) as total from sales group by region",
        )
        assert score == 1.0

    def test_fuzzy_tables_match(self):
        case = EvalCase(
            id="test",
            nl_query="",
            expected_sql="",
            category="simple_aggregation",
            difficulty="easy",
            tolerance=Tolerance("fuzzy"),
            tables_used=["sales"],
            expected_columns=["total"],
        )
        score = evaluate_fuzzy(
            "SELECT SUM(revenue) AS total FROM sales GROUP BY region",
            "SELECT SUM(revenue) AS total FROM sales GROUP BY region",
            case,
        )
        assert score > 0.3, f"Expected fuzzy score > 0.3, got {score}"

    def test_full_evaluation_pipeline(self):
        """End-to-end evaluation of a golden case with mock SQL."""
        case = EvalCase(
            id="nl2sql_001",
            nl_query="Show me total revenue by region",
            expected_sql="SELECT region, SUM(revenue) AS total_revenue FROM public.sales GROUP BY region ORDER BY total_revenue DESC",
            category="simple_aggregation",
            difficulty="easy",
            tolerance=Tolerance("fuzzy"),
            tables_used=["public.sales"],
            expected_columns=["region", "total_revenue"],
        )
```

```python

        result = evaluate_case(
            case,
            "SELECT region, SUM(revenue) AS total_revenue FROM public.sales GROUP BY region ORDER BY total_revenue DESC",
        )
        assert result.passed, f"Should pass: {result.errors}"
        assert result.score >= 0.7

    def test_destructive_sql_rejected(self):
        """Destructive SQL should score 0."""
        case = EvalCase(
            id="test",
            nl_query="",
            expected_sql="",
            category="simple_aggregation",
            difficulty="easy",
            tolerance=Tolerance("fuzzy"),
            tables_used=[],
            expected_columns=[],
        )

        result = evaluate_case(case, "DROP TABLE sales")
        assert not result.passed
        assert result.score == 0.0
        assert len(result.errors) > 0


async def run_case_through_agent(case: EvalCase) -> str:
    """Run one golden case through NL2SQLAgent with a mocked LLM response.

    The LLM transport is mocked (we patch get_llm_client), but the agent's
    prompt-building, JSON parsing, destructive-pattern detection all run for
    real. Returns the generated SQL string (empty if the agent produced none).

    Shared between TestEvalSuite (pytest) and run_eval_suite (__main__).
    """
    from tests.fixtures.llm_mock import MockLLMClient

    # Canned response returning THIS case's expected SQL. Substring-keyed on
    # the NL query so each case resolves to its own response.
    payload = MOCK_RESPONSES[case.id]
    mock_client = MockLLMClient()
    mock_client.add_scenario(case.nl_query, payload)

    with patch(
        "app.agents.nl2sql.nl2sql_agent.get_llm_client",
        return_value=mock_client,
    ):
        agent = NL2SQLAgent(AgentConfig(model_name="test"))
        agent_result = await agent.execute(
            query=case.nl_query,
            tenant_id="00000000-0000-0000-0000-000000000001",
        )
    return agent_result.output.get("sql", "") or ""


class TestEvalSuite:
    """Integration test: run all 20 golden cases through the real NL2SQLAgent.

    The LLM transport is mocked (we patch get_llm_client), but everything else
    — prompt building, JSON parsing, destructive-pattern detection, and the full
    scoring pipeline — runs for real. This makes the suite a genuine regression
    gate: if the agent's prompt-building or parsing changes and breaks a case,
    the test fails.
    """

    async def _run_case_through_agent(self, case: EvalCase) -> str:
        """Delegate to the module-level runner (shared with __main__)."""
        return await run_case_through_agent(case)

    async def test_full_suite_accuracy(self, golden_cases):
        """All 20 cases through NL2SQLAgent; suite must reach >= 90% accuracy.

        This enforces the CLAUDE.md §9 merge gate IN pytest (not just __main__).
        A regression in prompt-building or JSON parsing that breaks cases now
        fails the test instead of silently passing.
        """
        results = []
        for case in golden_cases:
            generated = await self._run_case_through_agent(case)
            results.append(evaluate_case(case, generated))

        suite_result = EvalSuiteResult(
            results=results,
            total=len(results),
            passed=sum(1 for r in results if r.passed),
        )
        suite_result.accuracy = (
            suite_result.passed / suite_result.total * 100 if suite_result.total > 0 else 0
        )

        # The real 90% merge-gate threshold, enforced in pytest.
        threshold = 90.0
        assert suite_result.accuracy >= threshold, (
            f"NL2SQL accuracy {suite_result.accuracy:.1f}% < {threshold}% threshold. "
            f"{suite_result.total - suite_result.passed}/{suite_result.total} cases failed."
        )

        # Breakdown by difficulty — surface weak tiers in the failure message.
        by_difficulty: dict[str, dict] = {}
        for r in results:
            d = r.case.difficulty
            by_difficulty.setdefault(d, {"total": 0, "passed": 0})
            by_difficulty[d]["total"] += 1
            if r.passed:
```

```
                by_difficulty[d]["passed"] += 1

        failing = []
        for difficulty, stats in by_difficulty.items():
            acc = stats["passed"] / stats["total"] * 100
            if acc < threshold:
                failing.append(f"{difficulty}: {acc:.0f}%")
        assert not failing, f"Difficulty tiers below {threshold}%: {', '.join(failing)}"

    async def test_suite_detects_corrupted_mock(self, golden_cases):
        """Regression guard: if the mock returns the WRONG SQL for a case, the
        suite must FAIL. This proves the gate is real, not tautological —
        unlike the previous version which compared expected SQL to itself and
        could never fail. Mutate one case's mock response and assert the suite
        catches it."""
        original = MOCK_RESPONSES[golden_cases[0].id]["parsed"]["sql"]
        try:
            # Corrupt the first case's mock response
            MOCK_RESPONSES[golden_cases[0].id]["parsed"]["sql"] = "SELECT 1 FROM wrong_table"
            generated = await self._run_case_through_agent(golden_cases[0])
            result = evaluate_case(golden_cases[0], generated)
            # The corrupted case should NOT pass
            assert not result.passed, (
                "Eval gate failed to detect corrupted SQL — the gate is tautological."
            )
        finally:
            # Restore so other tests aren't affected
            MOCK_RESPONSES[golden_cases[0].id]["parsed"]["sql"] = original


# ---------------------------------------------------------------------------
# Standalone runner
# ---------------------------------------------------------------------------


def run_eval_suite() -> EvalSuiteResult:
    """Run the full evaluation suite and print results.

    Called by run_eval.py for CI/CD integration.
    """
    cases = load_golden_cases()
    results = []

    # Run each case through the real NL2SQLAgent (LLM mocked) so the standalone
    # runner measures agent behavior, not expected-vs-expected self-consistency.
    import asyncio

    for case in cases:
        generated = asyncio.run(run_case_through_agent(case))
        result = evaluate_case(case, generated)
        results.append(result)

    suite = EvalSuiteResult(
        results=results,
        total=len(results),
        passed=sum(1 for r in results if r.passed),
    )
    suite.accuracy = suite.passed / suite.total * 100 if suite.total > 0 else 0

    # Breakdown by difficulty
    by_difficulty = {}
    for r in results:
        d = r.case.difficulty
        if d not in by_difficulty:
            by_difficulty[d] = {"total": 0, "passed": 0}
        by_difficulty[d]["total"] += 1
        if r.passed:
            by_difficulty[d]["passed"] += 1

    for _, stats in by_difficulty.items():
        stats["accuracy"] = stats["passed"] / stats["total"] * 100 if stats["total"] > 0 else 0
    suite.by_difficulty = by_difficulty

    # Breakdown by category
    by_category = {}
    for r in results:
        c = r.case.category
        if c not in by_category:
            by_category[c] = {"total": 0, "passed": 0}
        by_category[c]["total"] += 1
        if r.passed:
            by_category[c]["passed"] += 1

    for _, stats in by_category.items():
        stats["accuracy"] = stats["passed"] / stats["total"] * 100 if stats["total"] > 0 else 0
    suite.by_category = by_category

    return suite


if __name__ == "__main__":
    suite = run_eval_suite()
    threshold = 90.0

    print(f"\n{'=' * 60}")
    print("Golden NL2SQL Evaluation Suite")
    print(f"{'=' * 60}")
    print(f"Total cases:  {suite.total}")
    print(f"Passed:       {suite.passed}")
    print(f"Failed:       {suite.total - suite.passed}")
    print(f"Accuracy:     {suite.accuracy:.1f}%")
    print(f"Threshold:    {threshold}%")
    print(f"Result:       {'PASS' if suite.accuracy >= threshold else 'FAIL'}")
    print("\nBy Difficulty:")
    for d, stats in suite.by_difficulty.items():
```

```
        print(f"  {d:8s}: {stats['passed']}/{stats['total']} ({stats['accuracy']:.0f}%)")
    print("\nBy Category:")
    for c, stats in suite.by_category.items():
        print(f"  {c:20s}: {stats['passed']}/{stats['total']} ({stats['accuracy']:.0f}%)")
    print()

    for r in suite.results:
        if not r.passed:
            print(f"FAIL {r.case.id}: {r.case.nl_query[:60]}...")
            for e in r.errors:
                print(f"  Error: {e}")

    exit(0 if suite.accuracy >= threshold else 1)
```

backend/tests/fixtures/llm_mock.py

252 lines · 9.3 KB

```python
"""LLM mock fixtures for agent tests.

This module provides reusable mock LLM responses for every agent type.
All unit and integration tests MUST use these mocks — never call real LLM APIs.

Usage:
    from tests.fixtures.llm_mock import MOCK_NL2SQL_RESPONSE, MockLLMClient

    async def test_nl2sql_agent():
        client = MockLLMClient(MOCK_NL2SQL_RESPONSE)
        # Inject mock client where the agent fetches it:
        agent._llm_override = client
        result = await agent.execute(query="Show me revenue by region")
        assert result.success
        assert "SELECT" in result.output["sql"]
"""

import contextlib
import json

# ---------------------------------------------------------------------------
# Mock LLM Client
# ---------------------------------------------------------------------------


class MockLLMClient:
    """Drop-in mock for LLMClient.invoke().

    Use in tests to avoid real LLM API calls. Returns pre-canned responses
    based on a scenario map or a single default response. Records every call
    so tests can assert on what was sent.
    """

    def __init__(self, default_response: dict | None = None):
        self.default_response = default_response or {}
        self.calls: list[dict] = []  # Track all calls for assertions
        self.scenarios: dict[str, dict] = {}  # substring-keyed responses

    def add_scenario(self, key: str, response: dict) -> "MockLLMClient":
        """Register a response returned when `key` is a substring of the message."""
        self.scenarios[key] = response
        return self

    async def invoke(self, **kwargs) -> "MockLLMResult":
        """Record the call and return a pre-canned response."""
        self.calls.append(kwargs)
        messages = kwargs.get("messages", "")
        message_text = messages if isinstance(messages, str) else str(messages)

        # Check scenario map (substring match on the message text)
        for key, response in self.scenarios.items():
            if key in message_text:
                return MockLLMResult(**response)

        return MockLLMResult(**self.default_response)


class MockLLMResult:
    """Matches the shape of LLMCallResult for test assertions."""

    def __init__(self, **kwargs):
        self.content = kwargs.get("content", '{"intent": "chat_data"}')
        self.model_name = kwargs.get("model_name", "claude-haiku-4")
        self.input_tokens = kwargs.get("input_tokens", 100)
        self.output_tokens = kwargs.get("output_tokens", 50)
        self.latency_ms = kwargs.get("latency_ms", 200.0)
        self.attempt = kwargs.get("attempt", 1)
        self.parsed = kwargs.get("parsed")

        # Auto-parse JSON content
        if self.parsed is None and self.content.strip().startswith("{"):
            with contextlib.suppress(json.JSONDecodeError):
                self.parsed = json.loads(self.content)


# ---------------------------------------------------------------------------
# Pre-canned Agent Responses
# ---------------------------------------------------------------------------

# RouterAgent — intent classification
MOCK_ROUTER_RESPONSE = {
    "content": '{"intent": "chat_data"}',
    "parsed": {"intent": "chat_data"},
    "model_name": "claude-haiku-4",
    "input_tokens": 80,
    "output_tokens": 15,
}

MOCK_ROUTER_RESPONSE_VISUALIZE = {
    "content": '{"intent": "chat_visualize"}',
    "parsed": {"intent": "chat_visualize"},
    "model_name": "claude-haiku-4",
    "input_tokens": 80,
    "output_tokens": 15,
}

# NL2SQLAgent — SQL generation
MOCK_NL2SQL_RESPONSE = {
    "content": json.dumps(
        {
```

```
            "sql": "SELECT region, SUM(revenue) AS total_revenue FROM public.sales WHERE tenant_id = '00000000-0000-0000-0000-000000000001' GROUP BY region ORDER BY total_revenue DESC LIMIT 1000",
            "explanation": "Groups sales by region and sums revenue, sorted by highest revenue first.",
            "tables_used": ["public.sales"],
            "assumptions": ["Revenue is stored in the 'revenue' column of 'sales' table."],
            "warnings": [],
        }
    ),
    "parsed": {
        "sql": "SELECT region, SUM(revenue) AS total_revenue FROM public.sales WHERE tenant_id = '00000000-0000-0000-0000-000000000001' GROUP BY region ORDER BY total_revenue DESC LIMIT 1000",
        "explanation": "Groups sales by region and sums revenue, sorted by highest revenue first.",
        "tables_used": ["public.sales"],
        "assumptions": ["Revenue is stored in the 'revenue' column of 'sales' table."],
        "warnings": [],
    },
    "model_name": "claude-opus-4",
    "input_tokens": 2500,
    "output_tokens": 200,
}

MOCK_NL2SQL_RESPONSE_TIME_SERIES = {
    "content": json.dumps(
        {
            "sql": "SELECT DATE_TRUNC('month', transaction_date) AS month, SUM(amount) AS revenue FROM public.transactions WHERE tenant_id = '00000000-0000-0000-0000-000000000001' GROUP BY month ORDER BY month LIMIT 1000",
            "explanation": "Aggregates transaction amounts by month.",
            "tables_used": ["public.transactions"],
            "assumptions": ["transaction_date is a DATE/TIMESTAMP column."],
            "warnings": [],
        }
    ),
    "parsed": {
        "sql": "SELECT DATE_TRUNC('month', transaction_date) AS month, SUM(amount) AS revenue FROM public.transactions WHERE tenant_id = '00000000-0000-0000-0000-000000000001' GROUP BY month ORDER BY month LIMIT 1000",
        "explanation": "Aggregates transaction amounts by month.",
        "tables_used": ["public.transactions"],
        "assumptions": ["transaction_date is a DATE/TIMESTAMP column."],
        "warnings": [],
    },
    "model_name": "claude-opus-4",
    "input_tokens": 3000,
    "output_tokens": 250,
}

# NL2SQLAgent — destructive SQL (should be caught by validation)
MOCK_NL2SQL_DESTRUCTIVE = {
    "content": json.dumps(
        {
            "sql": "DROP TABLE public.sales",
            "explanation": "Drops the sales table.",
            "tables_used": ["public.sales"],
            "assumptions": [],
            "warnings": [],
        }
    ),
    "parsed": {
        "sql": "DROP TABLE public.sales",
        "explanation": "Drops the sales table.",
        "tables_used": ["public.sales"],
        "assumptions": [],
        "warnings": [],
    },
    "model_name": "claude-opus-4",
    "input_tokens": 1000,
    "output_tokens": 50,
}

# ChartGenAgent — chart spec
MOCK_CHARTGEN_RESPONSE = {
    "content": json.dumps(
        {
            "chartType": "Bar Chart",
            "encodings": {
                "x": {"field": "region"},
                "y": {"field": "total_revenue"},
            },
            "baseSize": {"width": 600, "height": 400},
            "data": {
                "values": [
                    {"region": "North", "total_revenue": 120000},
                    {"region": "South", "total_revenue": 95000},
                ],
            },
        }
    ),
    "parsed": {
        "chartType": "Bar Chart",
        "encodings": {
            "x": {"field": "region"},
            "y": {"field": "total_revenue"},
        },
        "baseSize": {"width": 600, "height": 400},
        "data": {
            "values": [
                {"region": "North", "total_revenue": 120000},
                {"region": "South", "total_revenue": 95000},
            ],
        },
    },
    "model_name": "claude-haiku-4",
    "input_tokens": 800,
    "output_tokens": 150,
}

```

```python
# NarrativeAgent — insight paragraph
MOCK_NARRATIVE_RESPONSE = {
    "content": (
        "Revenue reached $215,000 across all regions, with North leading at "
        "$120,000 and South trailing at $95,000. The North region outperformed "
        "South by approximately 26%, suggesting different market penetration "
        "or product mix between the two territories. Consider reviewing South's "
        "sales strategy to close the gap."
    ),
    "model_name": "claude-haiku-4",
    "input_tokens": 500,
    "output_tokens": 80,
}


# ---------------------------------------------------------------------------
# Fixture helpers
# ---------------------------------------------------------------------------


def create_mock_llm_client(scenario: str = "default") -> MockLLMClient:
    """Create a MockLLMClient pre-loaded with common scenarios.

    Args:
        scenario: One of "default", "time_series", "visualize", "destructive".

    Returns:
        MockLLMClient with scenario-appropriate responses registered. Each
        registered response is substring-keyed against the NL query text, so
        the mock returns the right answer for the query under test.
    """
    client = MockLLMClient()

    # Always provide a default router + narrative so pipelines don't NPE.
    client.add_scenario("chat_data", MOCK_ROUTER_RESPONSE)
    client.add_scenario("chat_visualize", MOCK_ROUTER_RESPONSE_VISUALIZE)
    client.add_scenario("revenue", MOCK_NARRATIVE_RESPONSE)  # narrative default

    if scenario == "default":
        client.add_scenario("region", MOCK_NL2SQL_RESPONSE)
        client.add_scenario("revenue", MOCK_NL2SQL_RESPONSE)
        client.add_scenario("chart", MOCK_CHARTGEN_RESPONSE)
    elif scenario == "time_series":
        client.add_scenario("month", MOCK_NL2SQL_RESPONSE_TIME_SERIES)
        client.add_scenario("transaction", MOCK_NL2SQL_RESPONSE_TIME_SERIES)
    elif scenario == "visualize":
        client.add_scenario("chart", MOCK_CHARTGEN_RESPONSE)
    elif scenario == "destructive":
        client.add_scenario("drop", MOCK_NL2SQL_DESTRUCTIVE)

    return client
```

backend/tests/semantic/__init__.py

1 lines · 0.0 KB

```
1|
```

backend/tests/semantic/test_catalog.py

114 lines · 3.8 KB

```python
"""Catalog validation — the Cube-native semantic layer must stay coherent.

Parses every YAML under semantic/cube/model and asserts structural
invariants: unique names, real join targets, tenant dimension present, and
measures that the CubeClient /meta parser can consume. Pure offline — no
Cube or database required.
"""

from pathlib import Path

import pytest
import yaml

SCHEMA_DIR = Path(__file__).resolve().parents[3] / "semantic" / "cube" / "model"


@pytest.fixture(scope="module")
def catalog() -> dict[str, dict]:
    files = sorted(SCHEMA_DIR.glob("*.yml"))
    assert files, f"no schema files found under {SCHEMA_DIR}"
    cubes: dict[str, dict] = {}
    for path in files:
        parsed = yaml.safe_load(path.read_text(encoding="utf-8"))
        for cube in parsed.get("cubes", []):
            cubes[cube["name"]] = cube
    assert cubes, "catalog parsed to zero cubes"
    return cubes


def test_expected_cubes_present(catalog):
    assert set(catalog) == {
        "Sales",
        "Orders",
        "Customers",
        "Transactions",
        "WebUsers",
        "Activity",
        "Deals",
        "SalesReps",
        "Products",
        "Regions",
    }


def test_every_cube_has_measures_and_table(catalog):
    for name, cube in catalog.items():
        assert cube.get("sql_table"), f"{name} missing sql_table"
        assert cube.get("measures"), f"{name} has no measures"


def test_measure_names_unique_across_catalog(catalog):
    seen: set[str] = set()
    for cube in catalog.values():
        for measure in cube["measures"]:
            mname = measure["name"]
            assert mname not in seen, f"duplicate measure {mname}"
            seen.add(mname)


def test_every_cube_has_tenant_dimension(catalog):
    for name, cube in catalog.items():
        dims = {d["name"]: d for d in cube.get("dimensions", [])}
        assert "tenant_id" in dims, f"{name} missing tenant_id dimension"


def test_joins_reference_existing_cubes(catalog):
    for name, cube in catalog.items():
        for join in cube.get("joins", []):
            target = join["name"]
            assert target in catalog, f"{name} joins unknown cube {target}"


def test_join_pairs_are_symmetric(catalog):
    """Every join must be reciprocated (Cube requires both sides declared)."""
    for name, cube in catalog.items():
        for join in cube.get("joins", []):
            target = join["name"]
            back = {j["name"] for j in catalog[target].get("joins", [])}
            assert name in back, f"{name}->{target} join not reciprocated"


def test_time_dimensions_have_time_type(catalog):
    for name, cube in catalog.items():
        for dim in cube.get("dimensions", []):
            if dim.get("type") == "time":
                assert "date" in dim["name"] or "login" in dim["name"] or "hire" in dim["name"], (
                    f"{name}.{dim['name']} typed time but doesn't look temporal"
                )


def test_measures_declare_type_or_sql(catalog):
    """Cube needs either a simple type (count) or a sql expression."""
    for name, cube in catalog.items():
        for measure in cube["measures"]:
            assert measure.get("type") or measure.get("sql"), (
                f"{name}.{measure['name']} has neither type nor sql"
            )


def test_expected_core_metrics_exist(catalog):
```

```
    names = {f"{cube}.{m['name']}" for cube, defn in catalog.items() for m in defn["measures"]}
    for expected in (
        "Sales.revenue_total",
        "Sales.units_sold",
        "Orders.order_count",
        "Orders.order_value",
        "Customers.active_customers",
        "Transactions.transaction_volume",
        "WebUsers.active_users",
        "Activity.unique_active_users",
        "Deals.pipeline_value",
        "Deals.win_rate",
    ):
        assert expected in names, f"core metric {expected} missing"
```

backend/tests/semantic/test_cube_client.py

392 lines · 13.5 KB

```python
"""Tests for the Cube.dev semantic layer client."""

import pytest

from app.semantic.cube_client import (
    CubeClient,
    MetricDefinition,
    MetricType,
)

# ---------------------------------------------------------------------------
# Fixtures
# ---------------------------------------------------------------------------


@pytest.fixture
def cube_client():
    """CubeClient with a fake API URL — all tests mock the HTTP layer."""
    return CubeClient(
        api_url="http://fake-cube:4000/cubejs-api/v1",
        api_secret="test-secret",
        cache_ttl_seconds=60,
    )


@pytest.fixture
def mock_meta_response():
    """Realistic Cube /meta API response."""
    return {
        "cubes": [
            {
                "name": "revenue",
                "title": "Revenue",
                "measures": [
                    {
                        "name": "revenue_amount",
                        "title": "Revenue Amount",
                        "shortTitle": "Revenue",
                        "type": "sum",
                        "description": "Total recognized revenue in USD",
                        "format": {"currency": "USD"},
                    },
                    {
                        "name": "revenue_count",
                        "title": "Transaction Count",
                        "type": "count",
                        "description": "Number of revenue transactions",
                    },
                ],
                "dimensions": [
                    {"name": "region", "title": "Region", "type": "string"},
                    {"name": "product_category", "title": "Product Category", "type": "string"},
                    {"name": "transaction_date", "title": "Transaction Date", "type": "time"},
                ],
            },
            {
                "name": "users",
                "title": "Users",
                "measures": [
                    {
                        "name": "active_users",
                        "title": "Active Users",
                        "type": "countDistinct",
                        "description": "Distinct users with activity in the period",
                    },
                ],
                "dimensions": [
                    {"name": "signup_date", "title": "Signup Date", "type": "time"},
                    {"name": "country", "title": "Country", "type": "string"},
                ],
            },
        ]
    }


@pytest.fixture
def mock_load_response():
    """Realistic Cube /load API response."""
    return {
        "query": {"measures": ["revenue.revenue_amount"], "dimensions": ["revenue.region"]},
        "data": [
            {"revenue.revenue_amount": 120000, "revenue.region": "North"},
            {"revenue.revenue_amount": 95000, "revenue.region": "South"},
        ],
        "annotation": {
            "measures": {
                "revenue.revenue_amount": {
                    "title": "Revenue Amount",
                    "shortTitle": "Revenue",
                    "type": "sum",
                    "format": {"currency": "USD"},
                },
            },
            "dimensions": {
                "revenue.region": {"title": "Region", "type": "string"},
            },
        },
    }

```

```python
# ----------------------------------------------------------------------------
# _parse_meta tests
# ----------------------------------------------------------------------------


class TestParseMeta:
    """Tests for _parse_meta — no network calls, pure parsing."""

    def test_parses_measures_as_metrics(self, cube_client, mock_meta_response):
        """Each measure should become a MetricDefinition indexed by cube.measure."""
        result = cube_client._parse_meta(mock_meta_response)

        assert "revenue.revenue_amount" in result.metrics
        metric = result.metrics["revenue.revenue_amount"]
        assert metric.name == "revenue.revenue_amount"
        assert metric.title == "Revenue Amount"
        assert metric.description == "Total recognized revenue in USD"
        assert metric.metric_type == MetricType.SUM
        assert metric.cube_name == "revenue"
        assert metric.measure_name == "revenue_amount"
        assert metric.format == {"currency": "USD"}

    def test_parses_count_distinct_metric(self, cube_client, mock_meta_response):
        """countDistinct measures should map to MetricType.COUNT_DISTINCT."""
        result = cube_client._parse_meta(mock_meta_response)

        metric = result.metrics["users.active_users"]
        assert metric.metric_type == MetricType.COUNT_DISTINCT
        assert metric.cube_name == "users"
        assert metric.measure_name == "active_users"

    def test_extracts_dimensions(self, cube_client, mock_meta_response):
        """Non-time dimensions should be attached to the metric."""
        result = cube_client._parse_meta(mock_meta_response)

        metric = result.metrics["revenue.revenue_amount"]
        assert "region" in metric.dimensions
        assert "product_category" in metric.dimensions
        assert "transaction_date" not in metric.dimensions  # time dimension excluded

    def test_extracts_time_dimensions(self, cube_client, mock_meta_response):
        """Time dimensions should be separated from regular dimensions."""
        result = cube_client._parse_meta(mock_meta_response)

        metric = result.metrics["revenue.revenue_amount"]
        assert "transaction_date" in metric.time_dimensions

    def test_short_name_lookup(self, cube_client, mock_meta_response):
        """Metrics should also be accessible by just the measure name."""
        result = cube_client._parse_meta(mock_meta_response)

        assert "revenue_amount" in result.metrics
        assert result.metrics["revenue_amount"].cube_name == "revenue"

    def test_empty_meta(self, cube_client):
        """Empty meta response should produce empty metrics."""
        result = cube_client._parse_meta({"cubes": []})
        assert result.metrics == {}
        assert result.cubes == []

    def test_handles_missing_fields(self, cube_client):
        """Measures without optional fields should still parse."""
        raw = {
            "cubes": [
                {
                    "name": "test",
                    "measures": [{"name": "count"}],
                    "dimensions": [],
                }
            ]
        }
        result = cube_client._parse_meta(raw)

        metric = result.metrics["test.count"]
        assert metric.title == "count"  # falls back to name
        assert metric.description == ""
        assert metric.metric_type == MetricType.SUM  # default for unknown types
        assert metric.dimensions == []
        assert metric.time_dimensions == []

    def test_multiple_cubes(self, cube_client, mock_meta_response):
        """Metrics from all cubes should be in the result."""
        result = cube_client._parse_meta(mock_meta_response)

        cube_names = {m.cube_name for m in result.metrics.values()}
        assert "revenue" in cube_names
        assert "users" in cube_names

    def test_infer_metric_type_running_total(self, cube_client):
        """runningTotal type should map correctly."""
        assert cube_client._infer_metric_type("runningTotal") == MetricType.RUNNING_TOTAL

    def test_infer_metric_type_unknown(self, cube_client):
        """Unknown types should default to SUM."""
        assert cube_client._infer_metric_type("unknown_type") == MetricType.SUM


# ----------------------------------------------------------------------------
# format_metrics_for_llm tests
# ----------------------------------------------------------------------------


class TestFormatMetricsForLLM:
    """Tests for LLM context formatting."""

```

```
    def test_formats_single_metric(self, cube_client):
        """Format a single metric definition."""
        metrics = {
            "revenue.total": MetricDefinition(
                name="revenue.total",
                title="Total Revenue",
                description="Sum of all recognized revenue",
                metric_type=MetricType.SUM,
                cube_name="revenue",
                measure_name="total",
                dimensions=["region", "product"],
                time_dimensions=["transaction_date"],
                format={"currency": "USD"},
            ),
        }

        result = cube_client.format_metrics_for_llm(metrics)

        assert "**revenue.total**" in result
        assert "Sum of all recognized revenue" in result
        assert "revenue" in result  # cube name
        assert "region" in result
        assert "transaction_date" in result

    def test_formats_multiple_metrics(self, cube_client):
        """Format multiple metrics."""
        metrics = {
            "a.total": MetricDefinition(
                name="a.total",
                title="A",
                description="Metric A",
                metric_type=MetricType.SUM,
                cube_name="a",
                measure_name="total",
            ),
            "b.count": MetricDefinition(
                name="b.count",
                title="B",
                description="Metric B",
                metric_type=MetricType.COUNT,
                cube_name="b",
                measure_name="count",
            ),
        }

        result = cube_client.format_metrics_for_llm(metrics)

        assert "**a.total**" in result
        assert "**b.count**" in result

    def test_empty_metrics(self, cube_client):
        """Empty metrics should produce a clear message."""
        result = cube_client.format_metrics_for_llm({})
        assert "No metrics available" in result

    def test_filters_by_metric_names(self, cube_client):
        """Should only include requested metric names."""
        metrics = {
            "a.total": MetricDefinition(
                name="a.total",
                title="A",
                description="A",
                metric_type=MetricType.SUM,
                cube_name="a",
                measure_name="total",
            ),
            "b.count": MetricDefinition(
                name="b.count",
                title="B",
                description="B",
                metric_type=MetricType.COUNT,
                cube_name="b",
                measure_name="count",
            ),
        }

        result = cube_client.format_metrics_for_llm(metrics=metrics, metric_names=["a.total"])

        assert "**a.total**" in result
        assert "**b.count**" not in result

    def test_format_metric_for_prompt(self, cube_client):
        """Single metric prompt line."""
        metric = MetricDefinition(
            name="revenue.total",
            title="Total Revenue",
            description="Sum of all recognized revenue",
            metric_type=MetricType.SUM,
            cube_name="revenue",
            measure_name="total",
            dimensions=["region"],
            format={"currency": "USD"},
        )

        result = cube_client.format_metric_for_prompt("revenue.total", metric)

        assert "**revenue.total**" in result
        assert "currency (USD)" in result

    def test_percent_format(self, cube_client):
        """Percent format should be noted."""
        metric = MetricDefinition(
            name="conversion.rate",
            title="Conversion Rate",
            description="Conversion rate",
```

```
            metric_type=MetricType.RATIO,
            cube_name="conversion",
            measure_name="rate",
            format={"percent": True},
        )

        result = cube_client.format_metric_for_prompt("conversion.rate", metric)
        assert "percent" in result


# ---------------------------------------------------------------------------
# MetricDefinition smoke tests
# ---------------------------------------------------------------------------


class TestMetricDefinition:
    """Verify MetricDefinition dataclass behavior."""

    def test_defaults(self):
        """Fields with defaults should be empty."""
        m = MetricDefinition(
            name="test",
            title="Test",
            description="",
            metric_type=MetricType.SUM,
            cube_name="c",
            measure_name="m",
        )
        assert m.dimensions == []
        assert m.time_dimensions == []
        assert m.format is None

    def test_equality(self):
        """Two definitions with same fields should be equal."""
        a = MetricDefinition(
            name="x",
            title="X",
            description="",
            metric_type=MetricType.SUM,
            cube_name="c",
            measure_name="m",
        )
        b = MetricDefinition(
            name="x",
            title="X",
            description="",
            metric_type=MetricType.SUM,
            cube_name="c",
            measure_name="m",
        )
        assert a == b


# ---------------------------------------------------------------------------
# Singleton tests
# ---------------------------------------------------------------------------


class TestGetCubeClient:
    """Tests for the singleton get_cube_client()."""

    def test_returns_same_instance(self):
        """Multiple calls should return the same instance."""
        # Force reset for test isolation
        import app.semantic.cube_client as mod
        from app.semantic.cube_client import get_cube_client

        mod._cube_client = None

        client1 = get_cube_client()
        client2 = get_cube_client()
        assert client1 is client2

    def test_uses_settings_defaults(self):
        """Without args, should use settings.CUBE_API_URL."""
        import app.semantic.cube_client as mod
        from app.semantic.cube_client import get_cube_client

        mod._cube_client = None

        client = get_cube_client()
        assert client.api_url.endswith("/cubejs-api/v1")
```

backend/tests/semantic/test_cube_client_phase9.py

163 lines · 5.1 KB

```python
"""Tests for Phase 9 CubeClient fixes: JWT auth, raw-meta caching,
agent-context metric ranking. All offline — no Cube, Redis, or DB.
"""

from datetime import UTC, datetime

from app.semantic.cube_client import (
    MAX_AGENT_CONTEXT_METRICS,
    CubeClient,
    MetricDefinition,
    MetricType,
)

# ---------------------------------------------------------------------------
# JWT auth
# ---------------------------------------------------------------------------


def test_auth_token_none_without_secret():
    client = CubeClient(api_url="http://cube", api_secret="")
    assert client._auth_token() is None


def test_auth_token_is_decodable_jwt():
    from jose import jwt

    secret = "test-secret"
    client = CubeClient(api_url="http://cube", api_secret=secret)

    token = client._auth_token()
    assert token is not None

    payload = jwt.decode(token, secret, algorithms=["HS256"])
    assert "exp" in payload
    assert payload["exp"] > datetime.now(UTC).timestamp()


def test_auth_token_cached_until_near_expiry():
    secret = "test-secret"
    client = CubeClient(api_url="http://cube", api_secret=secret)

    first = client._auth_token()
    second = client._auth_token()
    assert first == second, "token should be cached, not re-minted per request"


# ---------------------------------------------------------------------------
# Raw-meta cache re-parse (Redis hit path)
# ---------------------------------------------------------------------------


RAW_META = {
    "cubes": [
        {
            "name": "Sales",
            "title": "Sales",
            "measures": [{"name": "revenue_total", "title": "Total Revenue", "type": "sum"}],
            "dimensions": [
                {"name": "region", "type": "string"},
                {"name": "transaction_date", "type": "time"},
            ],
        }
    ]
}


async def test_meta_cache_hit_reparses_metrics(monkeypatch):
    """A raw /meta payload from cache must come back WITH parsed metrics."""
    client = CubeClient(api_url="http://cube", api_secret="s")

    async def fake_get_cached(key):
        return RAW_META

    async def fake_set_cached(key, value):
        pass

    monkeypatch.setattr(client, "_get_cached", fake_get_cached)
    monkeypatch.setattr(client, "_set_cached", fake_set_cached)

    meta = await client.get_meta()
    assert "Sales.revenue_total" in meta.metrics
    assert meta.metrics["Sales.revenue_total"].cube_name == "Sales"


# ---------------------------------------------------------------------------
# Agent-context metric ranking
# ---------------------------------------------------------------------------


def _metric(
    name: str, title: str = "", description: str = "", cube: str = "Sales"
) -> MetricDefinition:
    return MetricDefinition(
        name=name,
        title=title or name,
        description=description,
        metric_type=MetricType.SUM,
        cube_name=cube,
        measure_name=name,
    )
```

```python


def _metric_dict(*metrics: MetricDefinition) -> dict[str, MetricDefinition]:
    """Key by metric.name — mirrors how _parse_meta indexes metrics."""
    return {m.name: m for m in metrics}


def test_rank_metrics_scores_name_over_description():
    metrics = _metric_dict(
        _metric("Sales.revenue_total", description="revenue things"),
        _metric("Sales.units_sold", description="revenue mentioned in passing"),
    )
    ranked = CubeClient._rank_metrics_for_query(metrics, "total revenue")
    assert ranked[0].name == "Sales.revenue_total"


def test_rank_metrics_empty_when_no_overlap():
    metrics = _metric_dict(_metric("Sales.revenue_total"))
    assert CubeClient._rank_metrics_for_query(metrics, "completely unrelated") == []


def test_rank_metrics_respects_limit():
    metrics = _metric_dict(*(_metric(f"Sales.revenue_{i}") for i in range(50)))
    ranked = CubeClient._rank_metrics_for_query(metrics, "revenue", limit=20)
    assert len(ranked) == 20


async def test_agent_context_ranks_when_over_limit(monkeypatch):
    client = CubeClient(api_url="http://cube", api_secret="s")

    metrics = _metric_dict(
        _metric("Sales.revenue_total"),
        *(
            _metric(f"Noise.metric_{i}", description="unrelated")
            for i in range(MAX_AGENT_CONTEXT_METRICS + 4)
        ),
    )

    async def fake_list_metrics(force_refresh=False):
        return metrics

    monkeypatch.setattr(client, "list_metrics", fake_list_metrics)

    context = await client.get_agent_context(query="show me total revenue")
    assert "Sales.revenue_total" in context
    assert "Noise.metric_1" not in context
    assert len(metrics) > MAX_AGENT_CONTEXT_METRICS  # precondition


async def test_agent_context_full_list_when_no_match(monkeypatch):
    client = CubeClient(api_url="http://cube", api_secret="s")

    metrics = _metric_dict(
        *(_metric(f"Sales.metric_{i}") for i in range(MAX_AGENT_CONTEXT_METRICS + 3))
    )

    async def fake_list_metrics(force_refresh=False):
        return metrics

    monkeypatch.setattr(client, "list_metrics", fake_list_metrics)

    context = await client.get_agent_context(query="something entirely different")
    assert f"Sales.metric_{MAX_AGENT_CONTEXT_METRICS + 2}" in context
```

backend/tests/semantic/test_cube_query.py

135 lines · 4.4 KB

```python
"""Tests for Phase 10 CubeClient: tenant-scoped auth tokens and query()."""

from jose import jwt as jose_jwt

from app.semantic.cube_client import CubeClient

SECRET = "test-secret"
TENANT_A = "00000000-0000-0000-0000-00000000000a"
TENANT_B = "00000000-0000-0000-0000-00000000000b"


def _client() -> CubeClient:
    return CubeClient(api_url="http://fake-cube:4000/cubejs-api/v1", api_secret=SECRET)


# ---------------------------------------------------------------------------
# Tenant-scoped tokens
# ---------------------------------------------------------------------------


def test_anonymous_token_has_no_tenant_claim():
    token = _client()._auth_token()
    payload = jose_jwt.decode(token, SECRET, algorithms=["HS256"])
    assert "tenantId" not in payload


def test_tenant_token_carries_claim():
    token = _client()._auth_token(tenant_id=TENANT_A)
    payload = jose_jwt.decode(token, SECRET, algorithms=["HS256"])
    assert payload["tenantId"] == TENANT_A


def test_tokens_cached_per_tenant():
    client = _client()

    a1 = client._auth_token(tenant_id=TENANT_A)
    a2 = client._auth_token(tenant_id=TENANT_A)
    b = client._auth_token(tenant_id=TENANT_B)
    anon = client._auth_token()

    assert a1 == a2, "same tenant should reuse its cached token"
    assert a1 != b, "different tenants must get different tokens"
    assert a1 != anon


# ---------------------------------------------------------------------------
# query()
# ---------------------------------------------------------------------------


class FakeTransport:
    """Captures the last request and replays a canned /load response."""

    def __init__(self, response: dict):
        self.response = response
        self.last_json_body: dict | None = None
        self.last_headers: dict = {}

    async def __call__(self, path, method="GET", json_body=None, tenant_id=None):
        self.last_json_body = json_body
        self.tenant_id = tenant_id
        return self.response


LOAD_RESPONSE = {
    "query": {"measures": ["Sales.revenue_total"]},
    "data": [
        {"Sales.revenue_total": 120000, "Sales.region": "North"},
        {"Sales.revenue_total": 95000, "Sales.region": "South"},
    ],
    "annotation": {
        "measures": {"Sales.revenue_total": {"title": "Total Revenue", "type": "sum"}},
        "dimensions": {"Sales.region": {"title": "Region", "type": "string"}},
    },
}


async def test_query_flattens_and_totals(monkeypatch):
    client = _client()
    transport = FakeTransport(LOAD_RESPONSE)
    monkeypatch.setattr(client, "_request", transport)

    result = await client.query(
        metrics=["Sales.revenue_total"],
        dimensions=["Sales.region"],
        tenant_id=TENANT_A,
    )

    # Flattened rows strip the cube prefix
    assert result.data[0] == {"revenue_total": 120000, "region": "North"}
    # Single-metric total sums the qualified keys
    assert result.total == 215000
    assert result.metric_name == "Sales.revenue_total"
    assert result.annotation["measures"]["Sales.revenue_total"]["type"] == "sum"
    assert result.query_latency_ms >= 0


async def test_query_payload_shape(monkeypatch):
    client = _client()
    transport = FakeTransport({"data": [], "annotation": {}})
```

```
    monkeypatch.setattr(client, "_request", transport)

    await client.query(
        metrics=["Sales.revenue_total"],
        dimensions=["Sales.region"],
        time_dimensions=[{"dimension": "Sales.transaction_date", "granularity": "month"}],
        order=[["Sales.revenue_total", "desc"]],
        limit=50,
        tenant_id=TENANT_A,
    )

    assert transport.tenant_id == TENANT_A, "tenant must reach the transport for the JWT"
    body = transport.last_json_body["query"]
    assert body["measures"] == ["Sales.revenue_total"]
    assert body["dimensions"] == ["Sales.region"]
    assert body["timeDimensions"] == [
        {"dimension": "Sales.transaction_date", "granularity": "month"}
    ]
    assert body["order"] == {"Sales.revenue_total": "desc"}
    assert body["limit"] == 50


async def test_query_multi_metric_no_total(monkeypatch):
    client = _client()
    transport = FakeTransport(
        {
            "data": [{"Sales.revenue_total": 10, "Sales.units_sold": 2}],
            "annotation": {},
        }
    )
    monkeypatch.setattr(client, "_request", transport)

    result = await client.query(metrics=["Sales.revenue_total", "Sales.units_sold"])
    assert result.total is None
    assert result.metric_name == "multi"
```

backend/tests/services/test_audit.py

124 lines · 3.4 KB

```python
"""Tests for the Phase 12 audit writer (app.services.audit)."""

import uuid
from unittest.mock import AsyncMock, MagicMock

from app.services import audit as audit_module
from app.services.audit import write_audit_entry

SESSION = str(uuid.uuid4())
USER = str(uuid.uuid4())
TENANT = "00000000-0000-0000-0000-000000000001"


def _entry(**overrides) -> dict:
    entry = {
        "session_id": SESSION,
        "user_id": USER,
        "tenant_id": TENANT,
        "input_prompt_hash": "a" * 64,
        "generated_sql": "SELECT 1",
        "model_name": "claude-opus-4",
        "model_version": "latest",
        "input_tokens": 100,
        "output_tokens": 50,
        "latency_ms": 12.5,
    }
    entry.update(overrides)
    return entry


def _make_conn() -> MagicMock:
    conn = MagicMock()
    conn.execute = AsyncMock()
    tx = MagicMock()
    tx.__aenter__ = AsyncMock(return_value=None)
    tx.__aexit__ = AsyncMock(return_value=False)
    conn.transaction = MagicMock(return_value=tx)
    conn.close = AsyncMock()
    return conn


def _patch_asyncpg(monkeypatch, conn=None, error=None):
    async def fake_connect(dsn=None, **kwargs):
        if error:
            raise error
        return conn

    monkeypatch.setattr(audit_module.asyncpg, "connect", fake_connect)
    return conn


async def test_write_sets_guc_then_inserts(monkeypatch):
    conn = _make_conn()
    _patch_asyncpg(monkeypatch, conn=conn)

    await write_audit_entry(_entry())

    assert conn.execute.await_count == 2

    # First call: the tenant GUC — statement plus one bound parameter
    guc_call = conn.execute.call_args_list[0]
    assert guc_call.args[1] == TENANT

    # Second call: the module's exact parameterized insert, fully bound
    insert_call = conn.execute.call_args_list[1]
    assert insert_call.args[0] == audit_module._AUDIT_INSERT
    assert insert_call.args[1:] == (
        SESSION,
        USER,
        TENANT,
        "a" * 64,
        "SELECT 1",
        "claude-opus-4",
        "latest",
        100,
        50,
        12.5,
    )


async def test_skips_non_uuid_identities(monkeypatch):
    conn = _make_conn()
    _patch_asyncpg(monkeypatch, conn=conn)

    await write_audit_entry(_entry(user_id="unknown", tenant_id="default"))

    assert conn.execute.await_count == 0, "no DB work for non-UUID identity fallbacks"


async def test_fails_open_on_db_error(monkeypatch):
    _patch_asyncpg(monkeypatch, error=ConnectionError("db down"))

    # Must not raise — an audit outage never breaks the pipeline
    await write_audit_entry(_entry())


async def test_hash_is_real_prompt_hash():
    """The llm_client callback now hashes the prompt text, not model+tokens."""
    import hashlib

```

```
    from app.core.llm_client import LLMCallResult, LLMClient

    captured: list[dict] = []

    async def callback(entry):
        captured.append(entry)

    client = LLMClient()
    client.set_audit_callback(callback)
    result = LLMCallResult(
        content="{}", model_name="claude-opus-4", input_tokens=10, output_tokens=5, latency_ms=1.0
    )

    await client._audit(
        result=result,
        messages="Show revenue by region",
        user_id=USER,
        tenant_id=TENANT,
        session_id=SESSION,
    )

    assert captured[0]["input_prompt_hash"] == hashlib.sha256(b"Show revenue by region").hexdigest()
    assert captured[0]["user_id"] == USER
    assert captured[0]["generated_sql"] == ""
```

backend/tests/services/test_chat_pipeline.py

259 lines · 9.0 KB

```python
"""Chat pipeline orchestration tests (Phase 13).

Exercises ChatService.process_query / process_query_stream end-to-end with:
- the REAL deterministic ValidationAgent (only its connector is faked)
- mocked LLM agents (router/nl2sql/chart_gen/narrative) via get_agent
- a mocked cache returning canned query results (skips the real connector
  in _step_execute)

Covers the happy path, early exits, and the >1M-row confirmation gate in
both sync and streaming modes.
"""

import json
from unittest.mock import AsyncMock, MagicMock

from app.agents.base import AgentResult
from app.agents.validation.validation_agent import ValidationAgent
from app.services.chat_service import ChatService

TENANT = "00000000-0000-0000-0000-000000000001"
ROWS = [{"region": "North", "total": 120000}, {"region": "South", "total": 95000}]
CHART_SPEC = {"chartType": "Bar Chart", "encodings": {}, "baseSize": {}, "data": {}}


def _llm_agent(output: dict, warnings: list[str] | None = None, success: bool = True):
    agent = MagicMock()
    agent.execute = AsyncMock(
        return_value=AgentResult(
            agent_name="mock", success=success, output=output, warnings=warnings or []
        )
    )
    return MagicMock(return_value=agent)  # agent_cls


def _patch_pipeline(
    monkeypatch,
    *,
    sql="SELECT region, SUM(revenue) FROM public.sales GROUP BY region",
    validation_sql=None,
    router_output=None,
    chart_output=None,
    narrative_text="Revenue is highest in the North.",
):
    agents = {
        "router": _llm_agent(router_output or {"intent": "chat_data", "dispatch_plan": []}),
        "nl2sql": _llm_agent({"sql": sql, "explanation": "groups revenue"}),
        # "validation" intentionally absent — the REAL ValidationAgent runs
        "chart_gen": _llm_agent(chart_output or {"chart_spec": CHART_SPEC}),
        "narrative": _llm_agent({"narrative": narrative_text}),
    }
    monkeypatch.setattr(
        "app.services.chat_service.get_agent",
        # LLM agents mocked; the REAL deterministic ValidationAgent runs
        lambda name: agents.get(name) or (ValidationAgent if name == "validation" else None),
    )

    # Cache: metric/fewshot/schema context hits (empty), query result hit
    cache = MagicMock()
    cache.get_metric_definitions = AsyncMock(return_value="")
    cache.get_schema_context = AsyncMock(return_value=[])
    cache.set_schema_context = AsyncMock()
    cache.get_few_shot_examples = AsyncMock(return_value=[])
    cache.set_few_shot_examples = AsyncMock()
    cache.get_query_result = AsyncMock(return_value=ROWS)
    cache.set_query_result = AsyncMock()
    monkeypatch.setattr("app.services.chat_service.get_cache", lambda: cache)

    # Conversations (Phase 14): mock persistence so no real DB connect is
    # attempted (keeps these tests fast and offline).
    import app.services.conversations as conversations_module

    monkeypatch.setattr(conversations_module, "create_conversation", AsyncMock(return_value=None))
    monkeypatch.setattr(
        conversations_module,
        "list_messages",
        AsyncMock(return_value=[]),
    )
    monkeypatch.setattr(conversations_module, "append_message", AsyncMock())

    # ValidationAgent's short-lived EXPLAIN connector
    explain_conn = MagicMock()
    explain_conn.explain = AsyncMock(
        return_value={
            "plan_type": "EXPLAIN (FORMAT JSON)",
            "plan_text": "[plan]",
            "estimated_cost": 10.0,
            "estimated_rows": 0,
        }
    )
    explain_conn.__aenter__ = AsyncMock(return_value=explain_conn)
    explain_conn.__aexit__ = AsyncMock(return_value=False)
    monkeypatch.setattr(
        "app.connectors.postgresql_connector.PostgreSQLConnector",
        MagicMock(return_value=explain_conn),
    )

    # Chart hallucination check: pass-through no-op
    validator = MagicMock()
    validator.warnings = []
    validator.fix_summary = []
```

```
    validator.corrected_spec = None
    validator.is_valid = True
    monkeypatch.setattr(
        "app.agents.validation.chart_validator.validate_chart_spec",
        lambda spec, data, auto_correct: validator,
    )

    return {"agents": agents, "cache": cache, "explain_conn": explain_conn}


async def _run_sync(service_args=None, **kwargs):
    service = ChatService(tenant_id=TENANT)
    return await service.process_query(
        query="show revenue by region",
        user_id="user-1",
        roles=["user"],
        **(service_args or {}),
        **kwargs,
    )


async def _collect_stream(**kwargs):
    service = ChatService(tenant_id=TENANT)
    events = []
    async for raw in service.process_query_stream(
        query="show revenue by region", user_id="user-1", roles=["user"], **kwargs
    ):
        events.append(json.loads(raw.removeprefix("data: ").strip()))
    return events


# ---------------------------------------------------------------------------
# Sync pipeline
# ---------------------------------------------------------------------------


async def test_sync_happy_path(monkeypatch):
    _patch_pipeline(monkeypatch)

    response = await _run_sync()

    assert response["sql"].startswith("SELECT")
    assert response["chart_spec"] == CHART_SPEC
    assert response["narrative"].startswith("Revenue is highest")
    assert response["requires_confirmation"] is False
    assert response["row_estimate"] == 0
    assert isinstance(response["warnings"], list)


async def test_sync_no_sql_early_exit(monkeypatch):
    _patch_pipeline(monkeypatch, sql=None)
    # nl2sql mock with success but empty sql → output lacks sql key
    response = await _run_sync()

    assert response["sql"] is None
    assert any("Could not generate SQL" in w for w in response["warnings"])


async def test_sync_validation_failed(monkeypatch):
    patch = _patch_pipeline(monkeypatch, sql="DELETE FROM public.sales")

    response = await _run_sync()

    assert response["sql"] is not None
    assert any("failed safety validation" in w for w in response["warnings"])
    # The destructive query was never EXPLAINed
    assert patch["explain_conn"].explain.await_count == 0


async def test_sync_confirmation_gate_blocks_then_proceeds(monkeypatch):
    patch = _patch_pipeline(monkeypatch)
    patch["explain_conn"].explain = AsyncMock(
        return_value={
            "plan_type": "EXPLAIN",
            "plan_text": "[big-plan]",
            "estimated_cost": 1e9,
            "estimated_rows": 2_000_000,
        }
    )

    # Without confirmation: stops before execution, flags set
    blocked = await _run_sync()
    assert blocked["requires_confirmation"] is True
    assert blocked["row_estimate"] == 2_000_000
    assert blocked["chart_spec"] is None, "no chart before confirmation"
    assert patch["cache"].get_query_result.await_count == 0, "execute never ran"

    # With confirmation: full pipeline
    confirmed = await _run_sync(confirm_large_query=True)
    assert confirmed["requires_confirmation"] is False
    assert confirmed["chart_spec"] == CHART_SPEC
    assert patch["cache"].get_query_result.await_count == 1


# ---------------------------------------------------------------------------
# Streaming pipeline
# ---------------------------------------------------------------------------


async def test_stream_event_order_happy_path(monkeypatch):
    _patch_pipeline(monkeypatch)

    events = await _collect_stream()

    order = [e["event"] for e in events]
```

```
    assert order == ["start", "intent", "sql", "validation", "data", "chart", "narrative", "done"]
    assert events[-1]["status"] == "complete"

    validation = events[3]
    assert validation["valid"] is True
    assert validation["requires_confirmation"] is False
    assert "row_estimate" in validation


async def test_stream_confirmation_required(monkeypatch):
    patch = _patch_pipeline(monkeypatch)
    patch["explain_conn"].explain = AsyncMock(
        return_value={
            "plan_type": "EXPLAIN",
            "plan_text": "[big]",
            "estimated_cost": 1e9,
            "estimated_rows": 5_000_000,
        }
    )

    events = await _collect_stream()

    assert [e["event"] for e in events] == ["start", "intent", "sql", "validation", "done"]
    done = events[-1]
    assert done["status"] == "confirmation_required"
    assert done["requires_confirmation"] is True
    assert done["row_estimate"] == 5_000_000
    assert patch["cache"].get_query_result.await_count == 0


async def test_stream_validation_failed_status(monkeypatch):
    _patch_pipeline(monkeypatch, sql="TRUNCATE TABLE public.sales")

    events = await _collect_stream()

    assert events[-1]["status"] == "validation_failed"
    assert events[3]["valid"] is False


async def test_stream_confirmed_resend_completes(monkeypatch):
    patch = _patch_pipeline(monkeypatch)
    patch["explain_conn"].explain = AsyncMock(
        return_value={
            "plan_type": "EXPLAIN",
            "plan_text": "[big]",
            "estimated_cost": 1e9,
            "estimated_rows": 5_000_000,
        }
    )

    events = await _collect_stream(confirm_large_query=True)

    assert events[-1]["status"] == "complete"
    assert patch["cache"].get_query_result.await_count == 1
```

backend/tests/services/test_chat_service_context.py

120 lines · 4.4 KB

```python
"""Tests for the Phase 11 ChatService wiring: schema context and few-shot
examples flow into the NL2SQL agent, with graceful degradation on failure.

All collaborators are mocked (cache, Cube client, retrieval, agent) — these
tests pin the SERVICE wiring, not the retrieval internals (covered in
tests/services/test_schema_retrieval.py).
"""

from unittest.mock import AsyncMock, MagicMock

from app.agents.base import AgentResult
from app.services.chat_service import ChatService

TENANT = "00000000-0000-0000-0000-000000000001"


def _make_agent():
    agent = MagicMock()
    agent.execute = AsyncMock(
        return_value=AgentResult(agent_name="nl2sql", success=True, output={"sql": "SELECT 1"})
    )
    return agent


def _make_cache():
    cache = MagicMock()
    cache.get_metric_definitions = AsyncMock(return_value="metric context")
    cache.get_schema_context = AsyncMock(return_value=None)
    cache.set_schema_context = AsyncMock()
    cache.get_few_shot_examples = AsyncMock(return_value=None)
    cache.set_few_shot_examples = AsyncMock()
    return cache


def _patch_all(monkeypatch, agent, cache, schema_ctx=None, few_shot=None):
    agent_cls = MagicMock(return_value=agent)
    monkeypatch.setattr("app.services.chat_service.get_agent", lambda name: agent_cls)
    monkeypatch.setattr("app.services.chat_service.get_cache", lambda: cache)

    monkeypatch.setattr(
        "app.services.schema_retrieval.retrieve_schema_context",
        AsyncMock(return_value=schema_ctx or []),
    )
    monkeypatch.setattr(
        "app.services.schema_retrieval.retrieve_few_shot_examples",
        AsyncMock(return_value=few_shot or []),
    )


async def test_schema_and_few_shot_flow_into_agent(monkeypatch):
    agent = _make_agent()
    cache = _make_cache()
    schema_ctx = [{"table_name": "public.sales", "description": "", "columns": []}]
    few_shot = [{"nl_query": "q", "expected_sql": "s"}]
    _patch_all(monkeypatch, agent, cache, schema_ctx=schema_ctx, few_shot=few_shot)

    service = ChatService(tenant_id=TENANT)
    sql, warnings = await service._step_nl2sql("revenue by region", "user-1")

    assert sql == "SELECT 1"
    assert warnings == []

    kwargs = agent.execute.call_args.kwargs
    assert kwargs["schema_context"] == schema_ctx
    assert kwargs["few_shot_examples"] == few_shot
    assert kwargs["metric_definitions"] == "metric context"

    # Retrieval results were cached under the tenant
    cache.set_schema_context.assert_awaited_once()
    cache.set_few_shot_examples.assert_awaited_once()


async def test_cached_context_skips_retrieval(monkeypatch):
    agent = _make_agent()
    cache = _make_cache()
    cached_schema = [{"table_name": "public.orders", "description": "", "columns": []}]
    cached_few_shot = [{"nl_query": "cq", "expected_sql": "cs"}]
    cache.get_schema_context = AsyncMock(return_value=cached_schema)
    cache.get_few_shot_examples = AsyncMock(return_value=cached_few_shot)

    retrieve_schema = AsyncMock(return_value=[])
    retrieve_few_shot = AsyncMock(return_value=[])
    _patch_all(monkeypatch, agent, cache)
    import app.services.schema_retrieval as sr

    monkeypatch.setattr(sr, "retrieve_schema_context", retrieve_schema)
    monkeypatch.setattr(sr, "retrieve_few_shot_examples", retrieve_few_shot)

    service = ChatService(tenant_id=TENANT)
    await service._step_nl2sql("orders", "user-1")

    retrieve_schema.assert_not_awaited()
    retrieve_few_shot.assert_not_awaited()
    assert agent.execute.call_args.kwargs["schema_context"] == cached_schema
    assert agent.execute.call_args.kwargs["few_shot_examples"] == cached_few_shot


async def test_retrieval_failure_degrades_gracefully(monkeypatch):
    agent = _make_agent()
    cache = _make_cache()
```

```
    _patch_all(monkeypatch, agent, cache)

    import app.services.schema_retrieval as sr

    monkeypatch.setattr(
        sr, "retrieve_schema_context", AsyncMock(side_effect=RuntimeError("db down"))
    )
    monkeypatch.setattr(
        sr, "retrieve_few_shot_examples", AsyncMock(side_effect=RuntimeError("db down"))
    )

    service = ChatService(tenant_id=TENANT)
    sql, warnings = await service._step_nl2sql("revenue", "user-1")

    # Pipeline continues: SQL generated, context empty, no warnings leaked
    assert sql == "SELECT 1"
    assert warnings == []
    kwargs = agent.execute.call_args.kwargs
    assert kwargs["schema_context"] == []
    assert kwargs["few_shot_examples"] == []
```

backend/tests/services/test_conversation_wiring.py

207 lines · 7.5 KB

```python
"""Phase 14 wiring tests: NL2SQL prompt history section + ChatService turns.

The ChatService tests patch the conversations service module (the same
seam the endpoint uses) and reuse the pipeline-test harness shape from
test_chat_pipeline.py for the rest.
"""

import json
from unittest.mock import AsyncMock, MagicMock

from app.agents.base import AgentConfig
from app.agents.nl2sql.nl2sql_agent import NL2SQLAgent
from app.services.chat_service import ChatService

TENANT = "00000000-0000-0000-0000-000000000001"
USER = "00000000-0000-0000-0000-0000000000aa"
CONV = "00000000-0000-0000-0000-0000000000bb"


# ---------------------------------------------------------------------------
# NL2SQLAgent prompt section
# ---------------------------------------------------------------------------


def test_history_section_rendered():
    agent = NL2SQLAgent(AgentConfig(model_name="test"))
    message = agent._build_user_message(
        query="now break that down by region",
        schema_context=None,
        few_shot_examples=None,
        metric_definitions=None,
        tenant_id=TENANT,
        history=[
            {"role": "user", "content": "total revenue", "generated_sql": None},
            {
                "role": "assistant",
                "content": "Total revenue is $1.2M",
                "generated_sql": "SELECT SUM(revenue) FROM public.sales",
            },
        ],
    )

    assert "## Conversation History" in message
    assert "User: total revenue" in message
    assert "Assistant: Total revenue is $1.2M" in message
    assert "[SQL: SELECT SUM(revenue) FROM public.sales]" in message
    # The current question stays prominent
    assert "## User Question" in message


def test_no_history_section_when_empty():
    agent = NL2SQLAgent(AgentConfig(model_name="test"))
    message = agent._build_user_message(
        query="total revenue",
        schema_context=None,
        few_shot_examples=None,
        metric_definitions=None,
        tenant_id=TENANT,
        history=[],
    )
    assert "## Conversation History" not in message


# ---------------------------------------------------------------------------
# ChatService conversation wiring
# ---------------------------------------------------------------------------


def _patch_conversations(monkeypatch, history=None, create_returns=CONV):
    import app.services.conversations as svc

    monkeypatch.setattr(svc, "create_conversation", AsyncMock(return_value=create_returns))
    monkeypatch.setattr(svc, "list_messages", AsyncMock(return_value=history or []))
    monkeypatch.setattr(svc, "append_message", AsyncMock())
    return svc


def _patch_agents(monkeypatch):
    """Only the LLM agents; the real ValidationAgent runs (connector faked)."""
    from app.agents.base import AgentResult
    from app.agents.validation.validation_agent import ValidationAgent

    def llm_cls(output):
        agent = MagicMock()
        agent.execute = AsyncMock(
            return_value=AgentResult(agent_name="mock", success=True, output=output)
        )
        return MagicMock(return_value=agent)

    agents = {
        "router": llm_cls({"intent": "chat_data", "dispatch_plan": []}),
        "nl2sql": llm_cls({"sql": "SELECT region FROM public.sales"}),
        "chart_gen": llm_cls({"chart_spec": {}}),
        "narrative": llm_cls({"narrative": "North leads."}),
    }

    monkeypatch.setattr(
        "app.services.chat_service.get_agent",
        lambda name: agents.get(name) or (ValidationAgent if name == "validation" else None),
    )
```

```python

    cache = MagicMock()
    cache.get_metric_definitions = AsyncMock(return_value="")
    cache.get_schema_context = AsyncMock(return_value=[])
    cache.set_schema_context = AsyncMock()
    cache.get_few_shot_examples = AsyncMock(return_value=[])
    cache.set_few_shot_examples = AsyncMock()
    cache.get_query_result = AsyncMock(return_value=[{"region": "North"}])
    cache.set_query_result = AsyncMock()
    monkeypatch.setattr("app.services.chat_service.get_cache", lambda: cache)

    explain_conn = MagicMock()
    explain_conn.explain = AsyncMock(
        return_value={
            "plan_type": "EXPLAIN",
            "plan_text": "[p]",
            "estimated_cost": 1.0,
            "estimated_rows": 0,
        }
    )
    explain_conn.__aenter__ = AsyncMock(return_value=explain_conn)
    explain_conn.__aexit__ = AsyncMock(return_value=False)
    monkeypatch.setattr(
        "app.connectors.postgresql_connector.PostgreSQLConnector",
        MagicMock(return_value=explain_conn),
    )

    # Chart validator no-op
    validator = MagicMock()
    validator.warnings = []
    validator.fix_summary = []
    validator.corrected_spec = None
    monkeypatch.setattr(
        "app.agents.validation.chart_validator.validate_chart_spec",
        lambda spec, data, auto_correct: validator,
    )
    return agents


async def test_history_flows_into_agent_and_turns_persisted(monkeypatch):
    conv = _patch_conversations(
        monkeypatch,
        history=[
            {"role": "user", "content": "total revenue"},
            {"role": "assistant", "content": "It is $1.2M"},
        ],
    )
    agents = _patch_agents(monkeypatch)

    service = ChatService(tenant_id=TENANT)
    response = await service.process_query(
        query="now by region", user_id=USER, roles=["user"], conversation_id=CONV
    )

    # Given conversation_id → no create call; history loaded; turns persisted
    conv.create_conversation.assert_not_awaited()
    conv.list_messages.assert_awaited_once()
    agent_execute = agents["nl2sql"].return_value.execute
    assert agent_execute.call_args.kwargs["history"][0]["content"] == "total revenue"

    roles = [c.kwargs["role"] for c in conv.append_message.await_args_list]
    assert roles == ["user", "assistant"]
    assert conv.append_message.await_args_list[0].kwargs["content"] == "now by region"
    assert conv.append_message.await_args_list[1].kwargs["generated_sql"].startswith("SELECT")
    assert response["conversation_id"] == CONV


async def test_new_conversation_created_when_id_absent(monkeypatch):
    conv = _patch_conversations(monkeypatch, create_returns=CONV)
    _patch_agents(monkeypatch)

    service = ChatService(tenant_id=TENANT)
    response = await service.process_query(query="total revenue", user_id=USER, roles=["user"])

    conv.create_conversation.assert_awaited_once()
    assert conv.create_conversation.await_args.kwargs["title"] == "total revenue"
    assert response["conversation_id"] == CONV


async def test_conversation_create_failure_falls_back(monkeypatch):
    _patch_conversations(monkeypatch, create_returns=None)  # fail-open
    _patch_agents(monkeypatch)

    service = ChatService(tenant_id=TENANT)
    response = await service.process_query(query="total revenue", user_id=USER, roles=["user"])

    # Pipeline still completes with a generated (unpersisted) conversation id
    assert response["conversation_id"]
    assert response["sql"].startswith("SELECT")


async def test_history_load_failure_cold_start(monkeypatch):
    conv = _patch_conversations(monkeypatch)
    conv.list_messages = AsyncMock(side_effect=ConnectionError("db down"))
    agents = _patch_agents(monkeypatch)

    service = ChatService(tenant_id=TENANT)
    events = []
    async for raw in service.process_query_stream(
        query="total revenue", user_id=USER, roles=["user"]
    ):
        events.append(json.loads(raw.removeprefix("data: ").strip()))

    assert events[-1]["status"] == "complete"
    assert events[0]["conversation_id"] == CONV
```

```
    agent_execute = agents["nl2sql"].return_value.execute
    assert agent_execute.call_args.kwargs["history"] == []
```

backend/tests/services/test_conversations.py

122 lines · 4.0 KB

```python
"""Tests for the conversations service (Phase 14) — mocked asyncpg.

Pins the persistence contract: tenant GUC before every statement,
parameterized reads/writes, per-user scoping, fail-open writes,
raise-on-read (callers decide degradation).
"""

from datetime import UTC, datetime
from unittest.mock import AsyncMock, MagicMock

from app.services import conversations as svc

TENANT = "00000000-0000-0000-0000-000000000001"
USER = "00000000-0000-0000-0000-0000000000aa"
CONV = "00000000-0000-0000-0000-0000000000bb"
NOW = datetime(2026, 8, 15, tzinfo=UTC)


def _patch_connect(monkeypatch, fetch_rows=None, error=None):
    conn = MagicMock()
    conn.execute = AsyncMock()
    conn.close = AsyncMock()
    conn.fetch = AsyncMock(
        return_value=fetch_rows or [],
        side_effect=error if error else None,
    )
    tx = MagicMock()
    tx.__aenter__ = AsyncMock(return_value=None)
    tx.__aexit__ = AsyncMock(return_value=False)
    conn.transaction = MagicMock(return_value=tx)

    async def fake_connect(dsn=None, **kwargs):
        return conn

    monkeypatch.setattr(svc.asyncpg, "connect", fake_connect)
    return conn


async def test_create_conversation_sets_guc_then_inserts(monkeypatch):
    conn = _patch_connect(monkeypatch)

    created = await svc.create_conversation(USER, TENANT, title="revenue question")

    assert created is not None
    guc_call = conn.execute.call_args_list[0]
    assert guc_call.args[1] == TENANT
    insert_call = conn.execute.call_args_list[1]
    assert insert_call.args[0].startswith("INSERT INTO conversations")
    assert insert_call.args[1:] == (created, USER, TENANT, "revenue question")


async def test_create_conversation_fail_open(monkeypatch):
    conn = _patch_connect(monkeypatch)
    conn.execute = AsyncMock(side_effect=ConnectionError("db down"))

    assert await svc.create_conversation(USER, TENANT, title="t") is None


async def test_append_message_inserts_and_bumps_updated_at(monkeypatch):
    conn = _patch_connect(monkeypatch)

    await svc.append_message(CONV, TENANT, "user", "hello", generated_sql="SELECT 1")

    # GUC, INSERT, UPDATE — in order, inside the transaction
    statements = [c.args[0] for c in conn.execute.call_args_list]
    assert len(statements) == 3
    assert statements[1].startswith("INSERT INTO messages")
    assert statements[2].startswith("UPDATE conversations")
    assert conn.execute.call_args_list[1].args[1:] == (CONV, TENANT, "user", "hello", "SELECT 1")


async def test_append_message_fail_open(monkeypatch):
    conn = _patch_connect(monkeypatch)
    conn.execute = AsyncMock(side_effect=RuntimeError("gone"))

    # Must not raise
    await svc.append_message(CONV, TENANT, "assistant", "text")


async def test_list_conversations_scopes_to_user(monkeypatch):
    conn = _patch_connect(
        monkeypatch,
        fetch_rows=[{"id": CONV, "title": "Q3 revenue", "created_at": NOW, "updated_at": NOW}],
    )

    rows = await svc.list_conversations(USER, TENANT, limit=10)

    fetch_call = conn.fetch.call_args
    assert fetch_call.args[1] == USER  # user scoping parameter
    assert fetch_call.args[2] == 10
    assert rows[0]["id"] == CONV
    assert rows[0]["title"] == "Q3 revenue"


async def test_list_messages_reverses_to_chronological(monkeypatch):
    def row(role, content, i):
        return {
            "role": role,
            "content": content,
            "generated_sql": None,
```

```
            "created_at": NOW.replace(minute=i),
        }

    # DB returns newest-first; the service must return chronological
    conn = _patch_connect(
        monkeypatch,
        fetch_rows=[row("assistant", "answer", 1), row("user", "question", 0)],
    )

    rows = await svc.list_messages(CONV, TENANT)

    assert [r["role"] for r in rows] == ["user", "assistant"]
    assert conn.fetch.call_args.args[1] == CONV


async def test_list_messages_raises_on_db_error(monkeypatch):
    _patch_connect(monkeypatch, error=ConnectionError("db down"))

    import pytest

    with pytest.raises(ConnectionError):
        await svc.list_messages(CONV, TENANT)
```

backend/tests/services/test_login_rate_limit.py

150 lines · 4.8 KB

```python
"""Tests for login rate limiting: CacheService counter + /auth/login 429 path."""

from unittest.mock import AsyncMock, MagicMock

from httpx import ASGITransport, AsyncClient

from app.core.cache import CacheService

TEST_EMAIL = "rate-limit-test@genbi.local"
# Test-only credential derived from the fixture email (no literals).
TEST_PASSWORD = TEST_EMAIL.split("@")[0] + "!1"


# ---------------------------------------------------------------------------
# CacheService counter methods (fake Redis injected into the L2)
# ---------------------------------------------------------------------------


class FakeRedis:
    def __init__(self):
        self.counters: dict[str, int] = {}
        self.ttls: dict[str, int] = {}

    async def incr(self, key):
        self.counters[key] = self.counters.get(key, 0) + 1
        return self.counters[key]

    async def expire(self, key, seconds):
        self.ttls[key] = seconds

    async def get(self, key):
        return self.counters.get(key)

    async def delete(self, key):
        self.counters.pop(key, None)


def _cache_with_fake_redis() -> tuple[CacheService, FakeRedis]:
    cache = CacheService()
    redis = FakeRedis()
    cache._l2._redis = redis
    cache._l2._init_attempted = True  # skip the real connect path in _ensure
    cache._l2._available = True
    return cache, redis


async def test_register_increments_and_sets_ttl():
    cache, redis = _cache_with_fake_redis()

    assert await cache.register_failed_login("User@Example.com ") == 1
    assert await cache.register_failed_login("user@example.com") == 2

    # Key is normalized; TTL set once on first increment
    key = "genbi:loginfail:user@example.com"
    assert redis.counters[key] == 2
    assert redis.ttls[key] == 900


async def test_login_failures_reads_zero_for_unknown():
    cache, _ = _cache_with_fake_redis()
    assert await cache.login_failures("nobody@example.com") == 0


async def test_clear_resets_counter():
    cache, _ = _cache_with_fake_redis()
    await cache.register_failed_login("user@example.com")
    await cache.register_failed_login("user@example.com")

    await cache.clear_failed_logins("user@example.com")

    assert await cache.login_failures("user@example.com") == 0


async def test_counter_fails_open_without_redis():
    cache = CacheService()
    cache._l2._init_attempted = True  # memoized as unavailable — no real connect
    cache._l2._redis = None
    cache._l2._available = False

    assert await cache.register_failed_login("user@example.com") == 0
    assert await cache.login_failures("user@example.com") == 0
    await cache.clear_failed_logins("user@example.com")  # no raise


# ---------------------------------------------------------------------------
# /auth/login endpoint behavior
# ---------------------------------------------------------------------------


def _patch_auth_cache(monkeypatch, failures=0):
    cache = MagicMock()
    cache.login_failures = AsyncMock(return_value=failures)
    cache.register_failed_login = AsyncMock(return_value=failures + 1)
    cache.clear_failed_logins = AsyncMock()
    monkeypatch.setattr("app.api.v1.auth.get_cache", lambda: cache)
    return cache


async def _post_login(json_body):

```

```
    from app.db.session import get_auth_db
    from app.main import create_app

    app = create_app()

    # Override the auth DB dependency: the failure path under test returns
    # zero rows without needing a live database.
    session = MagicMock()
    query_result = MagicMock()
    query_result.all = MagicMock(return_value=[])
    session.execute = AsyncMock(return_value=query_result)

    async def fake_db():
        yield session

    app.dependency_overrides[get_auth_db] = fake_db

    transport = ASGITransport(app=app)
    async with AsyncClient(transport=transport, base_url="http://test") as client:
        return await client.post("/api/v1/auth/login", json=json_body)


async def test_login_returns_429_at_threshold(monkeypatch):
    _patch_auth_cache(monkeypatch, failures=5)

    res = await _post_login({"email": TEST_EMAIL, "password": TEST_PASSWORD})

    assert res.status_code == 429
    assert res.json()["detail"]["code"] == "TOO_MANY_ATTEMPTS"


async def test_login_failure_registers_attempt(monkeypatch):
    cache = _patch_auth_cache(monkeypatch, failures=0)

    res = await _post_login({"email": TEST_EMAIL, "password": TEST_PASSWORD + "-wrong"})

    assert res.status_code == 401
    cache.register_failed_login.assert_awaited_once_with(TEST_EMAIL)
    cache.clear_failed_logins.assert_not_awaited()


async def test_login_invalid_tenant_uuid_also_registers(monkeypatch):
    cache = _patch_auth_cache(monkeypatch, failures=0)

    res = await _post_login(
        {"email": TEST_EMAIL, "password": TEST_PASSWORD, "tenant_id": "not-a-uuid"}
    )

    assert res.status_code == 401
    cache.register_failed_login.assert_awaited_once_with(TEST_EMAIL)
```

backend/tests/services/test_reports.py

340 lines · 10.9 KB

```python
"""Phase 16 report tests: service pipeline + persistence contract.

All collaborators mocked: Cube client, LLM client, chart bridge, asyncpg.
"""

import json
from unittest.mock import AsyncMock, MagicMock

from app.services import reports as reports_service
from app.services.reports import (
    ReportGenerationError,
    _build_chart_spec,
    _plan_report,
    generate_report,
)

TENANT = "00000000-0000-0000-0000-000000000001"
USER = "00000000-0000-0000-0000-0000000000aa"


def _patch_cube(monkeypatch, query_result=None, query_error=None):
    fake = MagicMock()
    fake.get_agent_context = AsyncMock(return_value="## Available Metrics\n- Sales.revenue_total")
    fake.query = AsyncMock(
        return_value=query_result,
        side_effect=query_error if query_error else None,
    )
    monkeypatch.setattr("app.semantic.cube_client.get_cube_client", lambda: fake)
    return fake


def _patch_llm(monkeypatch, plan=None, summary="Overall summary."):
    """Two sequential invokes: planner (parsed JSON) then summarizer (content)."""
    calls = []

    class FakeResult:
        def __init__(self, parsed, content):
            self.parsed = parsed
            self.content = content

    async def fake_invoke(**kwargs):
        calls.append(kwargs)
        if plan is not None and len(calls) == 1:
            return FakeResult(plan, json.dumps(plan))
        return FakeResult(None, summary)

    client = MagicMock()
    client.invoke = fake_invoke
    monkeypatch.setattr("app.core.llm_client.get_llm_client", lambda: client)
    return calls


def _patch_bridge(monkeypatch, svg="<svg>chart</svg>"):
    bridge = MagicMock()
    bridge.render = AsyncMock(return_value={"success": True, "svg": svg})
    monkeypatch.setattr(
        "app.agents.chart.flint_bridge.FlintChartBridge", MagicMock(return_value=bridge)
    )
    return bridge


def _patch_save(monkeypatch, returns=True):
    mock = AsyncMock(return_value=returns)
    monkeypatch.setattr(reports_service, "save_report", mock)
    return mock


def _query_result(rows):
    result = MagicMock()
    result.data = rows
    result.total = None
    return result


# ---------------------------------------------------------------------------
# Planner
# ---------------------------------------------------------------------------


async def test_plan_report_parses_and_sanitizes(monkeypatch):
    _patch_cube(monkeypatch)
    calls = _patch_llm(
        monkeypatch,
        plan={
            "title": "Revenue report",
            "sections": [
                {
                    "metric": "Sales.revenue_total",
                    "title": "Revenue by Region",
                    "dimension": "Sales.region",
                },
                {"metric": "", "title": "bad — dropped"},  # no metric → dropped
                {"metric": "Orders.order_count", "title": "Orders"},
            ],
        },
    )

    plan = await _plan_report("revenue report", TENANT, USER, max_sections=3)

    assert plan["title"] == "Revenue report"
```

```
    assert len(plan["sections"]) == 2
    assert plan["sections"][0]["dimension"] == "Sales.region"
    # Planner call: temperature 0, JSON response format (deterministic plan)
    assert calls[0]["options"].response_format == "json"


async def test_plan_report_rejects_empty_plan(monkeypatch):
    _patch_cube(monkeypatch)
    _patch_llm(monkeypatch, plan={"title": "x", "sections": []})

    import pytest

    with pytest.raises(ReportGenerationError):
        await _plan_report("anything", TENANT, USER, max_sections=3)


# ---------------------------------------------------------------------------
# Chart spec (deterministic)
# ---------------------------------------------------------------------------


def test_bar_spec_for_categorical_slice():
    spec = _build_chart_spec(
        {"metric": "Sales.revenue_total"},
        [{"region": "North", "revenue_total": 100}, {"region": "South", "revenue_total": 50}],
    )
    assert spec["chartType"] == "Bar Chart"
    assert spec["encodings"]["x"]["field"] == "region"
    assert spec["encodings"]["y"]["field"] == "revenue_total"


def test_line_spec_for_time_granularity():
    spec = _build_chart_spec(
        {"metric": "Sales.revenue_total", "granularity": "month"},
        [{"month": "2026-01", "revenue_total": 100}],
    )
    assert spec["chartType"] == "Line Chart"
    assert spec["encodings"]["x"]["field"] == "month"


# ---------------------------------------------------------------------------
# Full pipeline
# ---------------------------------------------------------------------------


async def test_generate_report_happy_path(monkeypatch):
    fake_cube = _patch_cube(
        monkeypatch, query_result=_query_result([{"region": "North", "revenue_total": 100}])
    )
    _patch_llm(
        monkeypatch,
        plan={
            "title": "Q3 Report",
            "sections": [
                {"metric": "Sales.revenue_total", "title": "Revenue", "dimension": "Sales.region"}
            ],
        },
    )
    bridge = _patch_bridge(monkeypatch)
    save = _patch_save(monkeypatch, returns=True)

    report = await generate_report("Q3 revenue", TENANT, USER)

    # Tenant threading into the Cube query (RLS-correct data per ADR 008)
    assert fake_cube.query.call_args.kwargs["tenant_id"] == TENANT
    assert bridge.render.call_args.args[0]["chartType"] == "Bar Chart"

    assert report["title"] == "Q3 Report"
    assert report["summary"] == "Overall summary."
    assert report["sections"][0]["chart_svg"] == "<svg>chart</svg>"
    assert report["sections"][0]["data_total"] == 100  # single-metric total computed
    assert report["warnings"] == []
    save.assert_awaited_once()
    assert save.await_args.args[0]["report_id"] == report["report_id"]


async def test_failed_section_skipped_with_warning(monkeypatch):
    _patch_cube(monkeypatch, query_result=_query_result([]))  # no data → sections skip
    _patch_llm(
        monkeypatch,
        plan={
            "title": "T",
            "sections": [{"metric": "Sales.revenue_total", "title": "Revenue"}],
        },
    )
    _patch_bridge(monkeypatch)
    _patch_save(monkeypatch)

    import pytest

    with pytest.raises(ReportGenerationError):
        await generate_report("anything", TENANT, USER)


async def test_partial_failure_completes(monkeypatch):
    """One section errors, one succeeds → report completes with warning."""
    results = [
        ConnectionError("cube down"),  # first call raises...
        _query_result([{"region": "N", "order_count": 5}]),
    ]

    async def query_side_effect(**kwargs):
        item = results.pop(0)
        if isinstance(item, Exception):
            raise item
```

```
        return item

    fake_cube = MagicMock()
    fake_cube.get_agent_context = AsyncMock(return_value="catalog")
    fake_cube.query = AsyncMock(side_effect=query_side_effect)
    monkeypatch.setattr("app.semantic.cube_client.get_cube_client", lambda: fake_cube)
    _patch_llm(
        monkeypatch,
        plan={
            "title": "T",
            "sections": [
                {"metric": "Sales.revenue_total", "title": "Revenue"},
                {"metric": "Orders.order_count", "title": "Orders"},
            ],
        },
    )
    _patch_bridge(monkeypatch)
    _patch_save(monkeypatch)

    report = await generate_report("mixed", TENANT, USER)

    assert len(report["sections"]) == 1
    assert report["sections"][0]["metric_name"] == "Orders.order_count"
    assert any("Sales.revenue_total" in w for w in report["warnings"])


async def test_persistence_failure_returns_report_with_warning(monkeypatch):
    _patch_cube(monkeypatch, query_result=_query_result([{"region": "N", "revenue_total": 1}]))
    _patch_llm(
        monkeypatch,
        plan={"title": "T", "sections": [{"metric": "Sales.revenue_total", "title": "R"}]},
    )
    _patch_bridge(monkeypatch)
    _patch_save(monkeypatch, returns=False)

    report = await generate_report("p", TENANT, USER)

    assert report["sections"]
    assert any("not saved" in w for w in report["warnings"])


# ---------------------------------------------------------------------------
# Persistence contract (mocked asyncpg)
# ---------------------------------------------------------------------------


def _patch_asyncpg(monkeypatch):
    conn = MagicMock()
    conn.execute = AsyncMock()
    conn.fetch = AsyncMock(return_value=[])
    conn.fetchrow = AsyncMock(return_value=None)
    tx = MagicMock()
    tx.__aenter__ = AsyncMock(return_value=None)
    tx.__aexit__ = AsyncMock(return_value=False)
    conn.transaction = MagicMock(return_value=tx)
    conn.close = AsyncMock()

    async def fake_connect(dsn=None, **kwargs):
        return conn

    monkeypatch.setattr(reports_service.asyncpg, "connect", fake_connect)
    return conn


def _report_dict():
    return {
        "report_id": "00000000-0000-0000-0000-0000000000cc",
        "title": "T",
        "prompt": "p",
        "summary": "s",
        "status": "complete",
        "sections": [
            {
                "position": 0,
                "metric_name": "Sales.revenue_total",
                "section_title": "Revenue",
                "chart_spec": {"chartType": "Bar Chart"},
                "chart_svg": "<svg/>",
                "data_total": 10.0,
                "row_count": 2,
                "narrative": None,
            }
        ],
    }


async def test_save_report_sets_guc_then_inserts(monkeypatch):
    conn = _patch_asyncpg(monkeypatch)

    ok = await reports_service.save_report(_report_dict(), TENANT, USER)

    assert ok is True
    guc_call = conn.execute.call_args_list[0]
    assert guc_call.args[1] == TENANT
    report_insert = conn.execute.call_args_list[1]
    assert report_insert.args[0].startswith("INSERT INTO reports")
    section_insert = conn.execute.call_args_list[2]
    assert section_insert.args[0].startswith("INSERT INTO report_sections")
    # chart_spec rides as a JSON string for the ::jsonb cast
    section_params = section_insert.args[1:]
    assert json.loads(section_params[5]) == {"chartType": "Bar Chart"}


async def test_save_report_fails_open(monkeypatch):
    conn = _patch_asyncpg(monkeypatch)
```

```
    conn.execute = AsyncMock(side_effect=ConnectionError("db down"))

    assert await reports_service.save_report(_report_dict(), TENANT, USER) is False


async def test_list_reports_scopes_to_user(monkeypatch):
    conn = _patch_asyncpg(monkeypatch)
    conn.fetch = AsyncMock(
        return_value=[
            {
                "id": "r1",
                "title": "T1",
                "created_at": MagicMock(isoformat=lambda: "2026-08-15T00:00:00+00:00"),
                "section_count": 2,
            }
        ]
    )

    rows = await reports_service.list_reports(USER, TENANT, limit=10)

    assert conn.fetch.call_args.args[1] == USER
    assert rows[0]["title"] == "T1"
    assert rows[0]["section_count"] == 2


async def test_get_report_not_found_returns_none(monkeypatch):
    conn = _patch_asyncpg(monkeypatch)
    conn.fetchrow = AsyncMock(return_value=None)

    assert await reports_service.get_report("00000000-0000-0000-0000-0000000000cc", TENANT) is None
```

backend/tests/services/test_schema_retrieval.py

119 lines · 3.8 KB

```python
"""Tests for pgvector schema/few-shot retrieval (app.services.schema_retrieval).

Everything is mocked: the embedding call and the read data source (standard
unittest.mock doubles). These tests pin the retrieval contract: tenant
threading (RLS GUC), vector parameter format, result mapping, and the
fail-open behavior.
"""

import json
from unittest.mock import AsyncMock, MagicMock

from app.services import schema_retrieval
from app.services.schema_retrieval import (
    retrieve_few_shot_examples,
    retrieve_schema_context,
)

TENANT = "00000000-0000-0000-0000-000000000001"


def _patch_embedding(monkeypatch, vector=None, error=None):
    async def fake_embed(text):
        if error:
            raise error
        return vector or [0.1] * 8

    monkeypatch.setattr(schema_retrieval, "embed_text", fake_embed)


def _make_source(monkeypatch, rows):
    """Patch in a mock data source; returns (source, factory)."""
    source = MagicMock()
    source.execute = AsyncMock(return_value=rows)
    factory = MagicMock(return_value=source)
    monkeypatch.setattr("app.connectors.postgresql_connector.PostgreSQLConnector", factory)
    return source, factory


async def test_schema_retrieval_tenant_scoping_and_mapping(monkeypatch):
    _patch_embedding(monkeypatch, vector=[0.25] * 8)
    rows = [
        {
            "full_name": "public.sales",
            "table_description": "Sales transactions",
            "columns_json": [{"column_name": "region", "data_type": "character varying"}],
            "score": 0.92,
        }
    ]
    source, factory = _make_source(monkeypatch, rows)

    context = await retrieve_schema_context("revenue by region", TENANT, top_k=5)

    # The read ran scoped to the tenant (drives the RLS GUC) with a
    # vector-literal parameter and the requested top_k.
    assert factory.call_args.kwargs["tenant_id"] == TENANT
    params = source.execute.call_args.kwargs["params"]
    assert params["top_k"] == 5
    assert params["emb"].startswith("[")
    assert params["emb"].endswith("]")

    # Mapping matches NL2SQLAgent._build_user_message's expected shape
    assert context == [
        {
            "table_name": "public.sales",
            "description": "Sales transactions",
            "columns": [{"column_name": "region", "data_type": "character varying"}],
        }
    ]


async def test_schema_retrieval_parses_string_columns_json(monkeypatch):
    _patch_embedding(monkeypatch)
    rows = [
        {
            "full_name": "public.orders",
            "table_description": "",
            "columns_json": json.dumps([{"column_name": "status"}]),
            "score": 0.8,
        }
    ]
    _make_source(monkeypatch, rows)

    context = await retrieve_schema_context("order status", TENANT)
    assert context[0]["columns"] == [{"column_name": "status"}]


async def test_schema_retrieval_fails_open(monkeypatch):
    _patch_embedding(monkeypatch, error=RuntimeError("no key"))

    assert await retrieve_schema_context("anything", TENANT) == []


async def test_few_shot_retrieval_tenant_scoping_and_mapping(monkeypatch):
    _patch_embedding(monkeypatch)
    rows = [
        {
            "nl_query": "Show me total revenue by region",
            "expected_sql": "<revenue-by-region-query>",
            "score": 0.9,
        }
```

```
    ]
    source, factory = _make_source(monkeypatch, rows)

    examples = await retrieve_few_shot_examples("revenue by region", TENANT, top_k=3)

    assert factory.call_args.kwargs["tenant_id"] == TENANT
    assert source.execute.call_args.kwargs["params"]["top_k"] == 3
    assert examples == [
        {
            "nl_query": "Show me total revenue by region",
            "expected_sql": "<revenue-by-region-query>",
        }
    ]


async def test_few_shot_retrieval_fails_open(monkeypatch):
    _patch_embedding(monkeypatch, error=RuntimeError("api down"))

    assert await retrieve_few_shot_examples("anything", TENANT) == []
```

backend/tests/test_embeddings.py

64 lines · 1.9 KB

```python
"""Unit tests for the embedding provider (app.core.embeddings)."""

from types import SimpleNamespace

import pytest

from app.core.config import settings
from app.core.embeddings import embed_text, vector_literal


class _FakeEmbeddings:
    def __init__(self, vector):
        self._vector = vector

    async def create(self, model, input, dimensions):
        return SimpleNamespace(data=[SimpleNamespace(embedding=self._vector)])


async def test_embed_text_requires_key(monkeypatch):
    monkeypatch.setattr(settings, "OPENAI_API_KEY", "")
    with pytest.raises(RuntimeError, match="OPENAI_API_KEY"):
        await embed_text("hello")


async def test_embed_text_returns_vector(monkeypatch):
    monkeypatch.setattr(settings, "OPENAI_API_KEY", "test-key")
    vector = [0.1] * settings.EMBEDDING_DIMS
    monkeypatch.setattr(
        "openai.AsyncOpenAI",
        lambda api_key=None: SimpleNamespace(embeddings=_FakeEmbeddings(vector)),
    )

    result = await embed_text("total revenue by region")
    assert result == vector


async def test_embed_text_rejects_wrong_dims(monkeypatch):
    monkeypatch.setattr(settings, "OPENAI_API_KEY", "test-key")
    monkeypatch.setattr(
        "openai.AsyncOpenAI",
        lambda api_key=None: SimpleNamespace(embeddings=_FakeEmbeddings([0.1] * 7)),
    )

    with pytest.raises(RuntimeError, match="dims"):
        await embed_text("hello")


async def test_embed_text_wraps_api_errors(monkeypatch):
    monkeypatch.setattr(settings, "OPENAI_API_KEY", "test-key")

    class Boom:
        async def create(self, **kwargs):
            raise ValueError("api down")

    monkeypatch.setattr(
        "openai.AsyncOpenAI", lambda api_key=None: SimpleNamespace(embeddings=Boom())
    )

    with pytest.raises(RuntimeError, match="embedding API call failed"):
        await embed_text("hello")


def test_vector_literal_format():
    assert vector_literal([0.5, -0.25]) == "[0.5,-0.25]"
```

backend/tests/test_security.py

22 lines · 0.6 KB

```python
"""Unit tests for password hashing."""

from app.core.security import hash_password, verify_password


def test_hash_and_verify_roundtrip():
    hashed = hash_password("admin123")
    assert hashed != "admin123"
    assert verify_password("admin123", hashed)


def test_hash_uses_fresh_salt():
    assert hash_password("admin123") != hash_password("admin123")


def test_verify_rejects_wrong_password():
    hashed = hash_password("admin123")
    assert not verify_password("wrong-pass", hashed)


def test_verify_rejects_malformed_hash():
    assert not verify_password("admin123", "not-a-bcrypt-hash")
```

backend/uv.lock

3969 lines · 751.1 KB

```
version = 1
revision = 3
requires-python = ">=3.12"
resolution-markers = [
    "python_full_version >= '3.15' and sys_platform == 'win32'",
    "python_full_version >= '3.15' and sys_platform == 'emscripten'",
    "python_full_version >= '3.15' and sys_platform != 'emscripten' and sys_platform != 'win32'",
    "python_full_version == '3.14.*' and sys_platform == 'win32'",
    "python_full_version == '3.14.*' and sys_platform == 'emscripten'",
    "python_full_version == '3.14.*' and sys_platform != 'emscripten' and sys_platform != 'win32'",
    "python_full_version < '3.14' and sys_platform == 'win32'",
    "python_full_version < '3.14' and sys_platform == 'emscripten'",
    "python_full_version < '3.14' and sys_platform != 'emscripten' and sys_platform != 'win32'",
]

[[package]]
name = "aiohappyeyeballs"
version = "2.7.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/ce/f4/eec0465c2f67b2664688d0240b3212d5196fd89e741df67ddb81f8d35658/aiohappyeyeballs-2.7.1.tar.gz", hash = "sha256:065665c041c42a5938ed220bdcd7230f22527fbec085e1853d2402c8a3615d9d", size = 24757, upload-time = "2026-07-01T17:11:55.501Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/71/43/1947f06babed6b3f1d7f38b0c767f52df66bfb2bc10b468c4a7de9eceff2/aiohappyeyeballs-2.7.1-py3-none-any.whl", hash = "sha256:9243213661e29250eb41368e5daa826fc017156c3b8a11440826b2e3ed376472", size = 15038, upload-time = "2026-07-01T17:11:54.055Z" },
]

[[package]]
name = "aiohttp"
version = "3.14.3"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "aiohappyeyeballs" },
    { name = "aiosignal" },
    { name = "attrs" },
    { name = "frozenlist" },
    { name = "multidict" },
    { name = "propcache" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
    { name = "yarl" },
]
sdist = { url = "https://files.pythonhosted.org/packages/58/d9/22ce5786ac0c1653ae8b6c23bded02c1686d11f0dbb45b31ce128e0df985/aiohttp-3.14.3.tar.gz", hash = "sha256:9491196535a88924a60afd5b5f434b5b203b6cc616250878dbdb223a8f7844bc", size = 7971213, upload-time = "2026-07-23T01:57:27.037Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/18/d4/eb96299230e20acf2efae207cb8d69051f1f68e357e5ea5e479bf6fb097a/aiohttp-3.14.3-cp312-cp312-macosx_10_13_universal2.whl", hash = "sha256:39aded8c7f3b935b54aab1d8d73c70ec0ee2d3ec3b943e0e86611bc150ba47f5", size = 754690, upload-time = "2026-07-23T01:53:47.332Z" },
    { url = "https://files.pythonhosted.org/packages/88/11/e7a70a209eb9a067c0d3212b518a0134e3484f5178c7533878b6b514d469/aiohttp-3.14.3-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:5bcb6ff3fdab1258a192679ff1a05d44f59626430aa05cd1a9d2447423599228", size = 509484, upload-time = "2026-07-23T01:53:51.159Z" },
    { url = "https://files.pythonhosted.org/packages/30/07/4bbc222cc8dbe31d4c3e8a5baad2286e4d42026ac0c570027b89afce6344/aiohttp-3.14.3-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:617105e2c3018ee38d0c8ce5ee3c84f621a6d8b9f723202aacaff28449ca91ee", size = 511949, upload-time = "2026-07-23T01:53:55.083Z" },
    { url = "https://files.pythonhosted.org/packages/54/b9/42e74c46b7b7c794b995bbc1f573fb48950c38b19d8600c62a6804ee2d67/aiohttp-3.14.3-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:f631fe87a6f30df5fbe6d79640b25e4cffb38c31c7fb6f10871517b84b0f8c1a", size = 1765282, upload-time = "2026-07-23T01:53:59.662Z" },
    { url = "https://files.pythonhosted.org/packages/6b/ed/62bc4d74363ad346d518e0720363a949f63e2e23439a79eb5813d4d29bb3/aiohttp-3.14.3-cp312-cp312-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:a94dbaae5ae27bd849c93570669bff91e0510f33a80805738e3de72a7be0447b", size = 1741511, upload-time = "2026-07-23T01:54:04.063Z" },
    { url = "https://files.pythonhosted.org/packages/d0/9f/181e8a8bc79e47d13c7fc4540bd7a3b729d9505609c61f392a8dd2fbfe55/aiohttp-3.14.3-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:8f2f1c4c032c7cedd7d8da6f54c97b70266c6570c3108d3fdffee7188bb70529", size = 1810680, upload-time = "2026-07-23T01:54:09.882Z" },
    { url = "https://files.pythonhosted.org/packages/5c/9a/dec94d6ad694552fe3424e3f1928d7a606a5d9d9433a04e7ecdd9d38ae7f/aiohttp-3.14.3-cp312-cp312-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:ea05e1f97ceea523942d9b2a7d7c0359d781d683d6b043f5943a602b14da4787", size = 1905646, upload-time = "2026-07-23T01:54:13.475Z" },
    { url = "https://files.pythonhosted.org/packages/52/b7/7cd31f29d6055bd711ae6e669367fba6f5ae9de463910a793e30556a8db7/aiohttp-3.14.3-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:543906c127fb1d929b95076db19b83fa2d46751006ff1e23b093aa5ac4d8db42", size = 1792122, upload-time = "2026-07-23T01:54:15.752Z" },
    { url = "https://files.pythonhosted.org/packages/66/73/10b1ef93afa61f4963c746257b70ced619cf31a4798671de5fdb2608501d/aiohttp-3.14.3-cp312-cp312-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:0a5ff2dfbb9ce645fa5b8ef3e02c6c0b9cc3f6030ff863d0c51fffc50cb5541b", size = 1591127, upload-time = "2026-07-23T01:54:19.489Z" },
    { url = "https://files.pythonhosted.org/packages/49/ed/3b203fa6de1b338c14acdc06bf6ca9b043b7944f005966958c2ced932cde/aiohttp-3.14.3-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:041badb8f84396357c4d3ad26de6afd7a32b112f43d3c63045c0c8278cfd2043", size = 1725210, upload-time = "2026-07-23T01:54:24.129Z" },
    { url = "https://files.pythonhosted.org/packages/28/b7/1c2aab8c706436dcc28598452488ac9cd7c409da815237c28c27d58993e6/aiohttp-3.14.3-cp312-cp312-musllinux_1_2_armv7l.whl", hash = "sha256:530125ee1163c4219af35dc3aa1206e541e7b31b6efc1a3f93b70a136f65d427", size = 1764848, upload-time = "2026-07-23T01:54:27.973Z" },
    { url = "https://files.pythonhosted.org/packages/54/50/94c28f08b131c4bf10984ea2c7a536c9920608bb2d6e7f95642c30cc87b7/aiohttp-3.14.3-cp312-cp312-musllinux_1_2_ppc64le.whl", hash = "sha256:c8653fd547c93a61aadc612007790f5555cdd18946fa48cf45e26d8ea4ea473d", size = 1777102, upload-time = "2026-07-23T01:54:31.775Z" },
    { url = "https://files.pythonhosted.org/packages/13/d4/e7d09ba7d345fb2d74440fd2fa033c5e079fac05552927705986f41a364f/aiohttp-3.14.3-cp312-cp312-musllinux_1_2_riscv64.whl", hash = "sha256:89176250f686cb9853c0fb7ead90e639e915b84a6f43eedc2a4e7ec21f1037f0", size = 1580205, upload-time = "2026-07-23T01:54:34.518Z" },
    { url = "https://files.pythonhosted.org/packages/a3/84/072a91d68e1e1eb587985b54baab94221277f877e8ef274fc213a0ceae28/aiohttp-3.14.3-cp312-cp312-musllinux_1_2_s390x.whl", hash = "sha256:3a26434dafe408229ff3403458ca58de24fb51936504decac49ce6755f77e59d", size = 1797219, upload-time = "2026-07-23T01:54:36.995Z" },
    { url = "https://files.pythonhosted.org/packages/e0/eb/aad34e897e668424d6e995da5dff8a4a09af93363d3392488772957a63aa/aiohttp-3.14.3-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:d1558173930a5a8d3069cee5c92fc91c87c4dbcb099debbb3622053717145a19", size = 1768629, upload-time = "2026-07-23T01:54:40.103Z" },
    { url = "https://files.pythonhosted.org/packages/b6/2b/6bb88ddba0fecd9122aa3ebcad25996cf6c083a4a7040dbb3a4f97972af6/aiohttp-3.14.3-cp312-cp312-win32.whl", hash = "sha256:16100ad3ab8d649fdfbee87602d9d2dcdca9df0b9eda8a1b5fdc0d41f96da559", size = 451481, upload-time = "2026-07-23T01:54:42.547Z" },
    { url = "https://files.pythonhosted.org/packages/76/9b/f2f8f108da17ecef2cc3efc424e8b7ad3782b1a8360f7b8eae8ced84f6ea/aiohttp-3.14.3-cp312-cp312-win_amd64.whl", hash = "sha256:33a2d7c28d33797a2e99923dffa63f83d908a19b6bf26cfe80fa790aa5e1a75a", size = 476845, upload-time = "2026-07-23T01:54:44.853Z" },
    { url = "https://files.pythonhosted.org/packages/3e/44/28dac80a8941b604f4da10ce21097614ca1bf905ce93dca28d8d7de9c1e7/aiohttp-3.14.3-cp312-cp312-win_arm64.whl", hash = "sha256:362a3fd481769cac1a824514bcd86fda51c65e8fe6e051099e008fddde6db17c", size = 448050, upload-time = "2026-07-23T01:54:47.087Z" },
    { url = "https://files.pythonhosted.org/packages/57/be/5afd201cc0ab139029aadb75392efe85a293403d9dd3a3226161c21ce00c/aiohttp-3.14.3-cp313-cp313-android_21_arm64_v8a.whl", hash = "sha256:2e9878ae68e4a5f1c0abe4dd497dbc3d51946f5837b56759e2a02e78fa90ef86", size = 506269, upload-time = "2026-07-23T01:54:49.075Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/22/09/dec8189d62b45ade009f6792a2264b942a90cb88aeaf181239933cd72c3c/aiohttp-3.14.3-cp313-cp313-and
roid_21_x86_64.whl", hash = "sha256:f3d2669fe7dec7fc359ecdb5984b29b50d85d5d00f8c1cb61de4f4a24ee42627", size = 515166, upload-time = "2026-07-23T01:54:
51.894Z" },
    { url = "https://files.pythonhosted.org/packages/28/24/2854869d29ed8a8b19d74f9ec6629515f7e04d02dd329d9d179201e58e47/aiohttp-3.14.3-cp313-cp313-ios
_13_0_arm64_iphoneos.whl", hash = "sha256:cc7cb243a68167172f48c1fd43cee91ec4b1d40cefd190edd43369d1a6bc9c82", size = 486263, upload-time = "2026-07-23T
01:54:54.223Z" },
    { url = "https://files.pythonhosted.org/packages/d4/dd/57187c8be2a35aea65eaee3bd2c3dcbbcf0204f5106c89637e3610380cd1/aiohttp-3.14.3-cp313-cp313-ios
_13_0_arm64_iphonesimulator.whl", hash = "sha256:78253b573e6ffab5028924fc98bc281aae05445969982a10864bc360dea2016c", size = 492299, upload-time = "2026
-07-23T01:54:56.236Z" },
    { url = "https://files.pythonhosted.org/packages/b9/11/06ae6ed8f0d414edf4068861e233d8fe23ee699bfd4b3ceb8663db948a62/aiohttp-3.14.3-cp313-cp313-ios
_13_0_x86_64_iphonesimulator.whl", hash = "sha256:7041d52c3a7fa20c9e8c182b534704abb19502c8bdcbde7ab23bfda6f642394f", size = 502235, upload-time = "202
6-07-23T01:54:58.377Z" },
    { url = "https://files.pythonhosted.org/packages/7e/a3/559639c34a345d2cf7c52dff6838119f2eaf29eb508227b5b83f573af813/aiohttp-3.14.3-cp313-cp313-mac
osx_10_13_universal2.whl", hash = "sha256:ac74facc01463f138b0da5580329cfcc82818dea5656e83ddcd11268fc12ff80", size = 750883, upload-time = "2026-07-23T
01:55:00.65Z" },
    { url = "https://files.pythonhosted.org/packages/91/cd/41e131f13afd1e7b0172a9d9eda085ef90eb8439f41f0d279db81ed3ae60/aiohttp-3.14.3-cp313-cp313-mac
osx_10_13_x86_64.whl", hash = "sha256:d6218d92e450824e9b4881f44e8c09f1853b490f9a64130801024a4793b1b3b0", size = 508473, upload-time = "2026-07-23T01:5
5:02.945Z" },
    { url = "https://files.pythonhosted.org/packages/bc/6b/e7f13410d391c6e55b4c007a8de024355389d7d459e3d64c42b2d33617e5/aiohttp-3.14.3-cp313-cp313-mac
osx_11_0_arm64.whl", hash = "sha256:11fb37ef075669eee52ab1928fbf6e1741fada40409fa309ebde9607a962aebf", size = 509190, upload-time = "2026-07-23T01:55:
05.173Z" },
    { url = "https://files.pythonhosted.org/packages/97/21/6464573e53d69672cc1eada3e5c5cb2d2efa82701e8305a0f2047a576967/aiohttp-3.14.3-cp313-cp313-man
ylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:55bdcc472aafe2de4a253045cc128007a64f1e0264fb675791e132ea5edaa3bd
", size = 1761478, upload-time = "2026-07-23T01:55:07.383Z" },
    { url = "https://files.pythonhosted.org/packages/1a/81/d217043a4c17fbce360905e3b2bdd20139ebc9a2de836d035d179c4da006/aiohttp-3.14.3-cp313-cp313-man
ylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:c39846c3aad97a8530c89d7a3869a8f8e9e3762c6ac0504481e5c80948f7e807",
size = 1735092, upload-time = "2026-07-23T01:55:09.803Z" },
    { url = "https://files.pythonhosted.org/packages/a1/66/e13a02d0eeb1a9a502402a977abb4e4abff9fe4051c26f80558c57a7c975/aiohttp-3.14.3-cp313-cp313-man
ylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:5895ef58c4620afe02fa16044f023dc4dafec08158f9d08874a46a7dbc0341b8
", size = 1800546, upload-time = "2026-07-23T01:55:12.012Z" },
    { url = "https://files.pythonhosted.org/packages/26/5e/57d42fca1d18cb5acc1cad945d017fabc5d6ae71d8a08ad66be8dc3ee544/aiohttp-3.14.3-cp313-cp313-man
ylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:fa9467a8113aa69d3d7c55a70ef0b7c636010a40993f3df9d9d0d73b3eb7ef24", siz
e = 1895250, upload-time = "2026-07-23T01:55:14.357Z" },
    { url = "https://files.pythonhosted.org/packages/ca/1c/7da8d08e74d56f00070822f9638ff3f1c563f8ad87d1efa996c87bfc8644/aiohttp-3.14.3-cp313-cp313-man
ylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:d7d2deec16eeedf55f2c7cf75b521ea3856a5177e123844f8fd0f114ce252cb5",
size = 1789289, upload-time = "2026-07-23T01:55:16.668Z" },
    { url = "https://files.pythonhosted.org/packages/cd/0f/cf16bcf56896981c1a0319f5d5db9337994b5165730c48a8fa07e9b34be6/aiohttp-3.14.3-cp313-cp313-man
ylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:dd54d0e8717de95939766febac482ac0474d8ac3b048115f9f2b1d23a16e7db4", size = 1586706, upl
oad-time = "2026-07-23T01:55:18.913Z" },
    { url = "https://files.pythonhosted.org/packages/fe/6f/76eac12a7f2480e1e304f842efdb07db33256b0d9165b866b6ef0806c202/aiohttp-3.14.3-cp313-cp313-mus
llinux_1_2_aarch64.whl", hash = "sha256:df82f3787c940c94986b34222d59c9e38843fba85139f36e85255a82ad5355a9", size = 1724652, upload-time = "2026-07-23T0
1:55:21.296Z" },
    { url = "https://files.pythonhosted.org/packages/39/b6/19c8c592baeeb94b75f966547d40c02ac7590902306ec5863d5c027cf506/aiohttp-3.14.3-cp313-cp313-mus
llinux_1_2_armv7l.whl", hash = "sha256:42a67efc36300d052fb4508a53e8b6901b9284b599ae63945c377569c5fcc1e1", size = 1756239, upload-time = "2026-07-23T01
:55:23.705Z" },
    { url = "https://files.pythonhosted.org/packages/dc/c9/4e9383150296f97f873b680c4de8fb2cd88608fb9f48c79edcb111611abc/aiohttp-3.14.3-cp313-cp313-mus
llinux_1_2_ppc64le.whl", hash = "sha256:7a75aa63cbf9b21cfaf60dc2657e19df2c2867d91707d653fee171ffeedd1371", size = 1769161, upload-time = "2026-07-23T0
1:55:26.082Z" },
    { url = "https://files.pythonhosted.org/packages/aa/1e/147bdc6cc5de5f3ab011be8bf5d6e786633249f22c20bae06f85e45f5387/aiohttp-3.14.3-cp313-cp313-mus
llinux_1_2_riscv64.whl", hash = "sha256:e92eb8acc45eb6a9f4935071a77edf5b85cc6f8dfad5cd99e97653c26593cdde", size = 1578759, upload-time = "2026-07-23T0
1:55:28.846Z" },
    { url = "https://files.pythonhosted.org/packages/fd/31/78388a9d6040ece2e11df62ea229a822cf5e52d238374b220ae9975b2623/aiohttp-3.14.3-cp313-cp313-mus
llinux_1_2_s390x.whl", hash = "sha256:b014a6ed7cf912e787149fdc529166d3ceabac23f26efeea3158c9aba2354e7e", size = 1792025, upload-time = "2026-07-23T01:
55:31.457Z" },
    { url = "https://files.pythonhosted.org/packages/03/51/a3d29fdf2c25d796746af8ad6fe56a45d6256c38b0a8a2ed752e1160b3a2/aiohttp-3.14.3-cp313-cp313-mus
llinux_1_2_x86_64.whl", hash = "sha256:3d4f72af88ac2474bb5bca640030320e3d38a0163a1d7533500e87be458eef71", size = 1768477, upload-time = "2026-07-23T01
:55:33.87Z" },
    { url = "https://files.pythonhosted.org/packages/29/a6/442e18b5afeade534d877a2dc3c3e392aff8d49787890b0cf84790410267/aiohttp-3.14.3-cp313-cp313-win
32.whl", hash = "sha256:5f08ec777f35ee70720233b8b9811d3bb5d728137f30ac91b7457709c3261ac0", size = 451069, upload-time = "2026-07-23T01:55:36.121Z" },
    { url = "https://files.pythonhosted.org/packages/9d/69/3d876ac02659f271cf7f6769f14a8e3de5b6e888ed8b5a7e998086a4cec8/aiohttp-3.14.3-cp313-cp313-win
_amd64.whl", hash = "sha256:dff9461ec275f22135650d5ba4b4931a11f3958df7dfbb8db630000d4dee0883", size = 476518, upload-time = "2026-07-23T01:55:38.303Z"
 },
    { url = "https://files.pythonhosted.org/packages/b2/0e/50d6e6471cd31edce8b282bdec59375a3a69124d8a989a0b1313355cae52/aiohttp-3.14.3-cp313-cp313-win
_arm64.whl", hash = "sha256:ddcac3c6b382e81f1dd0499199d4136b877beb4cb5ef770bbbfba56c4b8f55d2", size = 447676, upload-time = "2026-07-23T01:55:40.451Z"
 },
    { url = "https://files.pythonhosted.org/packages/c8/20/887fdcf832326571b370ffc347b3e70abe101096f3720126aac161b1d872/aiohttp-3.14.3-cp314-cp314-and
roid_24_arm64_v8a.whl", hash = "sha256:49f7325beb0f85ef4aef5f48f490269575f83e6e2acad00a1d80b807eb027062", size = 509067, upload-time = "2026-07-23T01:
55:42.618Z" },
    { url = "https://files.pythonhosted.org/packages/ad/a3/92cec936f78cc4bf0fa5554ebe593b73459d94e3c62303e1902a4cccb6f7/aiohttp-3.14.3-cp314-cp314-and
roid_24_x86_64.whl", hash = "sha256:e3be98a7c30b8c25d573dafba7171d66dfb05ee6a9070fc46535464ff97700a6", size = 514774, upload-time = "2026-07-23T01:55:
44.937Z" },
    { url = "https://files.pythonhosted.org/packages/29/ba/2a0c38df3fc557620b6a5acd98364af050053b6285b4dc7ee74100c63c18/aiohttp-3.14.3-cp314-cp314-ios
_13_0_arm64_iphoneos.whl", hash = "sha256:614c61d478b83953e261d02bb2df750f17227cd33ef8002945bf5aebbde21919", size = 488134, upload-time = "2026-07-23T
01:55:47.135Z" },
    { url = "https://files.pythonhosted.org/packages/48/d6/d51b7d4bf309af3693940d8ffd2b9ed0b682434ef85959b7c9c137f60cf8/aiohttp-3.14.3-cp314-cp314-ios
_13_0_arm64_iphonesimulator.whl", hash = "sha256:1caa7b0d05f3e3a36f87788c59e970a7ee1cefcfcbb924a9f138c4a6551c9cb7", size = 494201, upload-time = "2026
-07-23T01:55:49.451Z" },
    { url = "https://files.pythonhosted.org/packages/3f/5a/8f624384e5f1efabb5229b94157eb966b021e97bdb188c62860c2ae243c2/aiohttp-3.14.3-cp314-cp314-ios
_13_0_x86_64_iphonesimulator.whl", hash = "sha256:dfa68deb2a443bdaa3ea5297b0699c1464f08aef3812b486d1348eee61b07dc0", size = 502766, upload-time = "202
6-07-23T01:55:51.656Z" },
    { url = "https://files.pythonhosted.org/packages/a6/26/4ff0164370deec18fb19254ee4ab10b7a73304ac0c860b13f5f84663759b/aiohttp-3.14.3-cp314-cp314-mac
osx_10_15_universal2.whl", hash = "sha256:e72ee89e28d907a18f46959b4eb0bb06701cc7f8cf4366e00029e2ccfaaf5924", size = 756557, upload-time = "2026-07-23T
01:55:53.964Z" },
    { url = "https://files.pythonhosted.org/packages/97/a3/7056b86dc0d9ec709ea9777eae3b0161428f943372f8b98c01c11593b682/aiohttp-3.14.3-cp314-cp314-mac
osx_10_15_x86_64.whl", hash = "sha256:ad4c8b7488d745d2ca4838ebd8ae5ba9b56341d30b1da43640e4ce87f9f49646", size = 510168, upload-time = "2026-07-23T01:5
5:56.22Z" },
    { url = "https://files.pythonhosted.org/packages/85/ed/0357a015892fd68058bf2d39d3fd1958e459b997a7db30aaa6aaa434ae96/aiohttp-3.14.3-cp314-cp314-mac
osx_11_0_arm64.whl", hash = "sha256:db332af25642007330fca8be5c4d194caf2bea7a7fc84415aff3497af5dfee6b", size = 512957, upload-time = "2026-07-23T01:55:
58.437Z" },
    { url = "https://files.pythonhosted.org/packages/47/d1/8aba53f15ccb2238405f5e9d30e2a8ca44f93878c26e7165ade00d374b1c/aiohttp-3.14.3-cp314-cp314-man
ylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:25bd2708db6bdf6a6630dd37bdcdfcb47c4434d22ac69c64665b802910140b30
", size = 1750149, upload-time = "2026-07-23T01:56:00.856Z" },
    { url = "https://files.pythonhosted.org/packages/49/bd/40c3fee32752928437 5c6701cbb0fa4600cc2e8432af1378f897e2ef7d3a/aiohttp-3.14.3-cp314-cp314-man
ylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:cef89a58e628c4efcac3275c2d68083f82426dcdc89c1492a6f654f9f7ea6ab9",
size = 1707685, upload-time = "2026-07-23T01:56:03.371Z" },
    { url = "https://files.pythonhosted.org/packages/2a/a3/ca0cc6724cca8114b05694abd916060758c79894c3aa5b012cdadc1bc28e/aiohttp-3.14.3-cp314-cp314-man
ylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:c23ec8ee9d5ab2f5421f9c7fffce208435607af27fd46d4a44e031954352838f
", size = 1803911, upload-time = "2026-07-23T01:56:05.817Z" },
    { url = "https://files.pythonhosted.org/packages/95/b5/85b099c299c3ffd38ad9b3e43694c8a346934e4a30c88c4fd5a841234f77/aiohttp-3.14.3-cp314-cp314-man
ylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:e2667f0bbe7eb6c74eae5e9691441ad186e5845ca3cff63230fc09c4e7514f5d", siz
e = 1876929, upload-time = "2026-07-23T01:56:08.413Z" },
    { url = "https://files.pythonhosted.org/packages/d5/b7/1da684a04175473fa4cddbf9a2f572e79514c3fd27a74597f43057d4f3da/aiohttp-3.14.3-cp314-cp314-man
ylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:18cb43369747b2ae007bd2655fb8e63a099c2ff1d207962943636dac989b3147",
size = 1761112, upload-time = "2026-07-23T01:56:10.918Z" },
    { url = "https://files.pythonhosted.org/packages/d1/16/bc4b55e3e5cb175fd69c53c90d60d2f47797cb343da5106e23863dc4dba4/aiohttp-3.14.3-cp314-cp314-man
```

```
ylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:d77640cc618c1d99fc4f8589c0f24a730adfa54eb1e57ef7bf0c8dfb78da898c", size = 1583500, upl
oad-time = "2026-07-23T01:56:13.613Z" },
    { url = "https://files.pythonhosted.org/packages/2a/e8/13a9d957a1ee40837f46aa30f0f4c657e673ad86a2e6362a9f9be20d26d9/aiohttp-3.14.3-cp314-cp314-mus
llinux_1_2_aarch64.whl", hash = "sha256:53e5179d8abb5710f8e83ba207c41c8d1261fcffd4616500e15ca2b7a33be10a", size = 1713940, upload-time = "2026-07-23T0
1:56:15.969Z" },
    { url = "https://files.pythonhosted.org/packages/38/05/d33c680c1bcf1c7e130f9cbfc1fc02fe8bb0c4af2a94a53dd5fb56131e5c/aiohttp-3.14.3-cp314-cp314-mus
llinux_1_2_armv7l.whl", hash = "sha256:cd817772b2fcf2b8c0905795318485f9ec16eae60b29feb7f4c77085311637f0", size = 1724413, upload-time = "2026-07-23T01
:56:18.591Z" },
    { url = "https://files.pythonhosted.org/packages/85/1d/af798d306f7a74b6a632dbcabcf62a4c91391b7582d2a8c6d7712e2cc54e/aiohttp-3.14.3-cp314-cp314-mus
llinux_1_2_ppc64le.whl", hash = "sha256:4e3ac92d90e92773b2362d506068e9a948192bd553e743c5b2429e28527c8661", size = 1770748, upload-time = "2026-07-23T0
1:56:21.074Z" },
    { url = "https://files.pythonhosted.org/packages/a8/92/ad720d472556a995049206867765e9410969684f86ee09423ff9969044c1/aiohttp-3.14.3-cp314-cp314-mus
llinux_1_2_riscv64.whl", hash = "sha256:3f42e9b78301f11c8f861746175d8b9c1ccef713fcad9eab396e2f6db8ed4a22", size = 1577564, upload-time = "2026-07-23T0
1:56:23.475Z" },
    { url = "https://files.pythonhosted.org/packages/60/ad/0ed7586cbef7a884e23a752fa2bb987a122e6a5dd50dab109258d0a95193/aiohttp-3.14.3-cp314-cp314-mus
llinux_1_2_s390x.whl", hash = "sha256:9d9edccfe496b476db5f398d97b865e9a6752bcf8aec4eef8390ce20fb64bb41", size = 1782080, upload-time = "2026-07-23T01:
56:25.994Z" },
    { url = "https://files.pythonhosted.org/packages/97/ea/dbaed0d73e8a69aad653b045dab451c67c2454bb731a37b45a86593e9422/aiohttp-3.14.3-cp314-cp314-mus
llinux_1_2_x86_64.whl", hash = "sha256:1c5ec8fb1bcc31a8466f74aaf26c345d5c386fa4bd08a3f0eb9c7a4a3fe8b5bf", size = 1745813, upload-time = "2026-07-23T01
:56:28.604Z" },
    { url = "https://files.pythonhosted.org/packages/81/1b/6893d4bc57e434fc93a6c9217c637d967a0b651d989f6e3265179375754a/aiohttp-3.14.3-cp314-cp314-win
32.whl", hash = "sha256:38901a84da3ce22249f6e860bf8f90d141bcab7da090cc398f8bb58c0e44b7da", size = 455872, upload-time = "2026-07-23T01:56:31.031Z" },
    { url = "https://files.pythonhosted.org/packages/f5/8b/c7baa1ba1eda4db6989baefe5de6d99834921b84ebd7918624febcb9f290/aiohttp-3.14.3-cp314-cp314-win
_amd64.whl", hash = "sha256:8b3b60de05f3dcb6f6a00f818bb2ec781cee4de0645f59ccaf99b1d1823b6100", size = 481030, upload-time = "2026-07-23T01:56:33.365Z"
 },
    { url = "https://files.pythonhosted.org/packages/22/8c/c29d067df825a2df88ca432db848aa2fe8199598359cc06c12b09320cac9/aiohttp-3.14.3-cp314-cp314-win
_arm64.whl", hash = "sha256:1576145bdceeb92382d899751e12743a3a5b8e460a841e3e50543859e54864dc", size = 453669, upload-time = "2026-07-23T01:56:35.731Z"
 },
    { url = "https://files.pythonhosted.org/packages/6a/a4/9c033beb355d39b6147980597ec9645e4729243f686ee4dc73945de72030/aiohttp-3.14.3-cp314-cp314t-ma
cosx_10_15_universal2.whl", hash = "sha256:8800c996b01c2772a783e3e46f3e1abd5823029adca0df54231960de9bfefa5b", size = 791403, upload-time = "2026-07-23
T01:56:37.972Z" },
    { url = "https://files.pythonhosted.org/packages/80/ca/87c32a0a7704583cfc49660bd817889bae5b830bf53b5dcb4e92145ac2da/aiohttp-3.14.3-cp314-cp314t-ma
cosx_10_15_x86_64.whl", hash = "sha256:ebe8e504f058fe91223351cecd2d9d6946c9d241bb0250d898ffbdf584cc72b0", size = 526413, upload-time = "2026-07-23T01:
56:40.523Z" },
    { url = "https://files.pythonhosted.org/packages/9e/d8/8ec0e471248c500acdce2be3f46db8fb62b5eb60efef072529cc85ee1d26/aiohttp-3.14.3-cp314-cp314t-ma
cosx_11_0_arm64.whl", hash = "sha256:30402d03a7c0ff52bce290b57e564e9079fd9d0cb545c8aba73f86a103162d2e", size = 532135, upload-time = "2026-07-23T01:56
:42.876Z" },
    { url = "https://files.pythonhosted.org/packages/fe/45/f8919fd936e8b79fcd9bda7b6d8e62613462a713f4f17987fd7c34399142/aiohttp-3.14.3-cp314-cp314t-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:9fc7b5bfec6573f3ae844f457fdde5adeb713f8b8e4a81ad64fc207b4938371
6", size = 1922742, upload-time = "2026-07-23T01:56:45.528Z" },
    { url = "https://files.pythonhosted.org/packages/f6/ec/9ca76b28a27525b0cc53e20842e0228b022f301ce1f436b7d814b4aaf2df/aiohttp-3.14.3-cp314-cp314t-ma
nylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:8a5fd34f7f7410d1730d5c2ba873cacb2eed3fede366feb268a70ba22581ed8f",
 size = 1787371, upload-time = "2026-07-23T01:56:48.045Z" },
    { url = "https://files.pythonhosted.org/packages/b1/04/6acdbf17315f7b55f1937e3387acb89a3cddeb4995689553d064af8e92ab/aiohttp-3.14.3-cp314-cp314t-ma
nylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:270d3dace9ca2f10f0da5d8ebe519b7a310fc6112ed916e32df5866df088855
3", size = 1912623, upload-time = "2026-07-23T01:56:50.605Z" },
    { url = "https://files.pythonhosted.org/packages/86/e6/438b0c79ca6f45eb9fd9817dd4c01a91919a38c0de5ee9e05e2b4dc0ece7/aiohttp-3.14.3-cp314-cp314t-ma
nylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:3ae5b3a59436d089b5395d910121a390feed4d00578eb95a0fd1a329fe963100", si
ze = 2005515, upload-time = "2026-07-23T01:56:53.153Z" },
    { url = "https://files.pythonhosted.org/packages/bb/6b/62cbd6577758699525f5c712d1ddef57d9875fbab0ae8d5f5a202fd598f8/aiohttp-3.14.3-cp314-cp314t-ma
nylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:2498f0fe69ead802f9675beca44a7c21c62fdaa4ec5145ea1c3ad6edbee29f85",
 size = 1879906, upload-time = "2026-07-23T01:56:55.818Z" },
    { url = "https://files.pythonhosted.org/packages/00/95/18bcbf830a21dc3aae24d8f6b6feaf3db1d2090242d00a7868db2ffb0b67/aiohttp-3.14.3-cp314-cp314t-ma
nylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:a0dc483c00da8b673abbb367eb6f8d8f4bcec30eb58529ea13cb42e7fd2dfa33", size = 1675849, up
load-time = "2026-07-23T01:56:58.861Z" },
    { url = "https://files.pythonhosted.org/packages/a9/19/47f4968659c5e23606c3790c80fc624e691c153d036148449ee84d31b287/aiohttp-3.14.3-cp314-cp314t-mu
sllinux_1_2_aarch64.whl", hash = "sha256:c7d3a97c678d34fc5b59da671ee9cd630096ddc643e7b5a30d54a2a6f3574d3f", size = 1843496, upload-time = "2026-07-23T
01:57:01.591Z" },
    { url = "https://files.pythonhosted.org/packages/64/af/38c33c4dd82fddcb4e56c4653b6f1072a8edbc6b7fa15809f14932c41e2d/aiohttp-3.14.3-cp314-cp314t-mu
sllinux_1_2_armv7l.whl", hash = "sha256:f8fb78a83c9e5f741ca3a68cfb455c1f5bb83b4e7249a3848b3cd78d0a8563b0", size = 1827746, upload-time = "2026-07-23T0
1:57:05.131Z" },
    { url = "https://files.pythonhosted.org/packages/a1/9d/0537cda4885ac8f5b7053d164dd06312f4c483a4edcb8ee5b8aaf2a989bf/aiohttp-3.14.3-cp314-cp314t-mu
sllinux_1_2_ppc64le.whl", hash = "sha256:74ab5b6a9fb13e873e5a90946588baecaf488745e1db1a4a5c433f971f035098", size = 1853810, upload-time = "2026-07-23T
01:57:08.043Z" },
    { url = "https://files.pythonhosted.org/packages/19/fe/26f9c5e6458385aa86497836b0dea6fb2f027827d63f37c7856cce9286ee/aiohttp-3.14.3-cp314-cp314t-mu
sllinux_1_2_riscv64.whl", hash = "sha256:bd52f811e65f6fb634b1047159657c98f52b407f8efec907bcfc09da9a4c0a25", size = 1668895, upload-time = "2026-07-23T
01:57:10.837Z" },
    { url = "https://files.pythonhosted.org/packages/ec/4c/618b1db9b9ba079b8875d2cdf78e7c4a3bf72903bd5850fee7dd9544600a/aiohttp-3.14.3-cp314-cp314t-mu
sllinux_1_2_s390x.whl", hash = "sha256:f0f177d1b195b9e06376cfd7d308d8a1b920909a609d03ac82a8c73bbb16d3b9", size = 1883833, upload-time = "2026-07-23T01
:57:13.672Z" },
    { url = "https://files.pythonhosted.org/packages/94/c6/bd959bd1e4771f9fd944e9e436224c48c77b018b73b519b5aad346335bcc/aiohttp-3.14.3-cp314-cp314t-mu
sllinux_1_2_x86_64.whl", hash = "sha256:498c6c623134f8e09a3c4e60bcd607a0b4590dd7dbf08dd40851b27cbb520ccb", size = 1844251, upload-time = "2026-07-23T0
1:57:16.593Z" },
    { url = "https://files.pythonhosted.org/packages/5e/19/08d41839658bdd44a0ed2480f3891705ecb487ce28c0dde62c9040c997e0/aiohttp-3.14.3-cp314-cp314t-wi
n32.whl", hash = "sha256:b304db572b4368edd8dda8a2274f73156fe15558fca4a917cb8a09fc47af5963", size = 474180, upload-time = "2026-07-23T01:57:19.306Z" },
    { url = "https://files.pythonhosted.org/packages/99/5d/3cd6ef0a2b2851f7ab913b5b079334781bd50ff56a323e4454063377a080/aiohttp-3.14.3-cp314-cp314t-wi
n_amd64.whl", hash = "sha256:b20032766aedf6261c7a566585a40867d092ac03a0d81592d5370ef9b054f99b", size = 500528, upload-time = "2026-07-23T01:57:21.762Z
" },
    { url = "https://files.pythonhosted.org/packages/a4/37/cfd1ed540a4d318da025590d96b728e63713c09e9377950fc655dadeb856/aiohttp-3.14.3-cp314-cp314t-wi
n_arm64.whl", hash = "sha256:2e1161602f45a54de2ce0905243a95f58cb42dcd378402f3697f5e0b21e9d2e7", size = 469280, upload-time = "2026-07-23T01:57:24.241Z
" },
]

[[package]]
name = "aiosignal"
version = "1.4.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "frozenlist" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/61/62/06741b579156360248d1ec624842ad0edf697050bbaf7c3e46394e106ad1/aiosignal-1.4.0.tar.gz", h
ash = "sha256:f47eecd9468083c2029cc99945502cb7708b082c232f9aca65da147157b251c7", size = 25007, upload-time = "2025-07-03T22:54:43.528Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/fb/76/641ae371508676492379f16e2fa48f4e2c11741bd63c48be4b12a6b09cba/aiosignal-1.4.0-py3-none-any.w
hl", hash = "sha256:053243f8b92b990551949e63930a839ff0cf0b0ebbe0597b0f3fb19e1a0fe82e", size = 7490, upload-time = "2025-07-03T22:54:42.156Z" },
]

[[package]]
name = "aiosqlite"
version = "0.22.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/4e/8a/64761f4005f17809769d23e518d915db74e6310474e733e3593cfc854ef1/aiosqlite-0.22.1.tar.gz",
hash = "sha256:043e0bd78d32888c0a9ca90fc788b38796843360c855a7262a532813133a0650", size = 14821, upload-time = "2025-12-23T19:25:43.997Z" }
wheels = [
```

```
    { url = "https://files.pythonhosted.org/packages/00/b7/e3bf5133d697a08128598c8d0abc5e16377b51465a33756de24fa7dee953/aiosqlite-0.22.1-py3-none-any.whl", hash = "sha256:21c002eb13823fad740196c5a2e9d8e62f6243bd9e7e4a1f87fb5e44ecb4fceb", size = 17405, upload-time = "2025-12-23T19:25:42.139Z" },
]

[[package]]
name = "alembic"
version = "1.19.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "mako" },
    { name = "sqlalchemy" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/fb/01/a48dab7827ac4421272399f7ed9a2ec17edd12c8bcde4417bd7b6821b71a/alembic-1.19.0.tar.gz", hash = "sha256:6487c612fc719dcfa22b17d2dd5b2b458929641e6aa2f0b65b135727f5e6d501", size = 2069906, upload-time = "2026-08-04T18:57:04.599Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/34/79/59ab6f1fe72eee229de91aa393225389a85df12a0fbbbfa264efcf6d7872/alembic-1.19.0-py3-none-any.whl", hash = "sha256:cf839d3849116aab3cc047e09c6968b9bd6b2fde61b6bb7c1e97352fe5503580", size = 265738, upload-time = "2026-08-04T18:57:06.219Z" },
]

[[package]]
name = "annotated-doc"
version = "0.0.5"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/5a/8e/38aa427ed5402449e226975b649c5dc73ccadfefeb95e6aecb8f8ea4b6b6/annotated_doc-0.0.5.tar.gz", hash = "sha256:c7e58ce09192557605d8bbd92836d7e1d520ac9580096042c0bfd197efacf1bb", size = 10758, upload-time = "2026-07-28T13:50:58.129Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/3e/30/e900b21425a860e195f32e37657aa1f7c7f2b1bfb26f03ca209b90933c06/annotated_doc-0.0.5-py3-none-any.whl", hash = "sha256:117bac03a25ede5df5440e855b32d556049ca169ead221505badf432fed4b101", size = 5302, upload-time = "2026-07-28T13:50:57.239Z" },
]

[[package]]
name = "annotated-types"
version = "0.8.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/5f/56/a8120250d128bed162cd73c76d45f6ef9991f3e068f62a8ee060afa3104a/annotated_types-0.8.0.tar.gz", hash = "sha256:13b2beaad985e05e2d6407ee4c4f35590b11f8d693a258a561055cac8f64cab7", size = 15893, upload-time = "2026-07-23T20:16:13.995Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/99/91/8acff4f5e50511b911bbccb72b8628a49c68ce14148cd9f6431094859a90/annotated_types-0.8.0-py3-none-any.whl", hash = "sha256:f072f4d804ea359e4eaf198b1af7a8b0943881a87f31bb764f8bf219bb9419e0", size = 13427, upload-time = "2026-07-23T20:16:12.938Z" },
]

[[package]]
name = "anthropic"
version = "0.120.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "distro" },
    { name = "docstring-parser" },
    { name = "httpx" },
    { name = "jiter" },
    { name = "pydantic" },
    { name = "sniffio" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/d7/10/4ca013cb166f226bd89e0aeb0fcaff94f45ddf716d4925ce89475d3c587b/anthropic-0.120.2.tar.gz", hash = "sha256:9722efc10c27a30a69f5338ddacdb35bc6a64297a4e4ba729bf83af873d5fb3a", size = 1008421, upload-time = "2026-07-28T17:38:26.986Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/63/af/0f5db57b9397a0f3b7fc204cbef143401a7cadaf982330f97f1ce3d39f34/anthropic-0.120.2-py3-none-any.whl", hash = "sha256:0f0bc2b381dc0eb41c8d886b815d79c2041cd2374f83aed36f574b6dc9c579c1", size = 1022851, upload-time = "2026-07-28T17:38:25.466Z" },
]

[[package]]
name = "anyio"
version = "4.14.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "idna" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/61/cc/a381afa6efea9f496eff839d4a6a1aed3bfafc7b3ab4b0d1b243a12573dd/anyio-4.14.2.tar.gz", hash = "sha256:cfa139f3ed1a23ee8f88a145ddb5ac7605b8bbfd8592baacd7ce3d8bb4313c7f", size = 260176, upload-time = "2026-07-12T20:29:07.082Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/da/35/f2287558c17e29fafc8ef3daf819bb9834061cfa43bff8014f7df7f63bdc/anyio-4.14.2-py3-none-any.whl", hash = "sha256:9f505dda5ac9f0c8309b5e8bd445a8c2bf7246f3ce950121e45ea15bc41d1494", size = 125813, upload-time = "2026-07-12T20:29:05.763Z" },
]

[[package]]
name = "asgiref"
version = "3.12.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/e6/26/3b59f2bdae5f640389becb1f673cded775287f5fc4f816309d9ca9a3f93d/asgiref-3.12.1.tar.gz", hash = "sha256:59dcb51c272ad209d59bed5708a64a333083e86017d7fcdd67498eeab7784340", size = 42378, upload-time = "2026-07-14T09:56:18.087Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/c0/1b/54f4ad77cd8a584fa70746c47df988e002cf1ee1eba43364d46f87803647/asgiref-3.12.1-py3-none-any.whl", hash = "sha256:fe386d1c2bff7259ea95929266d12a8cf9a8b5a1c2598402967d8792e7a7c094", size = 25478, upload-time = "2026-07-14T09:56:16.926Z" },
]

[[package]]
name = "ast-serialize"
version = "0.6.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/58/ad/0d70a3a2d6e01968d985415259e8ec7ad3f777903f9b1c1f3c8c44642c60/ast_serialize-0.6.0.tar.gz", hash = "sha256:aadd3ffcf4858c9726bf3515f7b199c7eadbe504f96028e4a87172c0da65a8fe", size = 61489, upload-time = "2026-06-30T20:02:55.555Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/3f/12/3e5f575f156555547c250a8b0d1347517a3a20fc7f4492e9703a69d4f45e/ast_serialize-0.6.0-cp314-cp314t-macosx_10_12_x86_64.whl", hash = "sha256:a7520b672827885bafeae7501f684d14d47d17e5f45256f9df547686cca52264", size = 1177640, upload-time = "2026-06-30T20:02:06.708Z" },
    { url = "https://files.pythonhosted.org/packages/a2/a4/921a9e27951627983b0f368859ea00f8330a551dc0bf4c2fdcb11855a98b/ast_serialize-0.6.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:a14191beec7e0c078d2fc1f6edc0aee88bcd4db9f18e1bc9f8052b559c22dddc", size = 1168111, upload-time = "2026-06-30T20:02:08.366Z" },
    { url = "https://files.pythonhosted.org/packages/00/69/950cf404de7b8782cf95e5c1237e25e2aa46177b287f39f9eeddf481fd6f/ast_serialize-0.6.0-cp314-cp314t-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:32ef62ec34cf6be20ad77d4799556638fbdf187f3ae10698dfb20ef9f2c89516", size = 1227656, upload-time = "2026-06-30T20:02:09.843Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/4c/a8/46f8f6a6479d9d2273980957bb091a506c55f5b95d3c029ee58518a78407/ast_serialize-0.6.0-cp314-cp314t-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:13b7769970a39983b0adf2f38917b1cd3b8946f76df045756c3d741bc689f089", size = 1227706, upload-time = "2026-06-30T20:02:11.367Z" },
    { url = "https://files.pythonhosted.org/packages/b7/b9/9ac415bda0a40e49eab8fea3b2741c19c98bb84d57d62c4cfc6230eb67be/ast_serialize-0.6.0-cp314-cp314t-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:6f7a408601bb3edaefb3bc67a4c01f5235e3253653b6a5729a2ee2382b35341c", size = 1431705, upload-time = "2026-06-30T20:02:12.737Z" },
    { url = "https://files.pythonhosted.org/packages/e5/06/8807115d441444879f7561b5eede5ac18fc80392f11826d61ccf31f503b1/ast_serialize-0.6.0-cp314-cp314t-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:8670bfa51208a2c0c8d138928e40e998fab158f9200d53bb80c088b5b8eda7b8", size = 1249533, upload-time = "2026-06-30T20:02:14.571Z" },
    { url = "https://files.pythonhosted.org/packages/3e/c0/c2ba82ef9618650357d9421a1fdb27ffec862a7f57e8e2de82a3ccd11e12/ast_serialize-0.6.0-cp314-cp314t-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:a4826809eb8597a8cd59fd924b6d7c285b8969a1e0007e2cb652cab62376270f", size = 1252619, upload-time = "2026-06-30T20:02:16.219Z" },
    { url = "https://files.pythonhosted.org/packages/0f/a7/fa31d52dd4102cede29fb9634e98d214129b2783b4f95528c6dc6a8f6587/ast_serialize-0.6.0-cp314-cp314t-manylinux_2_31_riscv64.whl", hash = "sha256:577a6c189068686869f5f1ddc38363f3ae1808a4753b577266f9202071a7bb66", size = 1242983, upload-time = "2026-06-30T20:02:17.813Z" },
    { url = "https://files.pythonhosted.org/packages/b1/20/ddf742b5ad3c4bafd3466f2265037cfd99bc1b9a5ee46a5d58c90d523242/ast_serialize-0.6.0-cp314-cp314t-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:085de7f62dc9cc247eb01e965a362707d1d90b1d89a82c5bf78301a60a3c417b", size = 1296148, upload-time = "2026-06-30T20:02:19.146Z" },
    { url = "https://files.pythonhosted.org/packages/24/cb/9f6f217cce8b3b632c5568b478d195a35e79dce4dbe309438cb89ba6ea4f/ast_serialize-0.6.0-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:9f8a8b78b13173de6a9ec22111d9be674874cd5bdccda04f14ae5ebc2bef403a", size = 1403826, upload-time = "2026-06-30T20:02:20.696Z" },
    { url = "https://files.pythonhosted.org/packages/2d/f8/9d16d4f0107a183924425cc0e7618d8bf76f96b45afa9ff19f924ed1ad57/ast_serialize-0.6.0-cp314-cp314t-musllinux_1_2_armv7l.whl", hash = "sha256:f2ff3baffc3a29c1f15bc9098aa0c09763410262d5e6cef42116f7356c184554", size = 1502943, upload-time = "2026-06-30T20:02:22.034Z" },
    { url = "https://files.pythonhosted.org/packages/80/dd/bbc1c38756350dddf7e24acae1c9482ef42051c267417e019aecc1ed4075/ast_serialize-0.6.0-cp314-cp314t-musllinux_1_2_i686.whl", hash = "sha256:0067b25fce104eaae5b88383de9ab803faeb671831e14ca698b771b356e2600f", size = 1497632, upload-time = "2026-06-30T20:02:23.517Z" },
    { url = "https://files.pythonhosted.org/packages/42/7e/9daffefcf5b97e6bb4c3e0b3c024c1aee9722f23d3cf7cd2ff80d6fb4a40/ast_serialize-0.6.0-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:c617417f9cbb0cb144f6283c3cbe0d2e0f01beaf9f608f662b21191058a626ec", size = 1448858, upload-time = "2026-06-30T20:02:24.889Z" },
    { url = "https://files.pythonhosted.org/packages/e5/1f/f9baaab81a677ea0af7d2458cac2f94ebcc85958f8a3c15ba9d9e5dab653/ast_serialize-0.6.0-cp314-cp314t-win32.whl", hash = "sha256:5337cb256dcea3df9288205213d1601581536526b8f4da44b6974f1180f3252a", size = 1052600, upload-time = "2026-06-30T20:02:26.263Z" },
    { url = "https://files.pythonhosted.org/packages/9e/1f/41b535866519512d8cf6669cb2cff7823b7672bb6279c0333b4ff89d7d9f/ast_serialize-0.6.0-cp314-cp314t-win_amd64.whl", hash = "sha256:2d947e45cafc4b09bd7528917fa84c517654a43de173c79785574b7b3068ac24", size = 1095570, upload-time = "2026-06-30T20:02:27.639Z" },
    { url = "https://files.pythonhosted.org/packages/50/64/e472fe3e3a2d33d874b987e8518aedf24562919e3b6161a4fa1797e89c0f/ast_serialize-0.6.0-cp314-cp314t-win_arm64.whl", hash = "sha256:6e15ec740436e1a0d62de848641abe5f3a2f89a7f94907d534795ac91bbacf14", size = 1067267, upload-time = "2026-06-30T20:02:28.949Z" },
    { url = "https://files.pythonhosted.org/packages/52/19/ac8348ae8711c9b5ae834634f635780cab62a0f5e6f988882e048b89c2ae/ast_serialize-0.6.0-cp39-abi3-macosx_10_12_x86_64.whl", hash = "sha256:093cb8bb91b720d8523580498d031791bb1bbaa048599c3d21085d380e11a596", size = 1185367, upload-time = "2026-06-30T20:02:30.427Z" },
    { url = "https://files.pythonhosted.org/packages/c1/f6/ec7ec652c51db77c2f61d8573338e13e4704303265ccc658cb4031d9f354/ast_serialize-0.6.0-cp39-abi3-macosx_11_0_arm64.whl", hash = "sha256:e61580a69faf47e3689795367ed211f2a10fd741478cc0f36a0f128793360aad", size = 1178657, upload-time = "2026-06-30T20:02:31.964Z" },
    { url = "https://files.pythonhosted.org/packages/6f/02/613a7534a41d0122f37d1e0c64aa8ac78bfb831f8c92f6db057a311abb3c/ast_serialize-0.6.0-cp39-abi3-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:305802f2ce2a7c4e87835078ea85c58b586ddda8095b92fe2ead9364ae19c80a", size = 1238620, upload-time = "2026-06-30T20:02:33.664Z" },
    { url = "https://files.pythonhosted.org/packages/4d/21/087957bba486242afc52f49b2d9e21c9dad00289356cf9efe67084015a9d/ast_serialize-0.6.0-cp39-abi3-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:c7b8b8f0c42f752ea00b2b7d7c090b3f80d9c1c5c75cadf16423790a0cc74081", size = 1236075, upload-time = "2026-06-30T20:02:34.936Z" },
    { url = "https://files.pythonhosted.org/packages/82/04/78128bbb170071c2c72a210a181f1c00e11cc1cec60a8beef747b07f9201/ast_serialize-0.6.0-cp39-abi3-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:cd5b91b9e6f2356ace3a556963b0cd783b395fbbb0bb17b4defc283415466e77", size = 1441348, upload-time = "2026-06-30T20:02:36.245Z" },
    { url = "https://files.pythonhosted.org/packages/64/64/62fb99d6faf199b4c3e5b08a07136e9a0d7664bb249c6de3670e5b63e9b6/ast_serialize-0.6.0-cp39-abi3-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:4d6ef91590258ada18909b9caea344dac4de2013906b035473cd674a43f4b790", size = 1258580, uploaded-time = "2026-06-30T20:02:37.53Z" },
    { url = "https://files.pythonhosted.org/packages/ca/87/b4d6c38e0ccd5e85dc54cecdf933a152c60b28fe5d993a6d8a72fa6d5896/ast_serialize-0.6.0-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:dcbed41e9386059fc0261d602445ede0976c2ecec2939688bcbcb9ed0b6f28b7", size = 1261693, upload-time = "2026-06-30T20:02:39.123Z" },
    { url = "https://files.pythonhosted.org/packages/0e/4b/3676ca2191f39bafb75f93f99b2f429ec464586158fece2165f3572805dc/ast_serialize-0.6.0-cp39-abi3-manylinux_2_31_riscv64.whl", hash = "sha256:cdc4e6f930b9090c2f92c9036ad12ffb8e6e44d4a5ba06f1458a05d60f203f7b", size = 1252517, upload-time = "2026-06-30T20:02:40.511Z" },
    { url = "https://files.pythonhosted.org/packages/f3/58/494ef8c4b4acb2f4a265ac934caf45f792a08fe27d6b853de35ad991941a/ast_serialize-0.6.0-cp39-abi3-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:897ac47b5637be41c0c07061c8a912fafa967ef1dc73fa115e4bfa70882a093b", size = 1304843, upload-time = "2026-06-30T20:02:41.961Z" },
    { url = "https://files.pythonhosted.org/packages/b1/f2/13736d920ab3d49bbee80ef1a277dd7b7aaf3b3545efd9d2a8114fe05525/ast_serialize-0.6.0-cp39-abi3-musllinux_1_2_aarch64.whl", hash = "sha256:c4af9a1386166e40ed01464991806f89038a2d89782576c7774876fa77034e32", size = 1413698, upload-time = "2026-06-30T20:02:44.179Z" },
    { url = "https://files.pythonhosted.org/packages/a8/5a/e046f3899e2acba4677d7427b76431443a1aa1a0e583dfb05b55b69d55cf/ast_serialize-0.6.0-cp39-abi3-musllinux_1_2_armv7l.whl", hash = "sha256:c901adbd750029b9ac4ad3d6aa56853e0ad4875119fbf52b7b8298afc223828b", size = 1512209, upload-time = "2026-06-30T20:02:45.584Z" },
    { url = "https://files.pythonhosted.org/packages/cc/c7/e42aaca7bb2d22a7c06d5a8c7930086c5a334e93d716e6fa5e6647a4515f/ast_serialize-0.6.0-cp39-abi3-musllinux_1_2_i686.whl", hash = "sha256:3ae22a366b752ab4496191525b78b097b5b72d531752e3c1dd7e383a8f2c8a1a", size = 1508464, upload-time = "2026-06-30T20:02:46.942Z" },
    { url = "https://files.pythonhosted.org/packages/95/93/5524a3dc6c3f593de3228ed9cbef73afa047625b7000ec21b7f58e6eb4d4/ast_serialize-0.6.0-cp39-abi3-musllinux_1_2_x86_64.whl", hash = "sha256:4ed29121da8b3fdc291002801a1de0f76248fa07dce89157a5f277842cf6126e", size = 1457164, upload-time = "2026-06-30T20:02:48.294Z" },
    { url = "https://files.pythonhosted.org/packages/4f/c0/36a6ffb4d653cf621427b4c4928671f53ad800c453474de2b82564a44ad9/ast_serialize-0.6.0-cp39-abi3-pyemscripten_2026_0_wasm32.whl", hash = "sha256:b1dac4e09d341c1300ba69cdcbe62867b32a8c75d90db9bf4d083bec3b039f0b", size = 863014, upload-time = "2026-06-30T20:02:49.742Z" },
    { url = "https://files.pythonhosted.org/packages/09/c7/7d5ad8b49e1278e1c2a1e0274bd7850560b3f09313aa00c13bc8d5544792/ast_serialize-0.6.0-cp39-abi3-win32.whl", hash = "sha256:82c312a7844d2fdeb4d5c48bd3d215bf940dafd4704e1a9bcf252a99010a99b1", size = 1063165, upload-time = "2026-06-30T20:02:50.98Z" },
    { url = "https://files.pythonhosted.org/packages/47/ae/6710c14ecb276031cf10249f6adf5a59e2d3fdb3b5183bd59f70524067ee/ast_serialize-0.6.0-cp39-abi3-win_amd64.whl", hash = "sha256:113b58346f9ceb664352032770caca817d4a3c86f611c6088e6ef65ddaa70f0e", size = 1101444, upload-time = "2026-06-30T20:02:52.554Z" },
    { url = "https://files.pythonhosted.org/packages/66/40/c53deb2cd0c9b0fb636d24d9f40924cf2e65028e6b20b10cd5c1eeb2c730/ast_serialize-0.6.0-cp39-abi3-win_arm64.whl", hash = "sha256:ccd132fe8db56f61fe743b1f644d01b8d65b83248a8da506f3132bda86d6ed5e", size = 1072965, upload-time = "2026-06-30T20:02:54.097Z" },
]

[[package]]
name = "asyncpg"
version = "0.31.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/fe/cc/d18065ce2380d80b1bcce927c24a2642efd38918e33fd724bc4bca904877/asyncpg-0.31.0.tar.gz", hash = "sha256:c989386c83940bfbd787180f2b1519415e2d3d6277a70d9d0f0145ac73500735", size = 993667, upload-time = "2025-11-24T23:27:00.812Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/2a/a6/59d0a146e61d20e18db7396583242e32e0f120693b67a8de43f1557033e2/asyncpg-0.31.0-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:b44c31e1efc1c15188ef183f287c728e2046abb1d26af4d20858215d50d91fad", size = 662042, upload-time = "2025-11-24T23:25:49.578Z" },
    { url = "https://files.pythonhosted.org/packages/36/01/ffaa189dcb63a2471720615e60185c3f6327716fdc0fc04334436fbb7c65/asyncpg-0.31.0-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:0c89ccf741c067614c9b5fc7f1fc6f3b61ab05ae4aaa966e6fd6b93097c7d20d", size = 638504, upload-time = "2025-11-24T23:25:51.501Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/9f/62/3f699ba45d8bd24c5d65392190d19656d74ff0185f42e19d0bbd973bb371/asyncpg-0.31.0-cp312-cp312-manylinux_2_28_aarch64.whl", hash = "sha256:12b3b2e39dc5470abd5e98c8d3373e4b1d1234d9fbdedf538798b2c13c64460a", size = 3426241, upload-time = "2025-11-24T23:25:53.278Z" },
    { url = "https://files.pythonhosted.org/packages/8c/d1/a867c2150f9c6e7af6462637f613ba67f78a314b00db220cd26ff559d532/asyncpg-0.31.0-cp312-cp312-manylinux_2_28_x86_64.whl", hash = "sha256:aad7a33913fb8bcb5454313377cc330fbb19a0cd5faa7272407d8a0c4257b671", size = 3520321, upload-time = "2025-11-24T23:25:54.982Z" },
    { url = "https://files.pythonhosted.org/packages/7a/1a/cce4c3f246805ecd285a3591222a2611141f1669d002163abef999b60f98/asyncpg-0.31.0-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:3df118d94f46d85b2e434fd62c84cb66d5834d5a890725fe625f498e72e4d5ec", size = 3316685, upload-time = "2025-11-24T23:25:57.43Z" },
    { url = "https://files.pythonhosted.org/packages/40/ae/0fc961179e78cc579e138fad6eb580448ecae64908f95b8cb8ee2f241f67/asyncpg-0.31.0-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:bd5b6efff3c17c3202d4b37189969acf8927438a238c6257f66be3c426beba20", size = 3471858, upload-time = "2025-11-24T23:25:59.636Z" },
    { url = "https://files.pythonhosted.org/packages/52/b2/b20e09670be031afa4cbfabd645caece7f85ec62d69c312239de568e058e/asyncpg-0.31.0-cp312-cp312-win32.whl", hash = "sha256:027eaa61361ec735926566f995d959ade4796f6a49d3bde17e5134b9964f9ba8", size = 527852, upload-time = "2025-11-24T23:26:01.084Z" },
    { url = "https://files.pythonhosted.org/packages/b5/f0/f2ed1de154e15b107dc692262395b3c17fc34eafe2a78fc2115931561730/asyncpg-0.31.0-cp312-cp312-win_amd64.whl", hash = "sha256:72d6bdcbc93d608a1158f17932de2321f68b1a967a13e014998db87a72ed3186", size = 597175, upload-time = "2025-11-24T23:26:02.564Z" },
    { url = "https://files.pythonhosted.org/packages/95/11/97b5c2af72a5d0b9bc3fa30cd4b9ce22284a9a943a150fdc768763caf035/asyncpg-0.31.0-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:c204fab1b91e08b0f47e90a75d1b3c62174dab21f670ad6c5d0f243a228f015b", size = 661111, upload-time = "2025-11-24T23:26:04.467Z" },
    { url = "https://files.pythonhosted.org/packages/1b/71/157d611c791a5e2d0423f09f027bd499935f0906e0c2a416ce712ba51ef3/asyncpg-0.31.0-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:54a64f91839ba59008eccf7aad2e93d6e3de688d796f35803235ea1c4898ae1e", size = 636928, upload-time = "2025-11-24T23:26:05.944Z" },
    { url = "https://files.pythonhosted.org/packages/2e/fc/9e3486fb2bbe69d4a867c0b76d68542650a7ff1574ca40e84c3111bb0c6e/asyncpg-0.31.0-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:c0e0822b1038dc7253b337b0f3f676cadc4ac31b126c5d42691c39691962e403", size = 3424067, upload-time = "2025-11-24T23:26:07.957Z" },
    { url = "https://files.pythonhosted.org/packages/12/c6/8c9d076f73f07f995013c791e018a1cd5f31823c2a3187fc8581706aa00f/asyncpg-0.31.0-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:bef056aa502ee34204c161c72ca1f3c274917596877f825968368b2c33f585f4", size = 3518156, upload-time = "2025-11-24T23:26:09.591Z" },
    { url = "https://files.pythonhosted.org/packages/ae/3b/60683a0baf50fbc546499cfb53132cb6835b92b529a05f6a81471ab60d0c/asyncpg-0.31.0-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:0bfbcc5b7ffcd9b75ab1558f00db2ae07db9c80637ad1b2469c43df79d7a5ae2", size = 3319636, upload-time = "2025-11-24T23:26:11.168Z" },
    { url = "https://files.pythonhosted.org/packages/50/dc/8487df0f69bd398a61e1792b3cba0e47477f214eff085ba0efa7eac9ce87/asyncpg-0.31.0-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:22bc525ebbdc24d1261ecbf6f504998244d4e3be1721784b5f64664d61fbe602", size = 3472079, upload-time = "2025-11-24T23:26:13.164Z" },
    { url = "https://files.pythonhosted.org/packages/13/a1/c5bbeeb8531c05c89135cb8b28575ac2fac618bcb60119ee9696c3faf71c/asyncpg-0.31.0-cp313-cp313-win32.whl", hash = "sha256:f890de5e1e4f7e14023619399a471ce4b71f5418cd67a51853b9910fdfa73696", size = 527606, upload-time = "2025-11-24T23:26:14.78Z" },
    { url = "https://files.pythonhosted.org/packages/91/66/b25ccb84a246b470eb943b0107c07edcae51804912b824054b3413995a10/asyncpg-0.31.0-cp313-cp313-win_amd64.whl", hash = "sha256:dc5f2fa9916f292e5c5c8b2ac2813763bcd7f58e130055b4ad8a0531314201ab", size = 596569, upload-time = "2025-11-24T23:26:16.189Z" },
    { url = "https://files.pythonhosted.org/packages/3c/36/e9450d62e84a13aea6580c83a47a437f26c7ca6fa0f0fd40b6670793ea30/asyncpg-0.31.0-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:f6b56b91bb0ffc328c4e3ed113136cddd9deefdf5f79ab448598b9772831df44", size = 660867, upload-time = "2025-11-24T23:26:17.631Z" },
    { url = "https://files.pythonhosted.org/packages/82/4b/1d0a2b33b3102d210439338e1beea616a6122267c0df459ff0265cd5807a/asyncpg-0.31.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:334dec28cf20d7f5bb9e45b39546ddf247f8042a690bff9b9573d00086e69cb5", size = 638349, upload-time = "2025-11-24T23:26:19.689Z" },
    { url = "https://files.pythonhosted.org/packages/41/aa/e7f7ac9a7974f08eff9183e392b2d62516f90412686532d27e196c0f0eeb/asyncpg-0.31.0-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:98cc158c53f46de7bb677fd20c417e264fc02b36d901cc2a43bd6cb0dc6dbfd2", size = 3410428, upload-time = "2025-11-24T23:26:21.275Z" },
    { url = "https://files.pythonhosted.org/packages/6f/de/bf1b60de3dede5c2731e6788617a512bc0ebd9693eac297ee74086f101d7/asyncpg-0.31.0-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9322b563e2661a52e3cdbc93eed3be7748b289f792e0011cb2720d278b366ce2", size = 3471678, upload-time = "2025-11-24T23:26:23.627Z" },
    { url = "https://files.pythonhosted.org/packages/46/78/fc3ade003e22d8bd53aaf8f75f4be48f0b460fa73738f0391b9c856a9147/asyncpg-0.31.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:19857a358fc811d82227449b7ca40afb46e75b33eb8897240c3839dd8b744218", size = 3313505, upload-time = "2025-11-24T23:26:25.235Z" },
    { url = "https://files.pythonhosted.org/packages/bf/e9/73eb8a6789e927816f4705291be21f2225687bfa97321e40cd23055e903a/asyncpg-0.31.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:ba5f8886e850882ff2c2ace5732300e99193823e8107e2c53ef01c1ebfa1e85d", size = 3434744, upload-time = "2025-11-24T23:26:26.944Z" },
    { url = "https://files.pythonhosted.org/packages/08/4b/f10b880534413c65c5b5862f79b8e81553a8f364e5238832ad4c0af71b7f/asyncpg-0.31.0-cp314-cp314-win32.whl", hash = "sha256:cea3a0b2a14f95834cee29432e4ddc399b95700eb1d51bbc5bfee8f31fa07b2b", size = 532251, upload-time = "2025-11-24T23:26:28.404Z" },
    { url = "https://files.pythonhosted.org/packages/d3/2d/7aa40750b7a19efa5d66e67fc06008ca0f27ba1bd082e457ad82f59aba49/asyncpg-0.31.0-cp314-cp314-win_amd64.whl", hash = "sha256:04d19392716af6b029411a0264d92093b6e5e8285ae97a39957b9a9c14ea72be", size = 604901, upload-time = "2025-11-24T23:26:30.34Z" },
    { url = "https://files.pythonhosted.org/packages/ce/fe/b9dfe349b83b9dee28cc42360d2c86b2cdce4cb551a2c2d27e156bcac84d/asyncpg-0.31.0-cp314-cp314t-macosx_10_15_x86_64.whl", hash = "sha256:bdb957706da132e982cc6856bb2f7b740603472b54c3ebc77fe60ea3e57e1bd2", size = 702280, upload-time = "2025-11-24T23:26:32Z" },
    { url = "https://files.pythonhosted.org/packages/6a/81/e6be6e37e560bd91e6c23ea8a6138a04fd057b08cf63d3c5055c98e81c1d/asyncpg-0.31.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:6d11b198111a72f47154fa03b85799f9be63701e068b43f84ac25da0bda9cb31", size = 682931, upload-time = "2025-11-24T23:26:33.572Z" },
    { url = "https://files.pythonhosted.org/packages/a6/45/6009040da85a1648dd5bc75b3b0a062081c483e75a1a29041ae63a0bf0dc/asyncpg-0.31.0-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:18c83b03bc0d1b23e6230f5bf8d4f217dc9bc08644ce0502a9d91dc9e634a9c7", size = 3581608, upload-time = "2025-11-24T23:26:35.638Z" },
    { url = "https://files.pythonhosted.org/packages/7e/06/2e3d4d7608b0b2b3adbee0d0bd6a2d29ca0fc4d8a78f8277df04e2d1fd7b/asyncpg-0.31.0-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:e009abc333464ff18b8f6fd146addffd9aaf63e79aa3bb40ab7a4c332d0c5e9e", size = 3498738, upload-time = "2025-11-24T23:26:37.275Z" },
    { url = "https://files.pythonhosted.org/packages/7d/aa/7d75ede780033141c51d83577ea23236ba7d3a23593929b32b49db8ed36e/asyncpg-0.31.0-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:3b1fbcb0e396a5ca435a8826a87e5c2c2cc0c8c68eb6fadf82168056b0e53a8c", size = 3401026, upload-time = "2025-11-24T23:26:39.423Z" },
    { url = "https://files.pythonhosted.org/packages/ba/7a/15e37d45e7f7c94facc1e9148c0e455e8f33c08f0b8a0b1deb2c5171771b/asyncpg-0.31.0-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:8df714dba348efcc162d2adf02d213e5fab1bd9f557e1305633e851a61814a7a", size = 3429426, upload-time = "2025-11-24T23:26:41.032Z" },
    { url = "https://files.pythonhosted.org/packages/13/d5/71437c5f6ae5f307828710efbe62163974e71237d5d46ebd2869ea052d10/asyncpg-0.31.0-cp314-cp314t-win32.whl", hash = "sha256:1b41f1afb1033f2b44f3234993b15096ddc9cd71b21a42dbd87fc6a57b43d65d", size = 614495, upload-time = "2025-11-24T23:26:42.659Z" },
    { url = "https://files.pythonhosted.org/packages/3c/d7/8fb3044eaef08a310acfe23dae9a8e2e07d305edc29a53497e52bc76eca7/asyncpg-0.31.0-cp314-cp314t-win_amd64.whl", hash = "sha256:bd4107bb7cdd0e9e65fae66a62afd3a249663b844fa34d479f6d5b3bef9c04c3", size = 706062, upload-time = "2025-11-24T23:26:44.086Z" },
]

[[package]]
name = "attrs"
version = "26.1.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/9a/8e/82a0fe20a541c03148528be8cac2408564a6c9a0cc7e9171802bc1d26985/attrs-26.1.0.tar.gz", hash = "sha256:d03ceb89cb322a8fd706d4fb91940737b6642aa36998fe130a9bc96c985eff32", size = 952055, upload-time = "2026-03-19T14:22:25.026Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/64/b4/17d4b0b2a2dc85a6df63d1157e028ed19f90d4cd97c36717afef2bc2f395/attrs-26.1.0-py3-none-any.whl", hash = "sha256:c647aa4a12dfbad9333ca4e71fe62ddc36f4e63b2d260a37a8b83d2f043ac309", size = 67548, upload-time = "2026-03-19T14:22:23.645Z" },
]

[[package]]
name = "backoff"
version = "2.2.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/47/d7/5bbeb12c44d7c4f2fb5b56abce497eb5ed9f34d85701de869acedd602619/backoff-2.2.1.tar.gz", hash = "sha256:03f829f5bb1923180821643f8753b0502c3b682293992485b0eef2807afa5cba", size = 17001, upload-time = "2022-10-05T19:19:32.061Z" }
```

```
wheels = [
    { url = "https://files.pythonhosted.org/packages/df/73/b6e24bd22e6720ca8ee9a85a0c4a2971af8497d8f3193fa05390cbd46e09/backoff-2.2.1-py3-none-any.whl", hash = "sha256:63579f9a0628e06278f7e47b7d7d5b6ce20dc65c5e96a6f3ca99a6adca0396e8", size = 15148, upload-time = "2022-10-05T19:19:30.546Z" },
]

[[package]]
name = "banks"
version = "2.4.5"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "deprecated" },
    { name = "filetype" },
    { name = "griffe" },
    { name = "jinja2" },
    { name = "platformdirs" },
    { name = "pydantic" },
]
sdist = { url = "https://files.pythonhosted.org/packages/a1/b5/4784ee9518b97f9f69c714a4303f9a6186a7e4ff2349f89e24767e9754d9/banks-2.4.5.tar.gz", hash = "sha256:ff575732fc67d5493a73c21e0d7268bc49e86fff02b0b8735e8efb9fcb9af3a4", size = 190822, upload-time = "2026-07-07T08:14:12.055Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/50/b0/4d34cb2fe538aeb6d4b0723b3339cb932120db084a5c9a3c6966a26bbe1b/banks-2.4.5-py3-none-any.whl", hash = "sha256:ac2e0091b4c79379d4773c9d04a138a0d937ee27c5803bf0142acc6d6769eea1", size = 36145, upload-time = "2026-07-07T08:14:10.974Z" },
]

[[package]]
name = "bcrypt"
version = "5.0.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/d4/36/3329e2518d70ad8e2e5817d5a4cac6bba05a47767ec416c7d020a965f408/bcrypt-5.0.0.tar.gz", hash = "sha256:f748f7c2d6fd375cc93d3fba7ef4a9e3a092421b8dbf34d8d4dc06be9492dfdd", size = 25386, upload-time = "2025-09-25T19:50:47.829Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/13/85/3e65e01985fddf25b64ca67275bb5bdb4040bd1a53b66d355c6c37c8a680/bcrypt-5.0.0-cp313-cp313t-macosx_10_12_universal2.whl", hash = "sha256:f3c08197f3039bec79cee59a606d62b96b16669cff3949f21e74796b6e3cd2be", size = 481806, upload-time = "2025-09-25T19:49:05.102Z" },
    { url = "https://files.pythonhosted.org/packages/44/dc/01eb79f12b177017a726cbf78330eb0eb442fae0e7b3dfd84ea2849552f3/bcrypt-5.0.0-cp313-cp313t-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:200af71bc25f22006f4069060c88ed36f8aa4ff7f53e67ff04d2ab3f1e79a5b2", size = 268626, upload-time = "2025-09-25T19:49:06.723Z" },
    { url = "https://files.pythonhosted.org/packages/8c/cf/e82388ad5959c40d6afd94fb4743cc077129d45b952d46bdc3180310e2df/bcrypt-5.0.0-cp313-cp313t-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:baade0a5657654c2984468efb7d6c110db87ea63ef5a4b54732e7e337253e44f", size = 271853, upload-time = "2025-09-25T19:49:08.028Z" },
    { url = "https://files.pythonhosted.org/packages/ec/86/7134b9dae7cf0efa85671651341f6afa695857fae172615e960fb6a466fa/bcrypt-5.0.0-cp313-cp313t-manylinux_2_28_aarch64.whl", hash = "sha256:c58b56cdfb03202b3bcc9fd8daee8e8e9b6d7e3163aa97c631dfcfcc24d36c86", size = 269793, upload-time = "2025-09-25T19:49:09.727Z" },
    { url = "https://files.pythonhosted.org/packages/cc/82/6296688ac1b9e503d034e7d0614d56e80c5d1a08402ff856a4549cb59207/bcrypt-5.0.0-cp313-cp313t-manylinux_2_28_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:4bfd2a34de661f34d0bda43c3e4e79df586e4716ef401fe31ea39d69d581ef23", size = 289930, upload-time = "2025-09-25T19:49:11.204Z" },
    { url = "https://files.pythonhosted.org/packages/d1/18/884a44aa47f2a3b88dd09bc05a1e40b57878ecd111d17e5bba6f09f8bb77/bcrypt-5.0.0-cp313-cp313t-manylinux_2_28_x86_64.whl", hash = "sha256:ed2e1365e31fc73f1825fa830f1c8f8917ca1b3ca6185773b349c20fd606cec2", size = 272194, upload-time = "2025-09-25T19:49:12.524Z" },
    { url = "https://files.pythonhosted.org/packages/0e/8f/371a3ab33c6982070b674f1788e05b656cfbf5685894acbfef0c65483a59/bcrypt-5.0.0-cp313-cp313t-manylinux_2_34_aarch64.whl", hash = "sha256:83e787d7a84dbbfba6f250dd7a5efd689e935f03dd83b0f919d39349e1f23f83", size = 269381, upload-time = "2025-09-25T19:49:14.308Z" },
    { url = "https://files.pythonhosted.org/packages/b1/34/7e4e6abb7a8778db6422e88b1f06eb07c47682313997ee8a8f9352e5a6f1/bcrypt-5.0.0-cp313-cp313t-manylinux_2_34_x86_64.whl", hash = "sha256:137c5156524328a24b9fac1cb5db0ba618bc97d11970b39184c1d87dc4bf1746", size = 271750, upload-time = "2025-09-25T19:49:15.584Z" },
    { url = "https://files.pythonhosted.org/packages/c0/1b/54f416be2499bd72123c70d98d36c6cd61a4e33d9b89562c22481c81bb30/bcrypt-5.0.0-cp313-cp313t-musllinux_1_1_aarch64.whl", hash = "sha256:38cac74101777a6a7d3b3e3cfefa57089b5ada650dce2baf0cbdd9d65db22a9e", size = 303757, upload-time = "2025-09-25T19:49:17.244Z" },
    { url = "https://files.pythonhosted.org/packages/13/62/062c24c7bcf9d2826a1a843d0d605c65a755bc98002923d01fd61270705a/bcrypt-5.0.0-cp313-cp313t-musllinux_1_1_x86_64.whl", hash = "sha256:d8d65b564ec849643d9f7ea05c6d9f0cd7ca23bdd4ac0c2dbef1104ab504543d", size = 306740, upload-time = "2025-09-25T19:49:18.693Z" },
    { url = "https://files.pythonhosted.org/packages/d5/c8/1fdbfc8c0f20875b6b4020f3c7dc447b8de60aa0be5faaf009d24242aec9/bcrypt-5.0.0-cp313-cp313t-musllinux_1_2_aarch64.whl", hash = "sha256:741449132f64b3524e95cd30e5cd3343006ce146088f074f31ab26b94e6c75ba", size = 334197, upload-time = "2025-09-25T19:49:20.523Z" },
    { url = "https://files.pythonhosted.org/packages/a6/c1/8b84545382d75bef226fbc6588af0f7b7d095f7cd6a670b42a86243183cd/bcrypt-5.0.0-cp313-cp313t-musllinux_1_2_x86_64.whl", hash = "sha256:212139484ab3207b1f0c00633d3be92fef3c5f0af17cad155679d03ff2ee1e41", size = 352974, upload-time = "2025-09-25T19:49:22.254Z" },
    { url = "https://files.pythonhosted.org/packages/10/a6/ffb49d4254ed085e62e3e5dd05982b4393e32fe1e49bb1130186617c29cd/bcrypt-5.0.0-cp313-cp313t-win32.whl", hash = "sha256:9d52ed507c2488eddd6a95bccee4e808d3234fa78dd370e24bac65a21212b861", size = 148498, upload-time = "2025-09-25T19:49:24.134Z" },
    { url = "https://files.pythonhosted.org/packages/48/a9/259559edc85258b6d5fc5471a62a3299a6aa37a6611a169756bf4689323c/bcrypt-5.0.0-cp313-cp313t-win_amd64.whl", hash = "sha256:f6984a24db30548fd39a44360532898c33528b74aedf81c26cf29c51ee47057e", size = 145853, upload-time = "2025-09-25T19:49:25.702Z" },
    { url = "https://files.pythonhosted.org/packages/2d/df/9714173403c7e8b245acf8e4be8876aac64a209d1b392af457c79e60492e/bcrypt-5.0.0-cp313-cp313t-win_arm64.whl", hash = "sha256:9fffdb387abe6aa775af36ef16f55e318dcda4194ddbf82007a6f21da29de8f5", size = 139626, upload-time = "2025-09-25T19:49:26.928Z" },
    { url = "https://files.pythonhosted.org/packages/f8/14/c18006f91816606a4abe294ccc5d1e6f0e42304df5a33710e9e8e95416e1/bcrypt-5.0.0-cp314-cp314t-macosx_10_12_universal2.whl", hash = "sha256:4870a52610537037adb382444fefd3706d96d663ac44cbb2f37e3919dca3d7ef", size = 481862, upload-time = "2025-09-25T19:49:28.365Z" },
    { url = "https://files.pythonhosted.org/packages/67/49/dd074d831f00e589537e07a0725cf0e220d1f0d5d8e85ad5bbff251c45aa/bcrypt-5.0.0-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:48f753100931605686f74e27a7b49238122aa761a9aefe9373265b8b7aa43ea4", size = 268544, upload-time = "2025-09-25T19:49:30.39Z" },
    { url = "https://files.pythonhosted.org/packages/f5/91/50ccba088b8c474545b034a1424d05195d9fcbaaf802ab8bfe2be5a4e0d7/bcrypt-5.0.0-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:f70aadb7a809305226daedf75d90379c397b094755a710d7014b8b117df1ebbf", size = 271787, upload-time = "2025-09-25T19:49:32.144Z" },
    { url = "https://files.pythonhosted.org/packages/aa/e7/d7dba133e02abcda3b52087a7eea8c0d4f64d3e593b4fffc10c31b7061f3/bcrypt-5.0.0-cp314-cp314t-manylinux_2_28_aarch64.whl", hash = "sha256:744d3c6b164caa658adcb72cb8cc9ad9b4b75c7db507ab4bc2480474a51989da", size = 269753, upload-time = "2025-09-25T19:49:33.885Z" },
    { url = "https://files.pythonhosted.org/packages/33/fc/5b145673c4b8d01018307b5c2c1fc87a6f5a436f0ad56607aee389de8ee3/bcrypt-5.0.0-cp314-cp314t-manylinux_2_28_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:a28bc05039bdf3289d757f49d616ab3efe8cf40d8e8001ccdd621cd4f98f4fc9", size = 289587, upload-time = "2025-09-25T19:49:35.144Z" },
    { url = "https://files.pythonhosted.org/packages/27/d7/1ff22703ec6d4f90e62f1a5654b8867ef96bafb8e8102c2288333e1a6ca6/bcrypt-5.0.0-cp314-cp314t-manylinux_2_28_x86_64.whl", hash = "sha256:7f277a4b3390ab4bebe597800a90da0edae882c6196d3038a73adf446c4f969f", size = 272178, upload-time = "2025-09-25T19:49:36.793Z" },
    { url = "https://files.pythonhosted.org/packages/c8/88/815b6d558a1e4d40ece04a2f84865b0fef233513bd85fd0e40c294272d62/bcrypt-5.0.0-cp314-cp314t-manylinux_2_34_aarch64.whl", hash = "sha256:79cfa161eda8d2ddf29acad370356b47f02387153b11d46042e93a0a95127493", size = 269295, upload-time = "2025-09-25T19:49:38.164Z" },
    { url = "https://files.pythonhosted.org/packages/51/8c/e0db387c79ab4931fc89827d37608c31cc57b6edc08ccd2386139028dc0d/bcrypt-5.0.0-cp314-cp314t-manylinux_2_34_x86_64.whl", hash = "sha256:a5393eae5722bcef046a990b84dff02b954904c36a194f6cfc817d7dca6c6f0b", size = 271700, upload-time = "2025-09-25T19:49:39.917Z" },
    { url = "https://files.pythonhosted.org/packages/06/83/1570eddd150f572dbe9fc00f6203a89fc7d4226821f67328a85c330f239/bcrypt-5.0.0-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:7f4c94dec1b5ab5d522750cb059bb9409ea8872d4494fd152b53cca99f1ddd8c", size = 334034, upload-time = "2025-09-25T19:49:41.227Z" },
    { url = "https://files.pythonhosted.org/packages/c9/f2/ea64e51a65e56ae7a8a4ec236c2bfbdd4b23008abd50ac33fbb2d1d15424/bcrypt-5.0.0-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:0cae4cb350934dfd74c020525eeae0a5f79257e8a201c0c176f4b84fdbf2a4b4", size = 352766, upload-time = "2025-09-25T19:49:43.08Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/d7/d4/1a388d21ee66876f27d1a1f41287897d0c0f1712ef97d395d708ba93004c/bcrypt-5.0.0-cp314-cp314t-win32.whl", hash = "sha256:b17366316c654e1ad0306a6858e189fc835eca39f7eb2cafd6aaca8ce0c40a2e", size = 152449, upload-time = "2025-09-25T19:49:44.971Z" },
    { url = "https://files.pythonhosted.org/packages/3f/61/3291c2243ae0229e5bca5d19f4032cecad5dfb05a2557169d3a69dc0ba91/bcrypt-5.0.0-cp314-cp314t-win_amd64.whl", hash = "sha256:92864f54fb48b4c718fc92a32825d0e42265a627f956bc0361fe869f1adc3e7d", size = 149310, upload-time = "2025-09-25T19:49:46.162Z" },
    { url = "https://files.pythonhosted.org/packages/3e/89/4b01c52ae0c1a681d4021e5dd3e45b111a8fb47254a274fa9a378d8d834b/bcrypt-5.0.0-cp314-cp314t-win_arm64.whl", hash = "sha256:dd19cf5184a90c873009244586396a6a884d591a5323f0e8a5922560718d4993", size = 143761, upload-time = "2025-09-25T19:49:47.345Z" },
    { url = "https://files.pythonhosted.org/packages/84/29/6237f151fbfe295fe3e074ecc6d44228faa1e842a81f6d34a02937ee1736/bcrypt-5.0.0-cp38-abi3-macosx_10_12_universal2.whl", hash = "sha256:fc746432b951e92b58317af8e0ca746efe93e66555f1b40888865ef5bf56446b", size = 494553, upload-time = "2025-09-25T19:49:49.006Z" },
    { url = "https://files.pythonhosted.org/packages/45/b6/4c1205dde5e464ea3bd88e8742e19f899c16fa8916fb8510a851fae985b5/bcrypt-5.0.0-cp38-abi3-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:c2388ca94ffee269b6038d48747f4ce8df0ffbea43f31abfa18ac72f0218effb", size = 275009, upload-time = "2025-09-25T19:49:50.581Z" },
    { url = "https://files.pythonhosted.org/packages/3b/71/427945e6ead72ccffe77894b2655b695ccf14ae1866cd977e185d606dd2f/bcrypt-5.0.0-cp38-abi3-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:560ddb6ec730386e7b3b26b8b4c88197aaed924430e7b74666a586ac997249ef", size = 278029, upload-time = "2025-09-25T19:49:52.533Z" },
    { url = "https://files.pythonhosted.org/packages/17/72/c344825e3b83c5389a369c8a8e58ffe1480b8a699f46c127c34580c4666b/bcrypt-5.0.0-cp38-abi3-manylinux_2_28_aarch64.whl", hash = "sha256:d79e5c65dcc9af213594d6f7f1fa2c98ad3fc10431e7aa53c176b441943efbdd", size = 275907, upload-time = "2025-09-25T19:49:54.709Z" },
    { url = "https://files.pythonhosted.org/packages/0b/7e/d4e47d2df1641a36d1212e5c0514f5291e1a956a7749f1e595c07a972038/bcrypt-5.0.0-cp38-abi3-manylinux_2_28_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:2b732e7d388fa22d48920baa267ba5d97cca38070b69c0e2d37087b381c681fd", size = 296500, upload-time = "2025-09-25T19:49:56.013Z" },
    { url = "https://files.pythonhosted.org/packages/0f/c3/0ae57a68be2039287ec28bc463b82e4b8dc23f9d12c0be331f4782e19108/bcrypt-5.0.0-cp38-abi3-manylinux_2_28_x86_64.whl", hash = "sha256:0c8e093ea2532601a6f686edbc2c6b2ec24131ff5c52f7610dd64fa4553b5464", size = 278412, upload-time = "2025-09-25T19:49:57.356Z" },
    { url = "https://files.pythonhosted.org/packages/45/2b/77424511adb11e6a99e3a00dcc7745034bee89036ad7d7e255a7e47be7d8/bcrypt-5.0.0-cp38-abi3-manylinux_2_34_aarch64.whl", hash = "sha256:5b1589f4839a0899c146e8892efe320c0fa096568abd9b95593efac50a87cb75", size = 275486, upload-time = "2025-09-25T19:49:59.116Z" },
    { url = "https://files.pythonhosted.org/packages/43/0a/405c753f6158e0f3f14b00b462d8bca31296f7ecfc8fc8bc7919c0c7d73a/bcrypt-5.0.0-cp38-abi3-manylinux_2_34_x86_64.whl", hash = "sha256:89042e61b5e808b67daf24a434d89bab164d4de1746b37a8d173b6b14f3db9ff", size = 277940, upload-time = "2025-09-25T19:50:00.869Z" },
    { url = "https://files.pythonhosted.org/packages/62/83/b3efc285d4aadc1fa83db385ec64dcfa1707e890eb42f03b127d66ac1b7b/bcrypt-5.0.0-cp38-abi3-musllinux_1_1_aarch64.whl", hash = "sha256:e3cf5b2560c7b5a142286f69bde914494b6d8f901aaa71e453078388a50881c4", size = 310776, upload-time = "2025-09-25T19:50:02.393Z" },
    { url = "https://files.pythonhosted.org/packages/95/7d/47ee337dacecde6d234890fe929936cb03ebc4c3a7460854bbd9c97780b8/bcrypt-5.0.0-cp38-abi3-musllinux_1_1_x86_64.whl", hash = "sha256:f632fd56fc4e61564f78b46a2269153122db34988e78b6be8b32d28507b7eaeb", size = 312922, upload-time = "2025-09-25T19:50:04.232Z" },
    { url = "https://files.pythonhosted.org/packages/d6/3a/43d494dfb728f55f4e1cf8fd435d50c16a2d75493225b54c8d06122523c6/bcrypt-5.0.0-cp38-abi3-musllinux_2_2_aarch64.whl", hash = "sha256:801cad5ccb6b87d1b430f183269b94c24f248dddbbc5c1f78b6ed231743e001c", size = 341367, upload-time = "2025-09-25T19:50:05.559Z" },
    { url = "https://files.pythonhosted.org/packages/55/ab/a0727a4547e383e2e22a630e0f908113db37904f58719dc48d4622139b5c/bcrypt-5.0.0-cp38-abi3-musllinux_1_2_x86_64.whl", hash = "sha256:3cf67a804fc66fc217e6914a5635000259fbbbb12e78a99488e4d5ba445a71eb", size = 359187, upload-time = "2025-09-25T19:50:06.916Z" },
    { url = "https://files.pythonhosted.org/packages/1b/bb/461f352fdca663524b4643d8b09e8435b4990f17fbf4fea6bc2a90aa0cc7/bcrypt-5.0.0-cp38-abi3-win32.whl", hash = "sha256:3abeb543874b2c0524ff40c57a4e14e5d3a66ff33fb423529c88f180fd756538", size = 153752, upload-time = "2025-09-25T19:50:08.515Z" },
    { url = "https://files.pythonhosted.org/packages/41/aa/4190e60921927b7056820291f56fc57d00d04757c8b316b2d3c0d1d6da2c/bcrypt-5.0.0-cp38-abi3-win_amd64.whl", hash = "sha256:35a77ec55b541e5e583eb3436ffbbf53b0ffa1fa16ca6782279daf95d146dcd9", size = 150881, upload-time = "2025-09-25T19:50:09.742Z" },
    { url = "https://files.pythonhosted.org/packages/54/12/cd77221719d0b39ac0b55dbd39358db1cd1246e0282e104366ebbfb8266a/bcrypt-5.0.0-cp38-abi3-win_arm64.whl", hash = "sha256:cde08734f12c6a4e28dc6755cd11d3bdfea608d93d958fffbe95a7026ebe4980", size = 144931, upload-time = "2025-09-25T19:50:11.016Z" },
    { url = "https://files.pythonhosted.org/packages/5d/ba/2af136406e1c3839aea9ecadc2f6be2bcd1eff255bd451dd39bcf302c47a/bcrypt-5.0.0-cp39-abi3-macosx_10_12_universal2.whl", hash = "sha256:0c418ca99fd47e9c59a301744d63328f17798b5947b0f791e9af3c1c499c2d0a", size = 495313, upload-time = "2025-09-25T19:50:12.309Z" },
    { url = "https://files.pythonhosted.org/packages/ac/ee/2f4985dbad090ace5ad1f7dd8ff94477fe089b5fab2040bd784a3d5f187b/bcrypt-5.0.0-cp39-abi3-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:ddb4e1500f6efdd402218ffe34d040a1196c072e07929b9820f363a1fd1f4191", size = 275290, upload-time = "2025-09-25T19:50:13.673Z" },
    { url = "https://files.pythonhosted.org/packages/e4/6e/b77ade812672d15cf50842e167eead80ac3514f3beacac8902915417f8b7/bcrypt-5.0.0-cp39-abi3-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:7aeef54b60ceddb6f30ee3db090351ecf0d40ec6e2abf41430997407a46d2254", size = 278253, upload-time = "2025-09-25T19:50:15.089Z" },
    { url = "https://files.pythonhosted.org/packages/36/c4/ed00ed32f1040f7990dac7115f82273e3c03da1e1a1587a778d8cea496d8/bcrypt-5.0.0-cp39-abi3-manylinux_2_28_aarch64.whl", hash = "sha256:f0ce778135f60799d89c9693b9b398819d15f1921ba15fe719acb3178215a7db", size = 276084, upload-time = "2025-09-25T19:50:16.699Z" },
    { url = "https://files.pythonhosted.org/packages/e7/c4/fa6e16145e145e87f1fa351bbd54b429354fd72145cd3d4e0c5157cf4c70/bcrypt-5.0.0-cp39-abi3-manylinux_2_28_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:a71f70ee269671460b37a449f5ff26982a6f2ba493b3eabdd687b4bf35f875ac", size = 297185, upload-time = "2025-09-25T19:50:18.525Z" },
    { url = "https://files.pythonhosted.org/packages/24/b4/11f8a31d8b67cca3371e046db49baa7c0594d71eb40ac8121e2fc0888db0/bcrypt-5.0.0-cp39-abi3-manylinux_2_28_x86_64.whl", hash = "sha256:f8429e1c410b4073944f03bd778a9e066e7fad723564a52ff91841d278dfc822", size = 278656, upload-time = "2025-09-25T19:50:19.809Z" },
    { url = "https://files.pythonhosted.org/packages/ac/31/79f11865f8078e192847d2cb526e3fa27c200933c982c5b2869720fa5fce/bcrypt-5.0.0-cp39-abi3-manylinux_2_34_aarch64.whl", hash = "sha256:edfcdcedd0d0f05850c52ba3127b1fce70b9f89e0fe5ff16517df7e81fa3cbb8", size = 275662, upload-time = "2025-09-25T19:50:21.567Z" },
    { url = "https://files.pythonhosted.org/packages/d4/8d/5e43d9584b3b3591a6f9b68f755a4da879a59712981ef5ad2a0ac1379f7a/bcrypt-5.0.0-cp39-abi3-manylinux_2_34_x86_64.whl", hash = "sha256:611f0a17aa4a25a69362dcc299fda5c8a3d4f160e2abb3831041feb77393a14a", size = 278240, upload-time = "2025-09-25T19:50:23.305Z" },
    { url = "https://files.pythonhosted.org/packages/89/48/44590e3fc158620f680a978aafe8f87a4c4320da81ed11552f0323aa9a57/bcrypt-5.0.0-cp39-abi3-musllinux_1_1_aarch64.whl", hash = "sha256:db99dca3b1fdc3db87d7c57eac0c82281242d1eabf19dcb8a6b10eb29a2e72d1", size = 311152, upload-time = "2025-09-25T19:50:24.597Z" },
    { url = "https://files.pythonhosted.org/packages/5f/85/e4fbfc46f14f47b0d20493669a625da5827d07e8a88ee460af6cd9768b44/bcrypt-5.0.0-cp39-abi3-musllinux_1_1_x86_64.whl", hash = "sha256:5feebf85a9cefda32966d8171f5db7e3ba964b77fdfe31919622256f80f9cf42", size = 313284, upload-time = "2025-09-25T19:50:26.268Z" },
    { url = "https://files.pythonhosted.org/packages/25/ae/479f81d3f4594456a01ea2f05b132a519eff9ab5768a70430fa1132384b1/bcrypt-5.0.0-cp39-abi3-musllinux_2_2_aarch64.whl", hash = "sha256:3ca8a166b1140436e058298a34d88032ab62f15aae1c598580333dc21d27ef10", size = 341643, upload-time = "2025-09-25T19:50:28.02Z" },
    { url = "https://files.pythonhosted.org/packages/df/d2/36a086dee1473b14276cd6ea7f61aef3b2648710b5d7f1c9e032c29b859f/bcrypt-5.0.0-cp39-abi3-musllinux_2_2_x86_64.whl", hash = "sha256:61afc381250c3182d9078551e3ac3a41da14154fbff647ddf52a769f588c4172", size = 359698, upload-time = "2025-09-25T19:50:31.347Z" },
    { url = "https://files.pythonhosted.org/packages/c0/f6/688d2cd64bfd0b14d805ddb8a565e11ca1fb0fd6817175d58b10052b6d88/bcrypt-5.0.0-cp39-abi3-win32.whl", hash = "sha256:64d7ce196203e468c457c37ec22390f1a61c85c6f0b8160fd752940ccfb3a683", size = 153725, upload-time = "2025-09-25T19:50:34.384Z" },
    { url = "https://files.pythonhosted.org/packages/9f/b9/9d9a641194a730bda138b3dfe53f584d61c58cd5230e37566e83ec2ffa0d/bcrypt-5.0.0-cp39-abi3-win_amd64.whl", hash = "sha256:64ee8434b0da054d830fa8e89e1c8bf30061d539044a39524ff7dec90481e5c2", size = 150912, upload-time = "2025-09-25T19:50:35.69Z" },
    { url = "https://files.pythonhosted.org/packages/27/44/d2ef5e87509158ad2187f4dd0852df80695bb1ee0cfe0a684727b01a69e0/bcrypt-5.0.0-cp39-abi3-win_arm64.whl", hash = "sha256:f2347d3534e76bf50bca5500989d6c1d05ed64b440408057a37673282c654927", size = 144953, upload-time = "2025-09-25T19:50:37.32Z" },
]

[[package]]
name = "certifi"
version = "2026.7.22"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a3/c2/24167ea9858356b47a87a50d39908bfdb72ceeefe0041586e704e5376b3a/certifi-2026.7.22.tar.gz", hash = "sha256:741e2c3b351ddf169a738da9f2c048608ff7f2c5cc02f1ebc6b118bb090d5d55", size = 138112, upload-time = "2026-07-22T03:35:12.644Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/0b/a7/71ac2cff56fec219ed242bb11b8efb69fcc4bec75db06fb7bfe35de520e6/certifi-2026.7.22-py3-none-any.whl", hash = "sha256:62f22742b58a1a33014a2b6b706588a8d7e2a88ae7bd1a6ebe8c992928483775", size = 136983, upload-time = "2026-07-22T03:35:11.276Z" },
]
```

```
[[package]]
name = "cffi"
version = "2.1.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "pycparser", marker = "implementation_name != 'PyPy'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/9e/ef/008a1939e372c06329a3fce4279c02f328488f3526744906eeec3da7ad5f/cffi-2.1.1.tar.gz", hash =
 "sha256:dd31f52ea1086513bb9df30f8fcee9b8918323ae067a3d5b78bc826a000712be", size = 530807, upload-time = "2026-08-03T21:21:18.939Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/10/69/43965eccfdead3b9220015fd1320e117be8c6ed01a62ffab76eeb752f5d5/cffi-2.1.1-cp312-cp312-macosx_
10_15_x86_64.whl", hash = "sha256:c8c69575568085ba0b1b10c0249d779a214aea6f6522e949a0fc9fb0fcb449d0", size = 184821, upload-time = "2026-08-03T21:19:44
.887Z" },
    { url = "https://files.pythonhosted.org/packages/54/7d/16e5a096677b5e313ca80cd5e5170efa3ea44624a82bb111925522da64b1/cffi-2.1.1-cp312-cp312-macosx_
11_0_arm64.whl", hash = "sha256:f81b3b8f3d4e343550fa4baa0e479bba9f2d29ce9c2e9b51d1ce1718d7442fcf", size = 184719, upload-time = "2026-08-03T21:19:46.1
29Z" },
    { url = "https://files.pythonhosted.org/packages/56/e6/8941622732edec876dd17d0453dce07317ae96db34f2ec1436c9d3785986/cffi-2.1.1-cp312-cp312-manylin
ux1_i686.manylinux2014_i686.manylinux_2_17_i686.manylinux_2_5_i686.whl", hash = "sha256:811bd1e21d32de12efca32393a0ab3f5133b54fce9bd44b8bd77ab07da14bf
6a", size = 214799, upload-time = "2026-08-03T21:19:47.218Z" },
    { url = "https://files.pythonhosted.org/packages/44/de/f98430906df1545ffde0d543dd124a7a439bc2cd32b36b9c53f805df7333/cffi-2.1.1-cp312-cp312-manylin
ux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:68e62fe11f30d5ca8289242866f0a5291402d8529ca2178ab8afc5c9694ae890", size = 222389, upload-ti
me = "2026-08-03T21:19:48.331Z" },
    { url = "https://files.pythonhosted.org/packages/6a/5b/717f1526b9957b34456313c31645c5b82b8fb5c3fe9e4752999be7128bfc/cffi-2.1.1-cp312-cp312-manylin
ux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:4a7c934f7360e8cd64fe9efadcbd10c7c6364f531e432b9a4bf5ccbc9e0e8b50", size = 210249, upload-ti
me = "2026-08-03T21:19:49.543Z" },
    { url = "https://files.pythonhosted.org/packages/64/b3/f8aa4f3e34986c7e4ec45072d1b1b9dd295b6b18007b45518d79726dd725/cffi-2.1.1-cp312-cp312-manylin
ux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:3143d81e29e1e20a9ce10901ec369012947876596f75a222235965f2b7ae832e", size = 208775, upload-time =
 "2026-08-03T21:19:50.918Z" },
    { url = "https://files.pythonhosted.org/packages/b1/db/dceb9dd5b231e1da801793f8acc9f3c52a7e1afe40bb1aae37e02b0faad5/cffi-2.1.1-cp312-cp312-manylin
ux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:c1453022f490d2459a11819d83ad1d586e9ff65a12ac3e705ffebd46d3685dcf", size = 221822, upload-time
 = "2026-08-03T21:19:52.054Z" },
    { url = "https://files.pythonhosted.org/packages/a0/d2/6cd24ae3be000a634109c247d1475d62e5616d0dc78c82770942ec384248/cffi-2.1.1-cp312-cp312-musllin
ux_1_2_aarch64.whl", hash = "sha256:208f941bb9d18e768138677f0a6d2ce01f590df56043dda1df1535ac57c88517", size = 225232, upload-time = "2026-08-03T21:19:
53.109Z" },
    { url = "https://files.pythonhosted.org/packages/cb/52/3fa190537004dd7f0ab860a6dc7c0175b8667f68d1e618a46f5498d30250/cffi-2.1.1-cp312-cp312-musllin
ux_1_2_x86_64.whl", hash = "sha256:210019b6c7cf07f081b4c54635c8cf744377001350e29cc0f81c4377b4797735", size = 223597, upload-time = "2026-08-03T21:19:5
4.515Z" },
    { url = "https://files.pythonhosted.org/packages/80/fb/0bb75b7039588c074b37ae99f40d9bfddf990ecb2fbc346ebccd2e56b9be/cffi-2.1.1-cp312-cp312-win32.w
hl", hash = "sha256:046bfc24911b37851ee1b51aab8bffe713d89c68c6a057b09484ce9fd5f69b4e", size = 175292, upload-time = "2026-08-03T21:19:55.566Z" },
    { url = "https://files.pythonhosted.org/packages/d9/79/615cc094e2fb508cade7de88d3b4f6c4ec2bab695c97bce9153dc65aadf5/cffi-2.1.1-cp312-cp312-win_amd
64.whl", hash = "sha256:f53e442b08449d42821fa4a4fba000095af9f62742a500f978a9f557ec44339a", size = 185919, upload-time = "2026-08-03T21:19:56.89Z" },
    { url = "https://files.pythonhosted.org/packages/70/c6/d0ea84713fe46b243a436a18fcd47d639732747e21635c8a27191b06dc30/cffi-2.1.1-cp312-cp312-win_arm
64.whl", hash = "sha256:7bde5e4cc5c10140859842b9d383af292b22639a4dffb725314baf45968cef80", size = 180093, upload-time = "2026-08-03T21:19:58.155Z" },
    { url = "https://files.pythonhosted.org/packages/9d/f4/035513d4117049066b4779dc3b7c0c0fdad175fa13731c9f4003f1cd1478/cffi-2.1.1-cp313-cp313-ios_13_
0_arm64_iphoneos.whl", hash = "sha256:b5bdfd1c873d4e093aabc0ca84c4ca6dbc4f752afb5c86f146d9742580c9da2e", size = 194248, upload-time = "2026-08-03T21:1
9:59.399Z" },
    { url = "https://files.pythonhosted.org/packages/76/af/2aeb4dbb5fc41a04161ae9ff1518de7cec08e164f44a8ce6a4cf7fd2cd1d/cffi-2.1.1-cp313-cp313-ios_13_
0_arm64_iphonesimulator.whl", hash = "sha256:31348097ff5bbe827ccc41795d4dd099d9f0625e7def00ee653c137a490c2a6c", size = 196908, upload-time = "2026-08-
03T21:20:00.746Z" },
    { url = "https://files.pythonhosted.org/packages/a7/46/2e5fdde8555706dd98139a910ca11be02809f3f605ce956f655d0214e100/cffi-2.1.1-cp313-cp313-macosx_
10_15_x86_64.whl", hash = "sha256:9d2055050ea716bd38b7f7f1579c275386646b4894c155a3e2f3cd62ed41b7c6", size = 184805, upload-time = "2026-08-03T21:20:02
.02Z" },
    { url = "https://files.pythonhosted.org/packages/55/41/4c7042f317b9217502988f0873af87e16ad606dc20f84e546e3e6ce9764c/cffi-2.1.1-cp313-cp313-macosx_
11_0_arm64.whl", hash = "sha256:19ee6127ee34de7d83ce3d371ebc5ed91addbdcc39f9ab15ce4eb35a4e534971", size = 184764, upload-time = "2026-08-03T21:20:03.1
41Z" },
    { url = "https://files.pythonhosted.org/packages/43/1f/1c3d90d91811c8f86ced9ed637956c54bfe5b79ca98fe976d7f8c8979f6b/cffi-2.1.1-cp313-cp313-manylin
ux1_i686.manylinux2014_i686.manylinux_2_17_i686.manylinux_2_5_i686.whl", hash = "sha256:6a8dddef476fab96d066d578fc88526767b836ab5ab21754e1d5bf3879c31c
7c", size = 214722, upload-time = "2026-08-03T21:20:04.377Z" },
    { url = "https://files.pythonhosted.org/packages/37/6f/3b5ce4c3b2192d250f04908f2bfd91ef34552ec8f7716a5d4abdb8d67bb2/cffi-2.1.1-cp313-cp313-manylin
ux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:f16c709686a78c727bbbf059f92b0bf41c6fc60deec706d2dc19f529175a6125", size = 222369, upload-ti
me = "2026-08-03T21:20:05.544Z" },
    { url = "https://files.pythonhosted.org/packages/02/10/4b3c75dde3d9663c9e02ba05c2668b954f671d4bbe346413ca8c696b295a/cffi-2.1.1-cp313-cp313-manylin
ux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:fcd22650c908d7b7da162bbfaab594a1227a15d1643a98c68b122ac642fa2264", size = 210175, upload-ti
me = "2026-08-03T21:20:06.75Z" },
    { url = "https://files.pythonhosted.org/packages/df/62/14f74b9543e605d17701dc797b815958b8bb70b7624ce1b832ddad48ed6c/cffi-2.1.1-cp313-cp313-manylin
ux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:aa9511c62d14da7aacc9b4bf51f3f697a621e83b2d6919008243c3aad168eea3", size = 208670, upload-time =
 "2026-08-03T21:20:08.04Z" },
    { url = "https://files.pythonhosted.org/packages/95/95/86342356ff5953b3fb06f7ef7c5bee212d45e770abc7218d451b9148313c/cffi-2.1.1-cp313-cp313-manylin
ux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:a931079504ecc49efed7744c476a5c343a92fabf66dec2db95edb1b2fdc770e2", size = 221824, upload-time
 = "2026-08-03T21:20:09.274Z" },
    { url = "https://files.pythonhosted.org/packages/eb/ff/7b3429ff53aafe931ed8a5fc69f481bbef7ba6de87ddcbb63d08f483f613/cffi-2.1.1-cp313-cp313-musllin
ux_1_2_aarch64.whl", hash = "sha256:a2d7755bef5a12ed488f4ef1f1b69ee9191d7396083b755a5d2295f6edb4768b", size = 225148, upload-time = "2026-08-03T21:20:
10.7Z" },
    { url = "https://files.pythonhosted.org/packages/34/34/a95870b9221e09cf4f2ce3178b1a210abdfe63a1bd357da940418d7b8d15/cffi-2.1.1-cp313-cp313-musllin
ux_1_2_x86_64.whl", hash = "sha256:e0bcb7e0f677f543555d2adff3bf19c05f66cdb4796e5ff602442ab2fe3c4ef7", size = 223564, upload-time = "2026-08-03T21:20:1
2.165Z" },
    { url = "https://files.pythonhosted.org/packages/70/ea/839b50531021a647fb5e929f72cf97bc1ff702b5472166164b5b6e76b851/cffi-2.1.1-cp313-cp313-win32.w
hl", hash = "sha256:334644fbac4eff73d985a17a91226df55d0f394160c4cfb880e084c8f7161cac", size = 175263, upload-time = "2026-08-03T21:20:13.559Z" },
    { url = "https://files.pythonhosted.org/packages/60/a6/8b149b2c3f2e11aaa1618ef64500b45f50f22c57a977a4dff1aff1f91042/cffi-2.1.1-cp313-cp313-win_amd
64.whl", hash = "sha256:1aa5645c30469b09530c4ebca77ebf8f17618293c58f8549cb1a543a50236e7d", size = 185688, upload-time = "2026-08-03T21:20:14.69Z" },
    { url = "https://files.pythonhosted.org/packages/01/9a/11f687cb39d6a3504060d5242f04f48c735afb4d3d533958a20594890cb2/cffi-2.1.1-cp313-cp313-win_arm
64.whl", hash = "sha256:63bbfd5ded17c4840ac07cd8f1c21ba9d9708141f840b324f422f41b207e3973", size = 180078, upload-time = "2026-08-03T21:20:15.917Z" },
    { url = "https://files.pythonhosted.org/packages/d3/7b/d6bbf82b8b96e7391438898c42f5bd96dd02030fd5b64937d248220003e2/cffi-2.1.1-cp314-cp314-ios_13_
0_arm64_iphoneos.whl", hash = "sha256:7dbb61fe3a7699468030f71bbe5f8a0e326a151daa91beb11a6fc1f980c55e1c", size = 194064, upload-time = "2026-08-03T21:2
0:17.148Z" },
    { url = "https://files.pythonhosted.org/packages/94/e6/bcc91b283be94735e268487a054004f0aa19947b6348fa367db53230abc8/cffi-2.1.1-cp314-cp314-ios_13_
0_arm64_iphonesimulator.whl", hash = "sha256:f24fb43132a4c6b4cb4eb029492919b2db645be6808d738f244fd146c03c32cb", size = 196720, upload-time = "2026-08-
03T21:20:18.268Z" },
    { url = "https://files.pythonhosted.org/packages/d9/99/c4b0c17cacdc9c3b8f280026286a9826d6a208c0f047591a3c3ce99b91fd/cffi-2.1.1-cp314-cp314-macosx_
10_15_x86_64.whl", hash = "sha256:d28630f5854ab07ab1fd4aba756de52326c82e6be15d414b12793f1975048b54", size = 184964, upload-time = "2026-08-03T21:20:19
.708Z" },
    { url = "https://files.pythonhosted.org/packages/b3/a9/9db617d05d7367c1ad0ab00b3aa6e6f9281edd689b4ee9ea0e5a84e89c97/cffi-2.1.1-cp314-cp314-macosx_
11_0_arm64.whl", hash = "sha256:661c298b4821edebead0c91edd2b00374d67ad7c5a1f7a91d4442633b79d6a72", size = 184962, upload-time = "2026-08-03T21:20:20.8
33Z" },
    { url = "https://files.pythonhosted.org/packages/67/b8/b42132ca113dc567d37684437b46ca1dafc885902b02a110a02d5b511857/cffi-2.1.1-cp314-cp314-manylin
ux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:58acb8ab8e295e6c5ea12f888cbb13cf21511ef2a3303a23f4325c29d17fe5c1", size = 222328, upload-ti
me = "2026-08-03T21:20:22.118Z" },
    { url = "https://files.pythonhosted.org/packages/80/10/c5c0cbf0a657aecf59ef511409734230bf556f05a0d6c9eed7aa5c0a0166/cffi-2.1.1-cp314-cp314-manylin
ux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:456a61fa52d579ebf9df2e9552ead5129855dbaff6c1e5a9b1bc408809bdc062", size = 209985, upload-ti
me = "2026-08-03T21:20:23.401Z" },
    { url = "https://files.pythonhosted.org/packages/d5/6c/bfa0b87b03b9238148beca990292843c9396ba069b54496596594173de7b/cffi-2.1.1-cp314-cp314-manylin
ux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:a4f00aa42f75d6e4595e8866e748cc1705adc0cddfeb2ca86d0d03993d63ba03", size = 208530, upload-time =
 "2026-08-03T21:20:24.628Z" },
    { url = "https://files.pythonhosted.org/packages/e9/02/4e7d553a7ac4b4238b38b3c1b80d486e9d4436f8d2acbf87a0997fe3f402/cffi-2.1.1-cp314-cp314-manylin
ux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:b0431303acaea1089ad4b3e9ce4e6518193def1118d4073ca848635ee4ea2e96", size = 221525, upload-time
```

```
 = "2026-08-03T21:20:25.758Z" },
    { url = "https://files.pythonhosted.org/packages/82/1d/a4aaf9babd75acb4d5f223bff71533bee748dd770a382619a798960ee9ba/cffi-2.1.1-cp314-cp314-musllin
ux_1_2_aarch64.whl", hash = "sha256:64faea20f4e2613363a1a9b9c7dd73058f3ecd00133a511e72ad7c511658f527", size = 225053, upload-time = "2026-08-03T21:20:
26.985Z" },
    { url = "https://files.pythonhosted.org/packages/81/10/5dc0e7bdd18e22107054288283380fc97a06ae3f1656a106908d666a3c88/cffi-2.1.1-cp314-cp314-musllin
ux_1_2_x86_64.whl", hash = "sha256:5c58fe613dc5e5336357eff555824a314d8e43282600435c8d1cb6a7a2fedd13", size = 223213, upload-time = "2026-08-03T21:20:2
8.277Z" },
    { url = "https://files.pythonhosted.org/packages/0b/e9/d0061c364cde06ee43168a0d076ac1da512cbc380d44767b844ba34fe2b6/cffi-2.1.1-cp314-cp314-win32.w
hl", hash = "sha256:1a18a57b58cfb21fc28d72e876acf10eaed67a1ed96226f92af4df681d571c4c", size = 177682, upload-time = "2026-08-03T21:20:44.288Z" },
    { url = "https://files.pythonhosted.org/packages/a7/06/1c3e01e3ba14c39f6d10bfbac52753b7e22259e38088e5cfe1d704918690/cffi-2.1.1-cp314-cp314-win_amd
64.whl", hash = "sha256:3222ba5d678f80a030e6afbcc33dc1ae5cb45facabb61cee2c7016b8432fde48", size = 187949, upload-time = "2026-08-03T21:20:45.623Z" },
    { url = "https://files.pythonhosted.org/packages/87/5b/da4e39efe18eeb89cf580ea9cfc66b6a7c3eadb808fc0cc1d3a295cb5a5d/cffi-2.1.1-cp314-cp314-win_arm
64.whl", hash = "sha256:ab36d55f9ed2d067327667c2fea18dda018eb628dd6347aa01dda6cf1f5d3836", size = 182947, upload-time = "2026-08-03T21:20:46.955Z" },
    { url = "https://files.pythonhosted.org/packages/23/59/40338bf421c5accea1d45158170c87006ef1cd371b05c077e76476949728/cffi-2.1.1-cp314-cp314t-macosx
_10_15_x86_64.whl", hash = "sha256:7750c6449dff7864bb9bb27ddfb0267756189201a3afc911d82b3caacd70dfc3", size = 188504, upload-time = "2026-08-03T21:20:2
9.495Z" },
    { url = "https://files.pythonhosted.org/packages/7d/47/5ecf1023850036e674c77ec4de86182d309ae344e39e7cba984b7df5d647/cffi-2.1.1-cp314-cp314t-macosx
_11_0_arm64.whl", hash = "sha256:0beceaabe56af686895136a2de78db54ecd8e4046b236b8fd6d6cb61389e9bf2", size = 188259, upload-time = "2026-08-03T21:20:31.
291Z" },
    { url = "https://files.pythonhosted.org/packages/2a/9c/92934c3bea9f785b23eba304538c0b4d37a2a96d2431eb3a1bc87a11aa19/cffi-2.1.1-cp314-cp314t-manyli
nux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:49cbc70e6542d4ccccb936558d1064a8012541e78f821f955cff24e357776c94", size = 223864, upload-t
ime = "2026-08-03T21:20:32.571Z" },
    { url = "https://files.pythonhosted.org/packages/4d/45/ba4c93527bc38616a8bd36488acb69a2212d60486794f0c1f318949bbb76/cffi-2.1.1-cp314-cp314t-manyli
nux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:e2d65b31f36619cda3999b78b2aa9632e76b78448e7a56fc4240824200e7c4fc", size = 211538, upload-t
ime = "2026-08-03T21:20:33.808Z" },
    { url = "https://files.pythonhosted.org/packages/80/e9/b6ef565e452acb932fb0cb5443f44a78efbd1233e566f02b5a83855e9115/cffi-2.1.1-cp314-cp314t-manyli
nux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:28907ab9bfb6aa13184cfc17c6b8e1023c5ab6fd7076d8c20a35e59fe04f8f29", size = 210688, upload-time
= "2026-08-03T21:20:34.974Z" },
    { url = "https://files.pythonhosted.org/packages/9a/95/eff5f0cee78d2eabc7eebffec40d3fc1876b5f3c95582e018bb4b99601f2/cffi-2.1.1-cp314-cp314t-manyli
nux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:51b31d1c98274844cfd7838ce00bfc27c7423a4dc00fc0772fc3331c2cc90676", size = 223803, upload-tim
e = "2026-08-03T21:20:36.564Z" },
    { url = "https://files.pythonhosted.org/packages/fa/01/579d39fb8bef00a335a23d83757b44feb24cd6345a2c451b64cb67b9c362/cffi-2.1.1-cp314-cp314t-muslli
nux_1_2_aarch64.whl", hash = "sha256:5e7cecbaadb83884793e05828cee59b210b24583b9c7425d0ba6a754fe22eb4e", size = 226763, upload-time = "2026-08-03T21:20
:37.816Z" },
    { url = "https://files.pythonhosted.org/packages/8d/b0/0b44f47c60b01b57b6e2bbd92343f13a85a1d93bc46ccf6e47e244acd99c/cffi-2.1.1-cp314-cp314t-muslli
nux_1_2_x86_64.whl", hash = "sha256:25792eac27877609e7bb06d42ff88278a6624fff2ba9bbb523c09616b117e80f", size = 225688, upload-time = "2026-08-03T21:20:
38.959Z" },
    { url = "https://files.pythonhosted.org/packages/eb/d2/3b7176cb570a1d3e27faf67b72f591af508036e0d8b2be2ef9af9e8c84bb/cffi-2.1.1-cp314-cp314t-win32.
whl", hash = "sha256:8ef53b2de9bcb9197d31854256575d59dbac0cba72ac627bb291ef5eceb74be4", size = 182868, upload-time = "2026-08-03T21:20:40.388Z" },
    { url = "https://files.pythonhosted.org/packages/56/78/31f00c1bcd97c9bbf55f1bfdf5bc809a5de8887473e90bb9960dca825e80/cffi-2.1.1-cp314-cp314t-win_am
d64.whl", hash = "sha256:616f097f2fe415bc92a247f02e11f634e1f9e9a83d327e3c915c15089c87869e", size = 194104, upload-time = "2026-08-03T21:20:41.725Z" },
    { url = "https://files.pythonhosted.org/packages/7b/1b/58496f2ed0a35de575250c02a43ab3cc2c04d494a88fed31c1cabc0fd176/cffi-2.1.1-cp314-cp314t-win_ar
m64.whl", hash = "sha256:ad2c86c495b899d862ea0f4b42891b8713a3bd45dd4105c7fd51c2a72f39f3a5", size = 186402, upload-time = "2026-08-03T21:20:43.042Z" },
    { url = "https://files.pythonhosted.org/packages/c1/8f/9ebe220eab48a093d1a5a5e339ab0dc7316eef3bb04d63c42f0251b61f50/cffi-2.1.1-cp315-cp315-ios_13_
0_arm64_iphoneos.whl", hash = "sha256:dddad92b554513a31f272570678ba307fb9f618f05e3d4a5eacafff9eae03e1d", size = 194043, upload-time = "2026-08-03T21:2
0:48.179Z" },
    { url = "https://files.pythonhosted.org/packages/ff/69/844bad3ece306c4782c2ecb93597035b6690d48704b803914c199da1e8b3/cffi-2.1.1-cp315-cp315-ios_13_
0_arm64_iphonesimulator.whl", hash = "sha256:da0e573f9f97159390c89d9f1a9e41908b66d408cc5b58d08cf3847d844c531b", size = 196737, upload-time = "2026-08-
03T21:20:49.457Z" },
    { url = "https://files.pythonhosted.org/packages/1b/8a/af668013284634733f02d683458a0728739c7d6ddb5e14cb0c20832266fe/cffi-2.1.1-cp315-cp315-macosx_
10_15_x86_64.whl", hash = "sha256:fb92203a88b3d3053034db775110081c49d28be6551923805e039924093761e4", size = 184933, upload-time = "2026-08-03T21:20:50
.639Z" },
    { url = "https://files.pythonhosted.org/packages/0c/75/2f5207ff6d1a613133b23a5203cc0c2a628313b5eb3974d7956ae3c57950/cffi-2.1.1-cp315-cp315-macosx_
11_0_arm64.whl", hash = "sha256:2ae64be792b8966f2c69538199728b290e34726562896df1e5dc8ffd8d8188e8", size = 185002, upload-time = "2026-08-03T21:20:52.1
73Z" },
    { url = "https://files.pythonhosted.org/packages/e2/31/9e1313b0a6e30e91b3b3d3fff51ae99c857c07738e3afcce1f7334e1b7ab/cffi-2.1.1-cp315-cp315-manylin
ux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:507a24c282e0f42f8ed737cf048572cbf580468da5555764a8331735e9c736b6", size = 222271, upload-ti
me = "2026-08-03T21:20:53.462Z" },
    { url = "https://files.pythonhosted.org/packages/50/e3/f6234a833e6e08c7007003074723c406559eecf9b48dfc97471e5a8eb7a0/cffi-2.1.1-cp315-cp315-manylin
ux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:246fa40ce8645a614ff682e0b70f37134e460eaf93a775e0cbe3cca585a67a80", size = 209919, upload-ti
me = "2026-08-03T21:20:54.783Z" },
    { url = "https://files.pythonhosted.org/packages/0d/fc/5f74e293fced6edb51af3a46c4ccf6c23c9943774ecb375ddbd522c76add/cffi-2.1.1-cp315-cp315-manylin
ux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:471cee653ae88de62096552e6d24ccb4a5adb8c8c9f10b5054d0122c15bf2779", size = 208529, upload-time =
 "2026-08-03T21:20:56.066Z" },
    { url = "https://files.pythonhosted.org/packages/44/16/29e6d01b388bef055ecd6ca8244b3f4d336bd09e92d5d892187b9601084e/cffi-2.1.1-cp315-cp315-manylin
ux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:aeae0e330c9f6acd681f647d46cefd30c29f93e3392882e792e82080c9691399", size = 221630, upload-time
 = "2026-08-03T21:20:57.336Z" },
    { url = "https://files.pythonhosted.org/packages/a4/18/fa7f1f6857d5eb88a4ca99ffcbfb7c387a287ccc154c64a73e86314745d7/cffi-2.1.1-cp315-cp315-muslli
ux_1_2_aarch64.whl", hash = "sha256:42a494cee34437f05546455144f2b5d9ac09b1face62bcfce597d2e521066688", size = 225134, upload-time = "2026-08-03T21:20:
58.675Z" },
    { url = "https://files.pythonhosted.org/packages/e0/9f/e8e3dfa04a1b4c241f8c91faacad872b4d4efd051d49764ad4e2fd4b9fea/cffi-2.1.1-cp315-cp315-muslli
ux_1_2_x86_64.whl", hash = "sha256:cc572dace3f60ef98d7b12ff411d20f5362feb31a0439eab0085bbfd349982d7", size = 223197, upload-time = "2026-08-03T21:20:5
9.968Z" },
    { url = "https://files.pythonhosted.org/packages/f8/7e/8debeb04f1ab9fe2a6963964cd6f1aaf7192627b83926586a6a4e089c9fa/cffi-2.1.1-cp315-cp315-win32.w
hl", hash = "sha256:4f42141fc14250de6dde5ee7ea4432be017252d91f19c5ad043c084cea629cac", size = 177683, upload-time = "2026-08-03T21:21:14.901Z" },
    { url = "https://files.pythonhosted.org/packages/e0/31/5158704cc474ab65c1647932e88be78dc0873f47130e253be38bcaf13d01/cffi-2.1.1-cp315-cp315-win_amd
64.whl", hash = "sha256:e6e8cff14d6fb0be70a09c0bdc58096f501952d04624ebf867e0e56da2df8960", size = 187897, upload-time = "2026-08-03T21:21:16.108Z" },
    { url = "https://files.pythonhosted.org/packages/cc/4b/b3a2da8570c704ffc0f9762cdc3ec0f02c8573798e0b5cf7f11c82bbb70f/cffi-2.1.1-cp315-cp315-win_arm
64.whl", hash = "sha256:27350daa11d4f10c540e6e89dada4c54feb7256ad03e9a4dc075ebad7ba360d1", size = 182935, upload-time = "2026-08-03T21:21:17.271Z" },
    { url = "https://files.pythonhosted.org/packages/d0/ef/5443574510a1207e6f6bc38ba6e1f1de36cb48fef07b2728bb896a21f430/cffi-2.1.1-cp315-cp315t-macosx
_10_15_x86_64.whl", hash = "sha256:c26608d2222fb1e94487e4a387d85f13eb55d5ed725cb25a0c589ac4ee60e7bc", size = 188464, upload-time = "2026-08-03T21:21:0
1.163Z" },
    { url = "https://files.pythonhosted.org/packages/7e/ae/a56fa8c4686ad50e148fcbc8d3ae0d03915ff5c30d795058988c24118cef/cffi-2.1.1-cp315-cp315t-macosx
_11_0_arm64.whl", hash = "sha256:4be96343e422f2dfcd12ab5c9f5aebe03f82f737c6bffeca6830b3875cb44aab", size = 188262, upload-time = "2026-08-03T21:21:02.
382Z" },
    { url = "https://files.pythonhosted.org/packages/53/b2/6187f46f2912276a3ae284076109cc5c8680482f11f766ccf26db4a86427/cffi-2.1.1-cp315-cp315t-manyli
nux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:937c0052c05a31ca1daf18de3158eed4dbfcb9cc107adbea227728d647be701e", size = 223779, upload-t
ime = "2026-08-03T21:21:03.553Z" },
    { url = "https://files.pythonhosted.org/packages/8a/f6/c3ad28bd19f77047a03084424fbd4cbe997303267c14423737324be0385d/cffi-2.1.1-cp315-cp315t-manyli
nux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:df423d40ee8654634421812bc3b196da3f9bd7d32929da813f8394c4348a5358", size = 211520, upload-t
ime = "2026-08-03T21:21:04.863Z" },
    { url = "https://files.pythonhosted.org/packages/a0/cd/ccac9013a5bd9fd764de118674ab9c805b5ca10c19270d90ee273f8b2240/cffi-2.1.1-cp315-cp315t-manyli
nux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:a730a083190634c65cca36ba5f489531576ebd79bcd5c8e172130f6453127231", size = 210673, upload-time
= "2026-08-03T21:21:06.223Z" },
    { url = "https://files.pythonhosted.org/packages/52/86/2976131c639aead931c5bee5aba67e4b09fbeb8018b6f282f70803f923a7/cffi-2.1.1-cp315-cp315t-manyli
nux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:363e05fa78e15116c3c32c210ee36884fd6b9afa6d440e47112c3bd511d64cb6", size = 223835, upload-tim
e = "2026-08-03T21:21:07.539Z" },
    { url = "https://files.pythonhosted.org/packages/ac/0c/33a7aeab2f9c76918c52e084beb39c570db3588133412929e8ec06fab90b/cffi-2.1.1-cp315-cp315t-muslli
nux_1_2_aarch64.whl", hash = "sha256:770de9db11e84213beec501cfcaa013b019820ca881e03344dea5844f7876d94", size = 226705, upload-time = "2026-08-03T21:21
:08.774Z" },
    { url = "https://files.pythonhosted.org/packages/e3/26/2cde30fdde421130bfc18f70395731a6e6b2053c6a1978a5258ff04e72fa/cffi-2.1.1-cp315-cp315t-muslli
nux_1_2_x86_64.whl", hash = "sha256:7da0c5eff80f0197f3b3d1232ec5a682a9325f4ae9016a78f5f5ca35f9ced1f5", size = 225539, upload-time = "2026-08-03T21:21:
09.911Z" },
    { url = "https://files.pythonhosted.org/packages/6d/cd/a361394c94b2129d604bb846f624a8e88255a3ee33129c434a00d715e64f/cffi-2.1.1-cp315-cp315t-win32.
whl", hash = "sha256:06c72bb76605a4b0cd0aad6930b69d4baf7dd5d806cfc409b824191099700e66", size = 182707, upload-time = "2026-08-03T21:21:11.226Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/9b/b5/ba2b299993c26577d529b6ae29841f9e15b9fcf004d65f423f4fcf94ade9/cffi-2.1.1-cp315-cp315t-win_amd64.whl", hash = "sha256:d9c275eaacd24aa73f94ffd6de08fc3f932424d8b6c376f4bed7cde376fe7bc3", size = 193772, upload-time = "2026-08-03T21:21:12.39Z" },
    { url = "https://files.pythonhosted.org/packages/aa/29/35e016098c814cd93de9cd320c66b5bfba14dc6ecedd3cb518fa7c408c69/cffi-2.1.1-cp315-cp315t-win_arm64.whl", hash = "sha256:d18e5ac0f2f03f4f518d3e23db0f0cad7faa1da8620e9c09461d443bbf6e6692", size = 186360, upload-time = "2026-08-03T21:21:13.636Z" },
]

[[package]]
name = "charset-normalizer"
version = "3.4.9"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/bd/2a/23f34ec9d04624958e137efdc394888716353190e75f25dd22c7a2c7a8aa/charset_normalizer-3.4.9.tar.gz", hash = "sha256:673611bbd43f0810bec0b0f028ddeaaa501190339cac411f347ac76917c3ae7b", size = 152439, upload-time = "2026-07-07T14:34:58.454Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/70/4a/ecbd131485c07fcdfad54e28946d513e3da22ef3b4bd854dcafae54ec739/charset_normalizer-3.4.9-cp312-cp312-macosx_10_13_universal2.whl", hash = "sha256:45b0cc4e3556cd875e09102988d1ab8356c998b596c9fced84547c8138b487a0", size = 319300, upload-time = "2026-07-07T14:33:15.666Z" },
    { url = "https://files.pythonhosted.org/packages/ec/96/5d9364e3342d69f3a045e1777bc47c85c383e6e9466d561b33fdb419d1f9/charset_normalizer-3.4.9-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:9b2aff1c7b3884512b9512c3eaadd9bab39fb45042ffaaa1dd08ff2b9f8109d9", size = 215802, upload-time = "2026-07-07T14:33:17.031Z" },
    { url = "https://files.pythonhosted.org/packages/4b/4c/5361f9aa7f2cb58d94f2ab831b3d493f69efb1d239654b4744e3c09527cb/charset_normalizer-3.4.9-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:9104ed0bd76a429d46f9ec0dbc9b08ad1d2dcdf2b00a5a0daa1c145329b35b44", size = 237171, upload-time = "2026-07-07T14:33:18.576Z" },
    { url = "https://files.pythonhosted.org/packages/50/78/ce342ca4ff30b2eb49fe6d9578df85974f90c67d294113e94efdd9664cbd/charset_normalizer-3.4.9-cp312-cp312-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:7b86a2b16095d250c6f58b3d9b2eee6f4147754344f3dab0922f7c9bf7d26c9", size = 233075, upload-time = "2026-07-07T14:33:20.084Z" },
    { url = "https://files.pythonhosted.org/packages/01/c4/4fa4c8b3097a11f3c5f09a35b72ed6855fb1d332469504962ab7bafcc702/charset_normalizer-3.4.9-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:5e226f6218febc71f6c1fc2fafb91c226f75bdc1d8fb12d66823716e891608fd", size = 224256, upload-time = "2026-07-07T14:33:21.747Z" },
    { url = "https://files.pythonhosted.org/packages/87/3a/ad914516df7e358a81aae018caa5e0470ba827fa6d763b1d2e87d920a5f6/charset_normalizer-3.4.9-cp312-cp312-manylinux_2_31_armv7l.whl", hash = "sha256:90c44bc373b7687f6948b693cceaea1348ae0975d7474746559494468e3c1d84", size = 208784, upload-time = "2026-07-07T14:33:23.313Z" },
    { url = "https://files.pythonhosted.org/packages/d7/74/3c12f9755717dfe5c5c87da63f35d765fa0c00382ec26bf23f7fae34f2ba/charset_normalizer-3.4.9-cp312-cp312-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:9cdef90ae47919cae358d8ab15797a800ed41da7aba5d72419fb510729e2ed4b", size = 219928, upload-time = "2026-07-07T14:33:24.814Z" },
    { url = "https://files.pythonhosted.org/packages/33/9a/895095b83e7907abd6d3d99aad3a38ad0d9686cc186cb0c94c24320fe63e/charset_normalizer-3.4.9-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:60f44ade2cf573dad7a277e6f8ca9a51a21dda572b13bd7d8539bb3cd5dbedde", size = 218489, upload-time = "2026-07-07T14:33:26.42Z" },
    { url = "https://files.pythonhosted.org/packages/a1/34/ef5c05f412f42520d7709b7d3784d19640839eb7366ded1755511585429f/charset_normalizer-3.4.9-cp312-cp312-musllinux_1_2_armv7l.whl", hash = "sha256:a1786910334ed46ab1dd73222f2cd1e05c2c3bb39f6dddb4f8b36fc382058a39", size = 210267, upload-time = "2026-07-07T14:33:27.952Z" },
    { url = "https://files.pythonhosted.org/packages/83/dc/9b29fa4412b318bf3bfea985c35d67eb55e04b59a7c3f2237168b0e0be6f/charset_normalizer-3.4.9-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:03d07803992c6c7bbc976327f34b18b6160327fc81cb82c9d504720ac0be3b62", size = 226030, upload-time = "2026-07-07T14:33:29.397Z" },
    { url = "https://files.pythonhosted.org/packages/0e/42/6dbc00b8cd16011691203e33570fa42ed5746599a2e878112d16eab403a3/charset_normalizer-3.4.9-cp312-cp312-win32.whl", hash = "sha256:78841cccf1af7b40f6f716338d50c0902dbe88d9f800b3c973b7a9a0a693a642", size = 151185, upload-time = "2026-07-07T14:33:30.781Z" },
    { url = "https://files.pythonhosted.org/packages/80/cc/f920afd1a23c58ccd53c1d36085a71893a4737ff5e66e0371efab6809850/charset_normalizer-3.4.9-cp312-cp312-win_amd64.whl", hash = "sha256:4b3dac63058cc36820b0dd072f89898604e2d39686fe05321729d00d8ac185a0", size = 162557, upload-time = "2026-07-07T14:33:32.176Z" },
    { url = "https://files.pythonhosted.org/packages/f0/e6/0386d43a261ff4e4b30c5857af7df877254b46bec7b9d1b74b6bf969a90b/charset_normalizer-3.4.9-cp312-cp312-win_arm64.whl", hash = "sha256:78fa18e436a1a0e58dbd7e02fc4473f3f32cceb12df9dfca542d075961c307d2", size = 152665, upload-time = "2026-07-07T14:33:33.711Z" },
    { url = "https://files.pythonhosted.org/packages/b2/06/97ec2aeae780b31d742b6352218b43841a6871e2564578ca522dce4a45c3/charset_normalizer-3.4.9-cp313-cp313-macosx_10_13_universal2.whl", hash = "sha256:440eede83796000d74978f0eba527be106b5b9aee0daf779d395276ed0b0614", size = 317688, upload-time = "2026-07-07T14:33:35.408Z" },
    { url = "https://files.pythonhosted.org/packages/d0/39/8ff066c672434225f8d25f8b739f992af250944392173dcc88362681c9bf/charset_normalizer-3.4.9-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:21e764fd1e70b6a3e205a0e46f3051701f98a8cb3fad66eeb80e48bb502f8698", size = 214982, upload-time = "2026-07-07T14:33:36.996Z" },
    { url = "https://files.pythonhosted.org/packages/92/8f/3a47a3667c83c2df9483d91644c6c107de3bf8874aa1793da9d3012eb986/charset_normalizer-3.4.9-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:e4fd89cc178bced6ad29cb3e6dd4aa63fa5017c3524dbd0b25998fb64a87cc8b", size = 236460, upload-time = "2026-07-07T14:33:38.536Z" },
    { url = "https://files.pythonhosted.org/packages/f1/60/b22cdbee7e4013dab8b0d7647fc6181120fbbbc8f7025c226d15bd5a47fc/charset_normalizer-3.4.9-cp313-cp313-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:bd47ba7fc3ca94896759ea0109775132d3e7ab921fbf54038e1bab2e46c313c9", size = 232003, upload-time = "2026-07-07T14:33:40.059Z" },
    { url = "https://files.pythonhosted.org/packages/ea/f8/72eb13dcabe7257035cea8aefd922caad2f110d252bf9f67c4c2ca763aee/charset_normalizer-3.4.9-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:84fd18bcc17526fc2b3c1af7d2b9217d32c9c04448c16ec693b9b4f1985c3d33", size = 223149, upload-time = "2026-07-07T14:33:41.631Z" },
    { url = "https://files.pythonhosted.org/packages/b0/3e/faee8f9de92b14ee1198e9163252bb15efee7301b31256a3b6d9ebfdd0dd/charset_normalizer-3.4.9-cp313-cp313-manylinux_2_31_armv7l.whl", hash = "sha256:5b10cd92fc5c498b35a8635df6d5a100207f88b63a4dc1de7ef9a548e1e2cd63", size = 207901, upload-time = "2026-07-07T14:33:43.209Z" },
    { url = "https://files.pythonhosted.org/packages/3a/25/45f30093ae27dd7b92a793b61882a38685f993700113ca36e0c9c14965e1/charset_normalizer-3.4.9-cp313-cp313-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:a4fbdde9dd4a9ce5fd52c2b3a347bb50cc89483ef783f1cb00d408c13f7a96c0", size = 219176, upload-time = "2026-07-07T14:33:44.725Z" },
    { url = "https://files.pythonhosted.org/packages/48/18/c8f397329c35e32f6a837e488986f4ae03bd2abebc453b48714991630c2f/charset_normalizer-3.4.9-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:416c229f77e5ea25b3dfd4b582f8d73d7e43c22320302b9ab128a2d3a0b38efe", size = 217356, upload-time = "2026-07-07T14:33:46.192Z" },
    { url = "https://files.pythonhosted.org/packages/86/7e/5ce0bba863470fd1902d5e5843968951bddf38abe4742fc97116ef4598b3/charset_normalizer-3.4.9-cp313-cp313-musllinux_1_2_armv7l.whl", hash = "sha256:75286256590a6320cf106a0d28970d3560aad9ee09aa7b34fb40524792436d35", size = 209614, upload-time = "2026-07-07T14:33:47.705Z" },
    { url = "https://files.pythonhosted.org/packages/6c/ef/2473d3c4d869155be4af1191111d59c4d5c4e0173026f7e85b176e23bf65/charset_normalizer-3.4.9-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:69b157c5d3292bcd443faca052f3096f637f1e074b98212a933c074ae23dc3b8", size = 224991, upload-time = "2026-07-07T14:33:49.238Z" },
    { url = "https://files.pythonhosted.org/packages/d0/a3/53ddae3db108a088156aa8ddfafd411ebbc1340f48c5573f697b27f69a39/charset_normalizer-3.4.9-cp313-cp313-win32.whl", hash = "sha256:51307f5c71007673a2bf8232ad973483d281e74cb99c8c5a990af1eefa6277d9", size = 150622, upload-time = "2026-07-07T14:33:50.711Z" },
    { url = "https://files.pythonhosted.org/packages/e8/ef/6953a77c7cf2c2ff9998e6f575ab3e380119f100223381565a4f94c1f836/charset_normalizer-3.4.9-cp313-cp313-win_amd64.whl", hash = "sha256:fe2c7201c642b7c308f1675355ad7ff7b66acfe3541625efe5a3ad38f29d6115", size = 161947, upload-time = "2026-07-07T14:33:52.197Z" },
    { url = "https://files.pythonhosted.org/packages/6e/fb/d560d1d1555debbfe7849d9cac6145c1b537709d79576bf22557ed803b82/charset_normalizer-3.4.9-cp313-cp313-win_arm64.whl", hash = "sha256:611057cc5d5c0afc743ba8be6bd828c17e0aaa8643f9d0a9b9bb7dea80eb8012", size = 152594, upload-time = "2026-07-07T14:33:53.486Z" },
    { url = "https://files.pythonhosted.org/packages/7e/8d/496817fa0944239ecae662dd57ea765cfeaec6a735f9f025d4b7b72e7143/charset_normalizer-3.4.9-cp314-cp314-macosx_10_15_universal2.whl", hash = "sha256:0327fcd59a935777d83410750c50600ee9571af2846f71ce40f25b13da1ef380", size = 317253, upload-time = "2026-07-07T14:33:54.994Z" },
    { url = "https://files.pythonhosted.org/packages/2b/f9/ef4a69ea338ad3c0deceea0f5f7d2380ae8b52132b06d652cb0d2cd86706/charset_normalizer-3.4.9-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:8a79d9f4d8001473a30c163556b3c3bfebec837495a412dde78b51672f6134f9", size = 215898, upload-time = "2026-07-07T14:33:56.334Z" },
    { url = "https://files.pythonhosted.org/packages/8c/e7/5ddfd76fc061eb52de219658a4aa431cbacadf0a0219c8854f00da50d289/charset_normalizer-3.4.9-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:33bdcc2a32c0a0e861f60841a512c8acc658c87c2ac59d89e3a46dacf7d866e4", size = 236718, upload-time = "2026-07-07T14:33:57.9Z" },
    { url = "https://files.pythonhosted.org/packages/49/ba/768fa3f36048d81c477a0ce61f813bc1454d80917ccfe550abd9f44f5e24/charset_normalizer-3.4.9-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:f840ed6d8ecba8255df8c42b87fadeda98ddfc6eeec05e2dc66e26d46dd6f58a", size = 232519, upload-time = "2026-07-07T14:33:59.811Z" },
    { url = "https://files.pythonhosted.org/packages/f4/c4/b3e049d2aa3766180c78507110543d9d50894cc97f57de543f1be521dcdc/charset_normalizer-3.4.9-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:c25fe15c70c59eb7c5ce8c06a1f3fa1da0ecc5ea1e7a5922c40fd2fa9
```

```
b0d5046", size = 223143, upload-time = "2026-07-07T14:34:01.517Z" },
    { url = "https://files.pythonhosted.org/packages/19/79/55c32d06d76ae4feafe053f061f3e3ab70bcf19f4007797ce8c3efda7830/charset_normalizer-3.4.9-cp314
-cp314-manylinux_2_31_armv7l.whl", hash = "sha256:f7fb7d750cfa0a070d2c24e831fd3481019a60dd317ea2b39acbcebc08b6ed81", size = 206742, upload-time = "202
6-07-07T14:34:03.04Z" },
    { url = "https://files.pythonhosted.org/packages/10/e0/47c079dd82d217c807479cd59ffd30af56307ea31c108b75758970459ad3/charset_normalizer-3.4.9-cp314
-cp314-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:4d1c96a7a18b9690a4d46df09e3e3382406ae3213727cd1019ebade1c4a81917", size = 21
9191, upload-time = "2026-07-07T14:34:04.657Z" },
    { url = "https://files.pythonhosted.org/packages/42/ab/b9bc2e77d6b44a7e46ef62ec5cac1c9a6ba7b9135a5d560f002696ec9995/charset_normalizer-3.4.9-cp314
-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:a4cfde78a9f2880208d16a93b795726a3017d5977e08d1e162a7a31322479c41", size = 218328, upload-time = "202
6-07-07T14:34:06.115Z" },
    { url = "https://files.pythonhosted.org/packages/f1/78/c9c71d599f5aa2d42bcdd35cbbd46d7f535351a57e40ff7d8e5a7e219401/charset_normalizer-3.4.9-cp314
-cp314-musllinux_1_2_armv7l.whl", hash = "sha256:d4d6fcde76f94f5cb9e43e9e9a61f16dacefd228cbbf6f1a09bd9b219a92f1a1", size = 207406, upload-time = "2026
-07-07T14:34:07.554Z" },
    { url = "https://files.pythonhosted.org/packages/f6/39/c914445c321a845097ce4f6ac7de9a18228a77b766272125a1ce00d851eb/charset_normalizer-3.4.9-cp314
-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:898f0e9068ca27d37f8e83a5b962821df851532e6c4a7d615c1c033f9da6eedf", size = 225157, upload-time = "2026
-07-07T14:34:09.061Z" },
    { url = "https://files.pythonhosted.org/packages/9b/f2/c0d4b8508565a36bc5c624e88ed297f5b0b1095011034d7f5b83a69908b5/charset_normalizer-3.4.9-cp314
-cp314-win32.whl", hash = "sha256:c1c948747b03be832dceed96ca815cef7360de9aa19d37c730f8e3f6101aca48", size = 151095, upload-time = "2026-07-07T14:34:10
.901Z" },
    { url = "https://files.pythonhosted.org/packages/49/fd/a1d26144398c67486422a72bf5812cda22cb4ccfcd95a290fb41ceb4b8e2/charset_normalizer-3.4.9-cp314
-cp314-win_amd64.whl", hash = "sha256:16b65ea0f2465b6fb52aa22de5eca612aa964ddfec00a912e26f4656cbef890b", size = 162796, upload-time = "2026-07-07T14:3
4:12.47Z" },
    { url = "https://files.pythonhosted.org/packages/20/95/d75e82f8ce9fd323ebf059c16c9aadefb22a1ecde13b7840b35835e4886c/charset_normalizer-3.4.9-cp314
-cp314-win_arm64.whl", hash = "sha256:40a126142a56b2dfc0aacbad1de8310cbf60da7656db0e6b16eebd48e3e93519", size = 153334, upload-time = "2026-07-07T14:3
4:14.044Z" },
    { url = "https://files.pythonhosted.org/packages/00/5e/17398df3a139985ba9d11ed072531986f408c8fca952835ef1ab1820c02b/charset_normalizer-3.4.9-cp314
-cp314t-macosx_10_15_universal2.whl", hash = "sha256:609b3ba8fcc0fb5ab7af00719d0fb6ad0cb518e48e7712d12fd68f1327951198", size = 338848, upload-time = "
2026-07-07T14:34:15.688Z" },
    { url = "https://files.pythonhosted.org/packages/cd/91/7253a32e86b7e1d1239b1b36ba6dd0f021a21107ab33054b53119cc083b9/charset_normalizer-3.4.9-cp314
-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:51447e9aa2684679af07ca5021c3db526e0284347ebf4ffcec115
4c3350cfe32", size = 223022, upload-time = "2026-07-07T14:34:17.248Z" },
    { url = "https://files.pythonhosted.org/packages/cb/32/2e64bd2be10e89c61e57ebe6a93fd98ae88eb7ebe414b5121f22c96c69eb/charset_normalizer-3.4.9-cp314
-cp314t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:cc1b0fff8ead343dae06305f954eb8468ba0ec1a97881f42489d1
98e4ce3c632", size = 241590, upload-time = "2026-07-07T14:34:18.813Z" },
    { url = "https://files.pythonhosted.org/packages/3d/ef/d96ec496cfea0c21db43b0ad03891308b02388d054cc902cf0e5a1ad6a88/charset_normalizer-3.4.9-cp314
-cp314t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:fa36ec09ef71d158186bc79e359ff5fdd6e7996fe8ab638f00d6b93139b
a4fcf", size = 239584, upload-time = "2026-07-07T14:34:20.52Z" },
    { url = "https://files.pythonhosted.org/packages/d4/ce/9af95f7876194bd7a14e3dfe4a4de2e0bff02666a3910d72beafd06cc297/charset_normalizer-3.4.9-cp314
-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:df115d4d83168fdf2cae48ef1ff6d1cb4c466364e30861b37121de0f
3bf1b990", size = 230224, upload-time = "2026-07-07T14:34:22.189Z" },
    { url = "https://files.pythonhosted.org/packages/52/94/af74dde74a3996bd959c350709bfe50e297823d70a8c1cbd54b838880863/charset_normalizer-3.4.9-cp314
-cp314t-manylinux_2_31_armv7l.whl", hash = "sha256:f86c6358749bd4fda175388691e3ba8c46e24c5347d0afd20f9b7edfc9faf07d", size = 212667, upload-time = "20
26-07-07T14:34:23.857Z" },
    { url = "https://files.pythonhosted.org/packages/ee/f0/f1c4fe746c395922961b5916ed1d7d6e7d4c84851d19ed43cc89980ec953/charset_normalizer-3.4.9-cp314
-cp314t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:32286a2c8d167e897177b673176c1e3e00d4057caf5d2b64eef9a3666b03018e", size = 2
27179, upload-time = "2026-07-07T14:34:25.586Z" },
    { url = "https://files.pythonhosted.org/packages/e4/56/6c745619ac397e8871e2bcd3cea1eec86b877488f33888b3aef5c3ed506e/charset_normalizer-3.4.9-cp314
-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:83aed2c10721ddd90f68140685391b50811a880af20654c59af6b6c66c40513c", size = 225372, upload-time = "20
26-07-07T14:34:27.212Z" },
    { url = "https://files.pythonhosted.org/packages/78/ad/98aae8630ac71f16711968e38a5acfecce41b778bf2f0312851020f565a8/charset_normalizer-3.4.9-cp314
-cp314t-musllinux_1_2_armv7l.whl", hash = "sha256:cd6c3d4b783c556fa00bf540854e42f135e2f256abd29669fcd0da0f2dec79c2", size = 215222, upload-time = "202
6-07-07T14:34:28.774Z" },
    { url = "https://files.pythonhosted.org/packages/f7/40/9593d54209765207a7f11073c06494c1721e4ca4a0a426c597679bf7f91e/charset_normalizer-3.4.9-cp314
-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:ee2f2a527e3c1a6e6411eb4209642e138b544a2d72fe5d0d76daf77b24063534", size = 231958, upload-time = "202
6-07-07T14:34:30.345Z" },
    { url = "https://files.pythonhosted.org/packages/b1/27/693ee5e8a18191eb38647360c51cd505013e2bd3b366aa43fd5344c21e3c/charset_normalizer-3.4.9-cp314
-cp314t-win32.whl", hash = "sha256:0d861473f743244d349b50f850d10eb87aeb22bbdcc8e64f79273c94af5a8226", size = 155580, upload-time = "2026-07-07T14:34:3
1.884Z" },
    { url = "https://files.pythonhosted.org/packages/80/3f/bd97d3d9c613013d07cb7733d299385b41df37f0471310f5a73dc359f0b8/charset_normalizer-3.4.9-cp314
-cp314t-win_amd64.whl", hash = "sha256:9b8e0f3107e2200b76f6054de99016eac3ee6762713587b36baaa7e4bd2ae177", size = 167620, upload-time = "2026-07-07T14:
34:33.438Z" },
    { url = "https://files.pythonhosted.org/packages/3d/c6/eee9dca4439b1061f76373f06ea855678cc4a64c1c3c90b50e479edbb8eb/charset_normalizer-3.4.9-cp314
-cp314t-win_arm64.whl", hash = "sha256:19ac87f93086ce37b86e098888555c4b4bc48102279bae3350098c0ed664b501", size = 158037, upload-time = "2026-07-07T14:
34:35.018Z" },
    { url = "https://files.pythonhosted.org/packages/98/2b/f97f1c193fb855c345d678f5077d6926034db0722df74c8f057020e05a25/charset_normalizer-3.4.9-py3-n
one-any.whl", hash = "sha256:68e5f26a1ad57ded6d1cfb85331d1c1a195314756471d97758c48498bb4dcdf5", size = 64538, upload-time = "2026-07-07T14:34:56.993Z"
 },
]

[[package]]
name = "click"
version = "8.4.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "colorama", marker = "sys_platform == 'win32'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/76/d4/81420972a676e8ffea40450d8c8c92943e7218a78fe9b64359836cc9876b/click-8.4.2.tar.gz", hash
= "sha256:9a6cea6e60b17ebe0a44c5cc636d94f09bd66142c1cd7d8b4cd731c4917a15f6", size = 338000, upload-time = "2026-06-24T17:45:15.148Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/fb/e2/79c688af8b210d232694e31e59da9f6ec747bae31c3f5946e4e9b98860d5/click-8.4.2-py3-none-any.whl",
 hash = "sha256:e6f9f66136c816745b9d65817da91d61d957fb16e02e4dcd0552553c5a197b76", size = 119243, upload-time = "2026-06-24T17:45:13.73Z" },
]

[[package]]
name = "colorama"
version = "0.4.6"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/d8/53/6f443c9a4a8358a93a6792e2acffb9d9d5cb0a5cfd8802644b7b1c9a02e4/colorama-0.4.6.tar.gz", ha
sh = "sha256:08695f5cb7ed6e0531a20572697297273c47b8cae5a63ffc6d6ed5c201be6e44", size = 27697, upload-time = "2022-10-25T02:36:22.414Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d1/d6/3965ed04c63042e047cb6a3e6ed1a63a35087b6a609aa3a15ed8ac56c221/colorama-0.4.6-py2.py3-none-an
y.whl", hash = "sha256:4f1d9991f5acc0ca119f9d443620b77f9d6b33703e51011c16baf57afb285fc6", size = 25335, upload-time = "2022-10-25T02:36:20.889Z" },
]

[[package]]
name = "coverage"
version = "7.15.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/f4/45/78dbf9604ee5b3db24efbf26bed1cb58862fb40480cba821963c69348751/coverage-7.15.3.tar.gz", h
ash = "sha256:ae7ea5a4614acf399ef0483c4cb34f8f8f01df848d8fcbe7d3ce0865733f1c4d", size = 935592, upload-time = "2026-08-02T18:50:17.006Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d1/6c/bac99d9d4c6abe856e93bf3f5212982ac0bfac126dd4a042753bd53bc5af/coverage-7.15.3-cp312-cp312-ma
cosx_10_13_x86_64.whl", hash = "sha256:79a3e32e83227d83d9684459ed579769b56c369ac2d7313099b2d9e031d2e10f", size = 222499, upload-time = "2026-08-02T18:
48:15.018Z" },
    { url = "https://files.pythonhosted.org/packages/aa/bc/cb9a39b083bc1aa70586482dab25c9be20bab0ec6c155340e50d9066bb1e/coverage-7.15.3-cp312-cp312-ma
```

cosx_11_0_arm64.whl", hash = "sha256:767feb87c5886d781d0a69fafd450a20826ddab7b79bce1665deb64d21441b60", size = 222866, upload-time = "2026-08-02T18:48:16.884Z" },
    { url = "https://files.pythonhosted.org/packages/58/fb/beaa453d62000a0a5b39838bee2a137afe609a50a71f55e83c73461e513b/coverage-7.15.3-cp312-cp312-manylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:50951e37033c40548d777b8a8454a2cd622dba1136780065678dccaec307c47f", size = 254367, upload-time = "2026-08-02T18:48:18.507Z" },
    { url = "https://files.pythonhosted.org/packages/66/64/43e72500ed6815cef189f9193f29d7af4b078830337c95ea976cd0c0d427/coverage-7.15.3-cp312-cp312-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:63a4ff67364afb2cac826b8bbd78a5c50ce656a7b7137436b44d7b96a9271088", size = 257103, upload-time = "2026-08-02T18:48:20.172Z" },
    { url = "https://files.pythonhosted.org/packages/66/3a/2893e2937adfe02f45fd38e4a8a0a0d8b7a02ff9e012ac3d009bee3c4f16/coverage-7.15.3-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:6e95e42856509675fe26560310313a6117640e96f9a1e19bb3d220116a27c94c", size = 258220, upload-time = "2026-08-02T18:48:21.963Z" },
    { url = "https://files.pythonhosted.org/packages/30/b4/d5e6e2eb1a62961083734291304b1f85df72e2abe95c76eb88a7f472afd0/coverage-7.15.3-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:abad631cba27094b4631993f4c72e89ac0ca1b3a0236c7abaf8ca79aea619851", size = 260481, upload-time = "2026-08-02T18:48:23.682Z" },
    { url = "https://files.pythonhosted.org/packages/dc/c9/9b72c5c6a9798a9a12cf65f66e077cc1fdd396e61915c862688f9afe1cae/coverage-7.15.3-cp312-cp312-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:2b0807f1f051dd82a234ad6acdb6f1425baede60be1e84e862496c8cc9262ab9", size = 254749, upload-time = "2026-08-02T18:48:25.32Z" },
    { url = "https://files.pythonhosted.org/packages/92/20/e1c2f759e2dbce559ba85c40c0e4acfecc6cff4b740c294c88e41ccc6111/coverage-7.15.3-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:d8d6df7aeb5bc464040bbc9ae173d875785d3677ebc4307817997d622d74225e", size = 256138, upload-time = "2026-08-02T18:48:27.064Z" },
    { url = "https://files.pythonhosted.org/packages/a5/ab/48cc7e760f769e86ae290a125ea6e7209dfbdbbbb7ff4f5d9d1ee7a45d57/coverage-7.15.3-cp312-cp312-musllinux_1_2_i686.whl", hash = "sha256:974471c506c9f5758808b47c1ebf7949ecd0848f5c1020e78675fefe5ff46866", size = 254283, upload-time = "2026-08-02T18:48:29.082Z" },
    { url = "https://files.pythonhosted.org/packages/15/26/39529a68154f99b3a1829debd8b25eac384effeec890a293b5bbdcb49186/coverage-7.15.3-cp312-cp312-musllinux_1_2_ppc64le.whl", hash = "sha256:5cba0c9c13e35c86df7998f1afaf6b1da224a3a39e4da59bdabf60c148046dcb", size = 258352, upload-time = "2026-08-02T18:48:30.892Z" },
    { url = "https://files.pythonhosted.org/packages/91/2f/55b82aa3d8d7dd8023a56e7c5c2a70e39a3c44b3353c6cf3faec9ad51566/coverage-7.15.3-cp312-cp312-musllinux_1_2_riscv64.whl", hash = "sha256:4d608dc36a364dce33acbf4fc3a50f9d2054c945f233bb0a2cdb4b90bfa17646", size = 253852, upload-time = "2026-08-02T18:48:32.934Z" },
    { url = "https://files.pythonhosted.org/packages/6a/6d/839f4045124cd3518ecf2c58967e58a911202834e7c5a03cfdf2ab0b29f6/coverage-7.15.3-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:2395869280554a1941da904423c12660c39f721315e1c02d076a7fe0971382f0", size = 255725, upload-time = "2026-08-02T18:48:34.848Z" },
    { url = "https://files.pythonhosted.org/packages/75/21/d25e3e2a9e327798078c877f469dfb6def860bf6e25036529046227d3e15/coverage-7.15.3-cp312-cp312-win32.whl", hash = "sha256:24f3b21840c3eb76cef3cc70b2bf6649010c64471a84a446538a39306e1ba04d", size = 224566, upload-time = "2026-08-02T18:48:36.661Z" },
    { url = "https://files.pythonhosted.org/packages/b1/0f/df90cc1e8d095ce263968a93e04829821b2afb31ac2752c06a2e0a8e3c13/coverage-7.15.3-cp312-cp312-win_amd64.whl", hash = "sha256:fa7b17902c3c1dd8a7adb52679b7f6340bba08443d710c8838e04db8cf62be2a", size = 225098, upload-time = "2026-08-02T18:48:38.941Z" },
    { url = "https://files.pythonhosted.org/packages/65/c7/ec49e43c58967a07163e2d1c6bbd58112b825b2772ab66784afd6a5400ba/coverage-7.15.3-cp312-cp312-win_arm64.whl", hash = "sha256:fcbe83fb7258eacd293bf5322d88807acb35ed12a5cfa99dd8215c083e3b0235", size = 224485, upload-time = "2026-08-02T18:48:40.682Z" },
    { url = "https://files.pythonhosted.org/packages/68/6e/62ae61e1fc434956bec38ed1d5b1c494f58cf579dbd998e77abffe7b3e6b/coverage-7.15.3-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:1182eed05674c63d40951fae27c43e822749f04d25f75df64c2e4fa3168678de", size = 222522, upload-time = "2026-08-02T18:48:42.476Z" },
    { url = "https://files.pythonhosted.org/packages/13/ff/c74c673d81e0e77b6608c3d21331e3db42e30daeb3c8a0a8860d4c9e2e14/coverage-7.15.3-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:c0c4b0d7c4cd56e470d0c9d8441f42e8a96cdfd95050fec027f1d4dd9f11006c", size = 222894, upload-time = "2026-08-02T18:48:44.274Z" },
    { url = "https://files.pythonhosted.org/packages/a1/91/ccb30f5ffafd7d69d0b18e5162f9b711a5654e807b7b0c13497f0826b33f/coverage-7.15.3-cp313-cp313-manylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:5c9fce9f4998b0d50a753da765b9215a14decc7863822c89d72da7a89ca625b3", size = 253890, upload-time = "2026-08-02T18:48:46.097Z" },
    { url = "https://files.pythonhosted.org/packages/29/c6/e92a66cda49a2751b09826d51258f199b92aa0cb005bc5f34e9729a52a9c/coverage-7.15.3-cp313-cp313-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:7a47e2a0a0ace9241e70ee00e44520f88b843094603dd54303f1bafecd929c30", size = 256484, upload-time = "2026-08-02T18:48:47.846Z" },
    { url = "https://files.pythonhosted.org/packages/96/7a/730929164b457cf25cf76c23898b90f9039a104a647890801b6586797b14/coverage-7.15.3-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:95bad94f83807ae60ed76f3ac012f69b2605ac9ea81bee959a5a483f7fa09c10", size = 257723, upload-time = "2026-08-02T18:48:49.664Z" },
    { url = "https://files.pythonhosted.org/packages/9e/be/04cb5672cb19f5c389eda81ba22d89807699a949653d3625b0e0fda169da/coverage-7.15.3-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:228e172a76c428bb17d1ab78a2ff188990b0597e5dbd291f52a4edf7412de049", size = 259854, upload-time = "2026-08-02T18:48:51.413Z" },
    { url = "https://files.pythonhosted.org/packages/96/25/5e7fd6af39f6507071455944b8906dd1fe5b7b6bffb6a163ceb20afa0d13/coverage-7.15.3-cp313-cp313-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:cea9fb33887c99349996266f1fd60abe5af3577a90633392001d27ef46b4b66e", size = 254085, upload-time = "2026-08-02T18:48:53.158Z" },
    { url = "https://files.pythonhosted.org/packages/23/c8/55e58a853f1e61163a6e755897bd14a059d78411e86560f39d9951c019b5/coverage-7.15.3-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:81760de3155d7f52c21860c4046628dc6bed182f72e3c028e2b4fd46f65aa040", size = 255850, upload-time = "2026-08-02T18:48:55.031Z" },
    { url = "https://files.pythonhosted.org/packages/be/74/8bcec66dbcf3d22bea2a0b2b77ee2fa6f766a647d0023d4eabbc4f2b2756/coverage-7.15.3-cp313-cp313-musllinux_1_2_i686.whl", hash = "sha256:b47ea0a1d3a3d089826c6cbfad8429d7d8872e28e86baa95ddef330f6875da21", size = 253818, upload-time = "2026-08-02T18:48:57.163Z" },
    { url = "https://files.pythonhosted.org/packages/ce/06/450b673fdfece0997b4e16a31d6bde6b18889c578f1013ddd34c962ac6f9/coverage-7.15.3-cp313-cp313-musllinux_1_2_ppc64le.whl", hash = "sha256:5459ba486b2a5d58a6c05254779ecdf525e7f20174d0210ceda75ba40fdb8f2c", size = 257973, upload-time = "2026-08-02T18:48:59.098Z" },
    { url = "https://files.pythonhosted.org/packages/56/fd/3ec7409aec0ddc943132452b65672f065f043b844f1830e1fe173c98b3ab/coverage-7.15.3-cp313-cp313-musllinux_1_2_riscv64.whl", hash = "sha256:c59209f80a08dbfcdd5109a80dc623cd3b9d22895c85757d34f57a6e6e95570f", size = 253638, upload-time = "2026-08-02T18:49:01.199Z" },
    { url = "https://files.pythonhosted.org/packages/75/20/30a8dabb194123631c93f860fdd86401ad405d56cfb1841873afbfe4e92b/coverage-7.15.3-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:f863856c1779d4a5bb6a94698a2f9073e09c6706501f76f3e7780e72df97d21c", size = 255407, upload-time = "2026-08-02T18:49:03.143Z" },
    { url = "https://files.pythonhosted.org/packages/13/4d/e14365b1953b43653341412f9088b0d752614c626a73a705ff9af400f3a3/coverage-7.15.3-cp313-cp313-win32.whl", hash = "sha256:00cbdc5e322927dc30c5e42b863819b1bb867cc66f26ab5372c585850876ab93", size = 224575, upload-time = "2026-08-02T18:49:05.011Z" },
    { url = "https://files.pythonhosted.org/packages/1c/64/88f762ea80de2070207246faef514513be874486b2773528f2cc2b4b515c/coverage-7.15.3-cp313-cp313-win_amd64.whl", hash = "sha256:835528518a1d823cf336740324b2f335f7c01e609e74abcb5d5163b3e66661e3", size = 225116, upload-time = "2026-08-02T18:49:06.894Z" },
    { url = "https://files.pythonhosted.org/packages/ab/66/03c34c53a319f522554cd29d4f2e16c5eab61aa4cdcf55753129fd7d926c/coverage-7.15.3-cp313-cp313-win_arm64.whl", hash = "sha256:0d2e1f2cbbf36b842f3e2aff8d118c60d677adb498bc6c7fa9c6838738f82767", size = 224509, upload-time = "2026-08-02T18:49:09.129Z" },
    { url = "https://files.pythonhosted.org/packages/35/6f/8c2dc014357618b3226c90f731b8282766c3685786f422558991dc49fbf2/coverage-7.15.3-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:1e3bb08ad574bd9fb6a991f645728f70d333c1c1958dd5fcde65e24cb862813d", size = 222571, upload-time = "2026-08-02T18:49:11.242Z" },
    { url = "https://files.pythonhosted.org/packages/07/50/d867c7ceae9d56b7e74ee61ea834f1aa4f9a1e1c7f0ce39393ba573b1c12/coverage-7.15.3-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:9e5860eaff02a0b7f1b73304bdf846596ee62ab3a78d25c68044ebf684cb1fef", size = 222902, upload-time = "2026-08-02T18:49:13.448Z" },
    { url = "https://files.pythonhosted.org/packages/62/77/4f6dfc490c5f2bcacb2d296d9aa4d1e128c43b48e94ad313fec7f49f09ad/coverage-7.15.3-cp314-cp314-manylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:60874e5bd67f0b1bdbe42ab42c7bafa66a6fb8de88721af6df3f7a02713960cd", size = 253947, upload-time = "2026-08-02T18:49:15.304Z" },
    { url = "https://files.pythonhosted.org/packages/16/8a/6777f192af264165103e2a3d3768dbadb9894a0a2359a16877141d9ae8f5/coverage-7.15.3-cp314-cp314-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:f9147be876e9d83765e0b82176674dc248a6b9283e25e01e7462611b97e9b731", size = 256452, upload-time = "2026-08-02T18:49:17.801Z" },
    { url = "https://files.pythonhosted.org/packages/7d/7b/3d7ac46a0234bc684f41ee42be95e29b2b6525695adb04083609d5ac2149/coverage-7.15.3-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:61a01f8c3804760fcc5a3d31c4f3cab792d660d44e17bf7adeaf0ea51e07821e", size = 257798, upload-time = "2026-08-02T18:49:19.878Z" },
    { url = "https://files.pythonhosted.org/packages/ff/1e/c6ee59c29afcb5fdb35f936381340d1a06429a07c48f20e809646647acbe/coverage-7.15.3-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:95bf3e7f26f792e25eb185f85a5a659d48479265176dcfe22b6f334fd0081b5c", size = 260112, upload-time = "2026-08-02T18:49:21.858Z" },
    { url = "https://files.pythonhosted.org/packages/c1/e1/e8ea39a46e89e3a143312ee5f80336e992e3ae8fe44bf9c76b83fefeed42/coverage-7.15.3-cp314-cp314-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:44c41eff9e413fed8740eca75d5438ebeb9d3e45e7cd37c67329213e7a72c764", size = 253944, upload-time = "2026-08-02T18:49:23.926Z" },

```
    { url = "https://files.pythonhosted.org/packages/95/67/31ab5f6a37fd887d1386f81f0da9306851ad2264e9baaa9c7f606e0b3e17/coverage-7.15.3-cp314-cp314-mu
sllinux_1_2_aarch64.whl", hash = "sha256:54146bafb61f3ba9895b43af0dd17eba01561d586d44ce84ea221b0cbbee5a9e", size = 255805, upload-time = "2026-08-02T1
8:49:25.973Z" },
    { url = "https://files.pythonhosted.org/packages/fb/6a/ee505a80c8fd89620fb337c0596daecff87f33171fbb4ee3015fc3d7331f/coverage-7.15.3-cp314-cp314-mu
sllinux_1_2_i686.whl", hash = "sha256:af000dd1bb859ff8066fda4c79512ff938c798116540307226b373099c7b151f", size = 253769, upload-time = "2026-08-02T18:4
9:27.883Z" },
    { url = "https://files.pythonhosted.org/packages/b0/41/6ab0f81c9e89660230d8f3f581d4732e5ddb75a885b0a5dfc73d315dc94f/coverage-7.15.3-cp314-cp314-mu
sllinux_1_2_ppc64le.whl", hash = "sha256:a1b82490577f3889950b5a04f18712aef0207243e0749d60fe28c3c73ebfd5fd", size = 258045, upload-time = "2026-08-02T1
8:49:30.201Z" },
    { url = "https://files.pythonhosted.org/packages/bc/62/c995e91cae28cf31d6defab3bfb553dda5ac83ac7381b0f2b121264c307a/coverage-7.15.3-cp314-cp314-mu
sllinux_1_2_riscv64.whl", hash = "sha256:c4fc90a60154c3e4b8a2dc206d6dbe852f1c235c249e0dc0cef909d032c9591a", size = 253587, upload-time = "2026-08-02T1
8:49:32.349Z" },
    { url = "https://files.pythonhosted.org/packages/84/df/f2049980f82d6890321f2065f9e66216eabbf4b2001815db958bc543f40a/coverage-7.15.3-cp314-cp314-mu
sllinux_1_2_x86_64.whl", hash = "sha256:f25bb884814a892948b4c20394db3f2364dd452d9492736479e7a493e63b0eb6", size = 255243, upload-time = "2026-08-02T18
:49:34.324Z" },
    { url = "https://files.pythonhosted.org/packages/1d/82/2c841b67a978c0eb9c3707630b68f93f9e7585d78bb906bc8823ec6b07a5/coverage-7.15.3-cp314-cp314-wi
n32.whl", hash = "sha256:722dbf8e7828fbcfe0dc8586167dc0a5ce85ad6ea171dbb21ed3f8d6581d3cb8", size = 224759, upload-time = "2026-08-02T18:49:36.326Z" },
    { url = "https://files.pythonhosted.org/packages/b3/78/5c93ec43784fd3e404ca23cd0584ae24bc1732de4a3fc194b68c3be88db0/coverage-7.15.3-cp314-cp314-wi
n_amd64.whl", hash = "sha256:64d0845f9c3ed47302bed265c15ab4dbb64aa4ec1490839b8e328f4e7fa914d2", size = 225246, upload-time = "2026-08-02T18:49:38.366Z
" },
    { url = "https://files.pythonhosted.org/packages/9d/77/813a054371f3b018cc63c6bdb46a3c35d5e95d4e3ed4f1449d4196106db5/coverage-7.15.3-cp314-cp314-wi
n_arm64.whl", hash = "sha256:69bc14684f8fbbee9f9dbaa4fe79719b0da9725fc37956785c06ec365acf6926", size = 224673, upload-time = "2026-08-02T18:49:40.552Z
" },
    { url = "https://files.pythonhosted.org/packages/8f/63/8c9f36cc71178d26db930baa03a4494abcc516d8d41bf820d0d85ef1d80b/coverage-7.15.3-cp314-cp314t-m
acosx_10_15_x86_64.whl", hash = "sha256:f92df943c24b96cb215ca26b4f6a2283e63c5db80f1635aceea7fff11311917b", size = 223298, upload-time = "2026-08-02T18
:49:42.634Z" },
    { url = "https://files.pythonhosted.org/packages/54/66/211f24d058ce9f56ebf1420d55b7574fdae924f6da3836f83c8bd4793e38/coverage-7.15.3-cp314-cp314t-m
acosx_11_0_arm64.whl", hash = "sha256:66591c46bdd2971d3ae2bc503a5f0459c2edcaf6b7e045b292000cc95bc6cb95", size = 223568, upload-time = "2026-08-02T18:4
9:44.706Z" },
    { url = "https://files.pythonhosted.org/packages/dd/bb/9c2ad5574a0d6420a96c6cade4f8a683931b9e79fe609f8924d7b6964616/coverage-7.15.3-cp314-cp314t-m
anylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:caa64458b81b18bfc67cdf1f6dc02b23e3edc672f2f8e11771fad75865415a43", size = 2
64932, upload-time = "2026-08-02T18:49:47.153Z" },
    { url = "https://files.pythonhosted.org/packages/ba/91/938c39e77bdd5a0a440412f975609ce3702dabbda6ac715719d93ca45a7b/coverage-7.15.3-cp314-cp314t-m
anylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:447f5421ccf5475956cf516d4ca1d575f487947b6f4e11f9d80c6aefe24b3dc8", si
ze = 267052, upload-time = "2026-08-02T18:49:49.324Z" },
    { url = "https://files.pythonhosted.org/packages/b0/a3/7b431a98af35d9cc6394e54cde9435b33b8591672fbece6a4931267d7a8e/coverage-7.15.3-cp314-cp314t-m
anylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:7a0c77ef8cd483a4987a5d12d1d9d5f7ee598dfdc6c0844417d847e5768dc7
79", size = 269473, upload-time = "2026-08-02T18:49:51.599Z" },
    { url = "https://files.pythonhosted.org/packages/32/58/dbc9951dce46be47a732823a1c571f62bcabdd54a68d8c281489a1a55cfb/coverage-7.15.3-cp314-cp314t-m
anylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:0b273f4ff657446a06c2d85bf80e134fa869a92852ba5f87854a70e1fb44da
77", size = 270591, upload-time = "2026-08-02T18:49:53.865Z" },
    { url = "https://files.pythonhosted.org/packages/71/bd/1d610772c7c0889bfe477a59c46ee66ea53e271f3f06951e9d55b317f7c6/coverage-7.15.3-cp314-cp314t-m
anylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:daea8c4fafa22488600405be2c2be525a9406fba3fc0a83acc726db3e14e2005", size = 264007, up
load-time = "2026-08-02T18:49:55.875Z" },
    { url = "https://files.pythonhosted.org/packages/69/97/852eb3dcdba156b1a9078503f098499916bf889f964b61ad4a08223ac169/coverage-7.15.3-cp314-cp314t-m
usllinux_1_2_aarch64.whl", hash = "sha256:93ff57c530f3fa7aa69f92fb9b8892b8aa82712aa970842f4abf28657f42fb57", size = 266926, upload-time = "2026-08-02T
18:49:57.944Z" },
    { url = "https://files.pythonhosted.org/packages/52/f8/b72cd238757fba2b587fc7dee047efe6e10b0c18343509faaaf502dd4680/coverage-7.15.3-cp314-cp314t-m
usllinux_1_2_i686.whl", hash = "sha256:4df21bef8b800eebda9018f53d49c9ace3aeb0090c850139b27923aafcb83e91", size = 264529, upload-time = "2026-08-02T18:
50:00.035Z" },
    { url = "https://files.pythonhosted.org/packages/b8/0a/6c52ec4b7fb007cb6433d1fcfda4080cb15d75ad37ef9c31025f3427293e/coverage-7.15.3-cp314-cp314t-m
usllinux_1_2_ppc64le.whl", hash = "sha256:db567b02685f26034adcbd85055f80d12cdf02111b8ed00886093d98b2874ce2", size = 268263, upload-time = "2026-08-02T
18:50:02.161Z" },
    { url = "https://files.pythonhosted.org/packages/c0/4e/f1f9aa3efd109a04353563a43fb5155340c1fdcdeaa6296ebed3b6f510ea/coverage-7.15.3-cp314-cp314t-m
usllinux_1_2_riscv64.whl", hash = "sha256:5318dd51b8600b947e058cf5a4fe54d183d9d13c49b97b64ca7be05a34df9bef", size = 263377, upload-time = "2026-08-02T
18:50:04.243Z" },
    { url = "https://files.pythonhosted.org/packages/dd/fb/6b268a0b2728ef1c379ad656b899274477a5f6bed1bf6765b4b387fb0601/coverage-7.15.3-cp314-cp314t-m
usllinux_1_2_x86_64.whl", hash = "sha256:c995bfa383c54704839b6c4c2627a1c00895597ada0e5e8190c81d8bd620555c", size = 265688, upload-time = "2026-08-02T1
8:50:06.428Z" },
    { url = "https://files.pythonhosted.org/packages/29/54/1a3ea96e5d5e7cd41dc432597bfc60692910e635d05e1cc25a8ccc243581/coverage-7.15.3-cp314-cp314t-w
in32.whl", hash = "sha256:6433fafb8da0e1d02eb53411e0ecdadb6b88f0224fdc23317e703c0e88937d42", size = 225066, upload-time = "2026-08-02T18:50:08.533Z" }
,
    { url = "https://files.pythonhosted.org/packages/31/9d/a7b0d9afd18ed5274dd00651a78e7810a931c70d94b79996f150bec1a30f/coverage-7.15.3-cp314-cp314t-w
in_amd64.whl", hash = "sha256:fe578952b1b29fe8c777f43f241d49efac4b56724a3434f5d22ebe3c208df429", size = 225897, upload-time = "2026-08-02T18:50:10.572
Z" },
    { url = "https://files.pythonhosted.org/packages/ca/11/34c5ae40b945e69aa72b87dc268135b7049905f3824af573b7073acbb946/coverage-7.15.3-cp314-cp314t-w
in_arm64.whl", hash = "sha256:d2e1acb7aee29dfa8f3e48c23f36670898baca1209d9bdd3985a50c7f982165e", size = 225212, upload-time = "2026-08-02T18:50:12.63Z
" },
    { url = "https://files.pythonhosted.org/packages/37/e7/7069b3d6c018917f49ba2e1c5fb910e498c7fefa3a1b78cb1b79e61ff45d/coverage-7.15.3-py3-none-any.w
hl", hash = "sha256:da78fa6fc7dafe4212839173133ee85afcf42c5cd5f3e47fa7c1c210453b445e", size = 214297, upload-time = "2026-08-02T18:50:14.709Z" },
]

[[package]]
name = "cryptography"
version = "50.0.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "cffi", marker = "platform_python_implementation != 'PyPy'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/de/41/6cbdcf9142d00fe82836fbb51e503e58088575cf7a0fe1dbff6695bf0840/cryptography-50.0.0.tar.gz
", hash = "sha256:eeac2acb5a20ed25e0ad6d1df9891a520b78b404266b6d11778f25d5d691a6c9", size = 880201, upload-time = "2026-07-31T14:25:10.11Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/c5/5c/59086b4aac5e879d38ddbcf74e4be7ade89cebc3eb199a55da998c3bb46a/cryptography-50.0.0-cp311-abi3
-macosx_11_0_arm64.whl", hash = "sha256:031e2d5dd4bb9caa3ca9c82e5a197fd8ae680232cee62603d1a813f3f07e3d03", size = 4001252, upload-time = "2026-07-31T1
4:23:33.331Z" },
    { url = "https://files.pythonhosted.org/packages/57/ef/8f2df13c7216bcad3e1c74e07f6e193d93e998e114f524a53877c9af27ad/cryptography-50.0.0-cp311-abi3
-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:fd9192b7b70c573d7f214eb1ae35e00d359f6f5e4b27c7e21e30de1fc6204645", size = 4719554,
upload-time = "2026-07-31T14:23:35.611Z" },
    { url = "https://files.pythonhosted.org/packages/d9/41/029086c34d91052fc3b88bcc8056f709a7c915c7a23b235a54eb800b1c97/cryptography-50.0.0-cp311-abi3
-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:06a32a980526a6ab9a4b9bf8f7385800791e2bb960903cb6b530e4817509a3b7", size = 4702130, up
load-time = "2026-07-31T14:23:37.635Z" },
    { url = "https://files.pythonhosted.org/packages/7d/ff/b6ce0954962e7f7b969f850a883744197bb3910bdfd7b6da162eab7d9f68/cryptography-50.0.0-cp311-abi3
-manylinux_2_28_aarch64.whl", hash = "sha256:a1b30560f2acc95aa8b2e06e716a13dbfc97314747b80d9707e307f77b40d6b3", size = 4725244, upload-time = "2026-07
-31T14:23:39.471Z" },
    { url = "https://files.pythonhosted.org/packages/06/1e/63a1027cb7fec360a182208e1b7767d5aa1fe57be3d6aa856e69a321edc0/cryptography-50.0.0-cp311-abi3
-manylinux_2_28_ppc64le.whl", hash = "sha256:8d89f3976b10b4ce31118de72329025f70d2c6ead14a8217c5514dd2c6d5a78f", size = 5342265, upload-time = "2026-07
-31T14:23:41.286Z" },
    { url = "https://files.pythonhosted.org/packages/6b/72/a1116d683a6d7ece94590013882515de087edf9ef0e6292aae615a44df73/cryptography-50.0.0-cp311-abi3
-manylinux_2_28_x86_64.whl", hash = "sha256:b42a28c1844fd9de8f3f7d540e36b66f3a9c83fceac7170ebc7a6a19edd9dcae", size = 4734609, upload-time = "2026-07-
31T14:23:43.139Z" },
    { url = "https://files.pythonhosted.org/packages/15/37/36a9c479bbe49acea2636c7fd3360d20f7b7e079c300352011c44850b181/cryptography-50.0.0-cp311-abi3
-manylinux_2_31_armv7l.whl", hash = "sha256:900131fafd8aead39ac7dd3a7e833be754c17a95cfd91221636949fe4eb0aa8a", size = 4356517, upload-time = "2026-07-
31T14:23:44.939Z" },
    { url = "https://files.pythonhosted.org/packages/32/98/8a151d64367204cbc63ec65d37502f1d9c53cf4bfc6ec3c532614dbec60d/cryptography-50.0.0-cp311-abi3
-manylinux_2_34_aarch64.whl", hash = "sha256:07949c449a1abcf60d1ee6e88956d89404c7df3c8258f46589e912988e551987", size = 4724529, upload-time = "2026-07
```

```
-31T14:23:46.93Z" },
    { url = "https://files.pythonhosted.org/packages/22/f6/ec13b470172126464a86bf54d2294a46d29837fc51ba3e45d4047946fb5e/cryptography-50.0.0-cp311-abi3
-manylinux_2_34_ppc64le.whl", hash = "sha256:f89831ef99dd7dd169ab06d63a831adb9e20a87aac6d380266bbda5823349169", size = 5299852, upload-time = "2026-07
-31T14:23:48.851Z" },
    { url = "https://files.pythonhosted.org/packages/da/3a/f05e32c99d440c9bb891ea0e36c9091891e36be5a9a87ab2ee6ea20729f6/cryptography-50.0.0-cp311-abi3
-manylinux_2_34_x86_64.whl", hash = "sha256:82148ec5bddac30b51a5b3c1945075f896fa022cb93f8e4a01e9f6ee95292c5f", size = 4734462, upload-time = "2026-07-
31T14:23:50.861Z" },
    { url = "https://files.pythonhosted.org/packages/ca/dc/bd72b26be8953f80625f63151efd38eee71c76ca6cf591c08ff34615a79e/cryptography-50.0.0-cp311-abi3
-musllinux_1_2_aarch64.whl", hash = "sha256:1489e263a8048bb8b6a8bac662eb2d402ea5d2b7b4699b72f385f1e2772db105", size = 4852708, upload-time = "2026-07-
31T14:23:52.715Z" },
    { url = "https://files.pythonhosted.org/packages/27/20/c930314a2ab476d15dec966ec87e2e9637bb02b06106b12c0396c57bb603/cryptography-50.0.0-cp311-abi3
-musllinux_1_2_x86_64.whl", hash = "sha256:7cec5b856506da6defb290f30c9ee687d5f5e8cb0bd3f6459dde43b0b4fa40ef", size = 5004179, upload-time = "2026-07-3
1T14:23:54.887Z" },
    { url = "https://files.pythonhosted.org/packages/32/2e/c9db68a0c4bfa28e310707527c0ee3a2bd254104d2e02e68f368e197aa4c/cryptography-50.0.0-cp311-abi3
-win_amd64.whl", hash = "sha256:bd1c592e4d5974f0d08d4888e432157adba757c66da0246918e43677fafa2d30", size = 3840395, upload-time = "2026-07-31T14:23:56.
677Z" },
    { url = "https://files.pythonhosted.org/packages/c3/fb/951032a3bf22a5697c83183fb6294a4843772947a70e616c57b3ff5f522e/cryptography-50.0.0-cp314-cp31
4t-macosx_11_0_arm64.whl", hash = "sha256:49e7d93abdbd2990caced757e5fade25302f719c3c8fb6e6fff2dde98999fc41", size = 3989258, upload-time = "2026-07-31
T14:23:58.881Z" },
    { url = "https://files.pythonhosted.org/packages/d4/67/91eb047e69c5e845f2f14b8a2e4a1aab0f283cb885531e9e22c8adb176bc/cryptography-50.0.0-cp314-cp31
4t-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:19736989797678c6af1e55cd49055cdbcb55d8f6b5583ac5335f933aba9101dc", size = 4700648
, upload-time = "2026-07-31T14:24:00.702Z" },
    { url = "https://files.pythonhosted.org/packages/30/82/85f0f7425c856b9f96459411eb12e74ef72df9caf6f8f15bf23a33ff131f/cryptography-50.0.0-cp314-cp31
4t-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:80b63928fa35083b33966ce1efb70e5b9607181e49dcd1c22c8c005e319f667f", size = 4682442,
upload-time = "2026-07-31T14:24:02.538Z" },
    { url = "https://files.pythonhosted.org/packages/1a/28/b555a365adff1cca2fbe7b9e487d68a40de6bc67ff2cb587473eb43de0e7/cryptography-50.0.0-cp314-cp31
4t-manylinux_2_28_aarch64.whl", hash = "sha256:d58c3db7cd6eed54e6c06744db55456b65ebd7492ddeae9c1e93cfca7aa857d3", size = 4707596, upload-time = "2026-
07-31T14:24:04.394Z" },
    { url = "https://files.pythonhosted.org/packages/72/d8/f52538140cc719df62a01cf87d1c7142318d235817109d6f4054d7c352d6/cryptography-50.0.0-cp314-cp31
4t-manylinux_2_28_ppc64le.whl", hash = "sha256:df2a58a472f332225671c35b0a830208b86d004f82baa8530fa3782c85646533", size = 5314552, upload-time = "2026-
07-31T14:24:06.31Z" },
    { url = "https://files.pythonhosted.org/packages/38/14/6120e5bd7c5aa022ad15424ba4d5c5269d0d9448ed4d55e492ea91e3c1c4/cryptography-50.0.0-cp314-cp31
4t-manylinux_2_28_x86_64.whl", hash = "sha256:11b74db56cdbe3cdee6e3f6982ecb70334fa10dce99ed58bf7894aaaa3b2a037", size = 4717113, upload-time = "2026-0
7-31T14:24:08.349Z" },
    { url = "https://files.pythonhosted.org/packages/fa/71/190bf38c3ee2e0f8efc9860ae100c9df4169742eef274b91e7aa1cb133b9/cryptography-50.0.0-cp314-cp31
4t-manylinux_2_31_armv7l.whl", hash = "sha256:f59e38625469987d7ef6d495323c55e7db6c212eaf6112267e0d3b565a2e9c9f", size = 4338580, upload-time = "2026-0
7-31T14:24:10.227Z" },
    { url = "https://files.pythonhosted.org/packages/3a/63/504ccfbbe61fd8aa983f7f146399cdf034c72c2fc55f5b2dfdcdcdb20c99/cryptography-50.0.0-cp314-cp31
4t-manylinux_2_34_aarch64.whl", hash = "sha256:ecfed7367f965a0328cfbdd70da860f15441f002f613185668c6e6ebf5a0ac11", size = 4707038, upload-time = "2026-
07-31T14:24:12.169Z" },
    { url = "https://files.pythonhosted.org/packages/01/77/2cf79bbfc4d12ca106437a6e170d6aaa01a373e93093118aaaef0e801bd4/cryptography-50.0.0-cp314-cp31
4t-manylinux_2_34_ppc64le.whl", hash = "sha256:9aa87839c383bdbab6ef865787a1fb877af8dd03464c4400322726feaaadfc6d", size = 5273110, upload-time = "2026-
07-31T14:24:14.38Z" },
    { url = "https://files.pythonhosted.org/packages/e5/45/8aae2972c520145377ea3559a605a899bebe227bf070b33cdb445929a9b9/cryptography-50.0.0-cp314-cp31
4t-manylinux_2_34_x86_64.whl", hash = "sha256:6ba6a53445bd3cfa809ef3ef5f1589aa6ba08784a1d962bf47d0940e871dab1c", size = 4716439, upload-time = "2026-0
7-31T14:24:16.415Z" },
    { url = "https://files.pythonhosted.org/packages/7b/20/4fe50b619a48c2525cc46e2dbc1ac490708d704be5d467bdaac6dc955682/cryptography-50.0.0-cp314-cp31
4t-musllinux_1_2_aarch64.whl", hash = "sha256:3f5735ffe4996d28b809371756219f5354864902a3b9e7c0b9ee87041209fc9c", size = 4837383, upload-time = "2026-0
7-31T14:24:18.553Z" },
    { url = "https://files.pythonhosted.org/packages/92/91/3a31366e183343d3703f8995c095f5734676bd6938118047e50fcf279eb4/cryptography-50.0.0-cp314-cp31
4t-musllinux_1_2_x86_64.whl", hash = "sha256:1b4a266766514614f8aa60416e71f2fc6e575d36e7bdc90f644fadb2f4b75b95", size = 4985772, upload-time = "2026-07
-31T14:24:20.385Z" },
    { url = "https://files.pythonhosted.org/packages/74/9a/02ffe35b2853d121689871eb5dce862092562b3a1ed5cc98f1aaed441506/cryptography-50.0.0-cp314-cp31
4t-win_amd64.whl", hash = "sha256:12b9c6996425c76ea6c457ace4f3073e715b8c545add07cd1a8f3a4f90691269", size = 3816291, upload-time = "2026-07-31T14:24:2
2.125Z" },
    { url = "https://files.pythonhosted.org/packages/03/37/73d005be173aff344af30e9fd2a576575cb2391a7101d9cd3842e1fa8cce/cryptography-50.0.0-cp39-abi3-
macosx_11_0_arm64.whl", hash = "sha256:ccdc4a71a4dabae05de219404f9f4abc38e3b58422177ff93d0da05967dafa07", size = 4036009, upload-time = "2026-07-31T14
:24:24.122Z" },
    { url = "https://files.pythonhosted.org/packages/ff/c6/7a6202a534e32103a285b7834a120869557fe198d51d7cfe59754c8bda9c/cryptography-50.0.0-cp39-abi3-
manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:910e1d2668e7de9648f2bcee30e180db2a6b15c30f887d7c4c93ddf96e3992e3", size = 4745252, u
pload-time = "2026-07-31T14:24:26.118Z" },
    { url = "https://files.pythonhosted.org/packages/85/4f/0fa8c2f4428198f15d9ff8d63400e27afbf94ce833f6108da1eb3753f945/cryptography-50.0.0-cp39-abi3-
manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:a91296cb61e8df6f86d0c19cc4068228da256bf59bf86049fbd821084565327f", size = 4728939, upl
oad-time = "2026-07-31T14:24:27.994Z" },
    { url = "https://files.pythonhosted.org/packages/d1/63/54dd723490ba2dc09b299682c10b38db38f159728bcaae8c591b8af2f22d/cryptography-50.0.0-cp39-abi3-
manylinux_2_28_aarch64.whl", hash = "sha256:e722f16708d854fe924790e051061f6704a472c3bac347b6fd88033ea8dd0dc5", size = 4748483, upload-time = "2026-07-
31T14:24:30.254Z" },
    { url = "https://files.pythonhosted.org/packages/1d/dd/7c77d26285cc7f6991efce64a0f5b4f9383bfa5dd8c5033003eaf7db4cdb/cryptography-50.0.0-cp39-abi3-
manylinux_2_28_ppc64le.whl", hash = "sha256:d764dcf130c428ef66786f866dd750f53182bc608813489915e9fc106bb0c82f", size = 5367599, upload-time = "2026-07-
31T14:24:32.457Z" },
    { url = "https://files.pythonhosted.org/packages/46/c9/f60aed34c013f317f92817b6c171c2d22a78270fa41109bd4b08af26b194/cryptography-50.0.0-cp39-abi3-
manylinux_2_28_x86_64.whl", hash = "sha256:105110f43a471dbd0060b9c9516cb8a6a79233631a04cc2ba16f28323ac6e025", size = 4762647, upload-time = "2026-07-3
1T14:24:34.599Z" },
    { url = "https://files.pythonhosted.org/packages/be/f3/f9a0173b139372c3a48ed98154b45cc6b9de17c789d5ab552e621c293609/cryptography-50.0.0-cp39-abi3-
manylinux_2_31_armv7l.whl", hash = "sha256:828743d939e9629bc267b8e2d08d8bb67cd4319c771a33d4b18b22dd8fb7440a", size = 4385197, upload-time = "2026-07-3
1T14:24:36.647Z" },
    { url = "https://files.pythonhosted.org/packages/d8/36/83bb81f6e569bc38e1e4a7bc80f29b46bb9601920bc455fc8e888f5d5742/cryptography-50.0.0-cp39-abi3-
manylinux_2_34_aarch64.whl", hash = "sha256:2a8183b489dc1f7f80f135780fadc1108f14b31b8a40411c7a5b17425f65f28b", size = 4748095, upload-time = "2026-07-
31T14:24:39.493Z" },
    { url = "https://files.pythonhosted.org/packages/6b/16/d3008eff98c764979865834c3d386d4fd041b5f52e7f34fc29ac1a5eb515/cryptography-50.0.0-cp39-abi3-
manylinux_2_34_ppc64le.whl", hash = "sha256:6e7d61120573a7f2cd94cc095f9e81f6967c61ccdf194285aa143ecec8e0b708", size = 5325948, upload-time = "2026-07-
31T14:24:41.556Z" },
    { url = "https://files.pythonhosted.org/packages/9c/f8/d97f9603efda3888187bfdb893f26c41be4735c10631d05d284ee6b047c4/cryptography-50.0.0-cp39-abi3-
manylinux_2_34_x86_64.whl", hash = "sha256:37fdb0d0111f1e2ff07139dfb79f1b49531f8e213c46f1163dd7642979b58c47", size = 4762400, upload-time = "2026-07-3
1T14:24:43.636Z" },
    { url = "https://files.pythonhosted.org/packages/64/a2/4615c8f7d81a00b1d6e6afe19f694e1543582349fb5f4076f6cb5dc36485/cryptography-50.0.0-cp39-abi3-
musllinux_1_2_aarch64.whl", hash = "sha256:c87f62a3d3b9888ed0fdde100ec06aa61ca9cd44bad9057d1dff9a516b5f5bb9", size = 4878208, upload-time = "2026-07-3
1T14:24:45.522Z" },
    { url = "https://files.pythonhosted.org/packages/d2/1a/efcfb02f91407149a0dacffffab791f7e19bf6385f63b3666dc8b5e5c9c8/cryptography-50.0.0-cp39-abi3-
musllinux_1_2_x86_64.whl", hash = "sha256:65c2c3add92b45fd0709db8594536aea39c2a67af0e27ffcf049c498501140b7", size = 5037050, upload-time = "2026-07-31
T14:24:47.697Z" },
    { url = "https://files.pythonhosted.org/packages/57/30/4a22984d4f1bdfb8c054f07a92bc176b97a3134cc1d6c4b3bffb1f3688b4/cryptography-50.0.0-cp39-abi3-
win_amd64.whl", hash = "sha256:d24fead1d4d076e1bfb006dcec392074a3cd8d7b4fc8a595aa64073b2b7a96ba", size = 3874135, upload-time = "2026-07-31T14:24:50.0
85Z" },
]

[[package]]
name = "dataclasses-json"
version = "0.6.7"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "marshmallow" },
    { name = "typing-inspect" },
]
sdist = { url = "https://files.pythonhosted.org/packages/64/a4/f71d9cf3a5ac257c993b5ca3f93df5f7fb395c725e7f1e6479d2514173c3/dataclasses_json-0.6.7.tar
```

```
.gz", hash = "sha256:b6b3e528266ea45b9535223bc53ca645f5208833c29229e847b3f26a1cc55fc0", size = 32227, upload-time = "2024-06-09T16:20:19.103Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/c3/be/d0d44e092656fe7a06b55e6103cbce807cdbdee17884a5367c68c9860853/dataclasses_json-0.6.7-py3-non
e-any.whl", hash = "sha256:0dbf33f26c8d5305befd61b39d2b3414e8a407bedc2834dea9b8d642666fb40a", size = 28686, upload-time = "2024-06-09T16:20:16.715Z" }
,
]

[[package]]
name = "defusedxml"
version = "0.7.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/0f/d5/c66da9b79e5bdb124974bfe172b4daf3c984ebd9c2a06e2b8a4dc7331c72/defusedxml-0.7.1.tar.gz",
hash = "sha256:1bb3032db185915b62d7c6209c5a8792be6a32ab2fedacc84e01b52c51aa3e69", size = 75520, upload-time = "2021-03-08T10:59:26.269Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/07/6c/aa3f2f849e01cb6a001cd8554a88d4c77c5c1a31c95bdf1cf9301e6d9ef4/defusedxml-0.7.1-py2.py3-none-
any.whl", hash = "sha256:a352e7e428770286cc899e2542b6cdaedb2b4953ff269a210103ec58f6198a61", size = 25604, upload-time = "2021-03-08T10:59:24.45Z" },
]

[[package]]
name = "deprecated"
version = "1.3.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "wrapt" },
]
sdist = { url = "https://files.pythonhosted.org/packages/49/85/12f0a49a7c4ffb70572b6c2ef13c90c88fd190debda93b23f026b25f9634/deprecated-1.3.1.tar.gz",
hash = "sha256:b1b50e0ff0c1fddaa5708a2c6b0a6588bb09b892825ab2b214ac9ea9d92a5223", size = 2932523, upload-time = "2025-10-30T08:19:02.757Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/84/d0/205d54408c08b13550c733c4b85429e7ead111c7f0014309637425520a9a/deprecated-1.3.1-py2.py3-none-
any.whl", hash = "sha256:597bfef186b6f60181535a29fbe44865ce137a5079f295b479886c82729d5f3f", size = 11298, upload-time = "2025-10-30T08:19:00.758Z" },
]

[[package]]
name = "dirtyjson"
version = "1.0.8"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/db/04/d24f6e645ad82ba0ef092fa17d9ef7a21953781663648a01c9371d9e8e98/dirtyjson-1.0.8.tar.gz", h
ash = "sha256:90ca4a18f3ff30ce849d100dcf4a003953c79d3a2348ef056f1d9c22231a25fd", size = 30782, upload-time = "2022-11-28T23:32:33.319Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/68/69/1bcf70f81de1b4a9f21b3a62ec0c83bdff991c88d6cc2267d02408457e88/dirtyjson-1.0.8-py3-none-any.w
hl", hash = "sha256:125e27248435a58acace26d5c2c4c11a1c0de0a9c5124c5a94ba78e517d74f53", size = 25197, upload-time = "2022-11-28T23:32:31.219Z" },
]

[[package]]
name = "distro"
version = "1.9.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/fc/f8/98eea607f65de6527f8a2e8885fc8015d3e6f5775df186e443e0964a11c3/distro-1.9.0.tar.gz", hash
 = "sha256:2fa77c6fd8940f116ee1d6b94a2f90b13b5ea8d019b98bc8bafdcabcdd9bdbed", size = 60722, upload-time = "2023-12-24T09:54:32.31Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/12/b3/231ffd4ab1fc9d679809f356cebee130ac7daa00d6d6f3206dd4fd137e9e/distro-1.9.0-py3-none-any.whl"
, hash = "sha256:7bffd925d65168f85027d8da9af6bddab658135b840670a223589bc0c8ef02b2", size = 20277, upload-time = "2023-12-24T09:54:30.421Z" },
]

[[package]]
name = "docstring-parser"
version = "0.18.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/e0/4d/f332313098c1de1b2d2ff91cf2674415cc7cddab2ca1b01ae29774bd5fdf/docstring_parser-0.18.0.ta
r.gz", hash = "sha256:292510982205c12b1248696f44959db3cdd1740237a968ea1e2e7a900eeb2015", size = 29341, upload-time = "2026-04-14T04:09:19.867Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a7/5f/ed01f9a3cdffbd5a008556fc7b2a08ddb1cc6ace7effa7340604b1d16699/docstring_parser-0.18.0-py3-no
ne-any.whl", hash = "sha256:b3fcbed555c47d8479be0796ef7e19c2670d428d72e96da63f3a40122860374b", size = 22484, upload-time = "2026-04-14T04:09:18.638Z"
},
]

[[package]]
name = "ecdsa"
version = "0.19.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "six" },
]
sdist = { url = "https://files.pythonhosted.org/packages/25/ca/8de7744cb3bc966c85430ca2d0fcaeea872507c6a4cf6e007f7fe269ed9d/ecdsa-0.19.2.tar.gz", hash
 = "sha256:62635b0ac1ca2e027f82122b5b81cb706edc38cd91c63dda28e4f3455a2bf930", size = 202432, upload-time = "2026-03-26T09:58:17.675Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/51/79/119091c98e2bf49e24ed9f3ae69f816d715d2904aefa6a2baa039a2ba0b0/ecdsa-0.19.2-py2.py3-none-any.
whl", hash = "sha256:840f5dc5e375c68f36c1a7a5b9caad28f95daa65185c9253c0c08dd952bb7399", size = 150818, upload-time = "2026-03-26T09:58:15.808Z" },
]

[[package]]
name = "fastapi"
version = "0.141.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "annotated-doc" },
    { name = "pydantic" },
    { name = "starlette" },
    { name = "typing-extensions" },
    { name = "typing-inspection" },
]
sdist = { url = "https://files.pythonhosted.org/packages/8a/02/91e3416a8fdd715abb903a952a6bec7cdd8d14eed55d415fc8595524c319/fastapi-0.141.1.tar.gz", h
ash = "sha256:e8822fc40db1e1858054d7a949a888695bc9bdce70139178e33bd2871a453ca1", size = 425799, upload-time = "2026-07-29T17:18:05.568Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/cb/03/10388a42375ee7e4ac9b94eb2c5c569c8b5795e377e701c9ac3ad63de890/fastapi-0.141.1-py3-none-any.w
hl", hash = "sha256:bfb91aa2d334c61cb35ba9a116fc123b3d3df31640b801cf57a7a78ec3f603b3", size = 131954, upload-time = "2026-07-29T17:18:04.364Z" },
]

[[package]]
name = "filetype"
version = "1.2.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/bb/29/745f7d30d47fe0f251d3ad3dc2978a2314191766199876 3bebb6da007eb1/filetype-1.2.0.tar.gz", ha
sh = "sha256:66b56cd6474bf41d8c54660347d37afcc3f7d1970648de365c102ef77548aadb", size = 998020, upload-time = "2022-11-02T17:34:04.141Z" }
wheels = [
```

```
    { url = "https://files.pythonhosted.org/packages/18/79/1b8fa1bb3568781e84c9200f951c735f3f157429f44be0495da55894d620/filetype-1.2.0-py2.py3-none-any.whl", hash = "sha256:7ce71b6880181241cf7ac8697a2f1eb6a8bd9b429f7ad6d27b8db9ba5f1c2d25", size = 19970, upload-time = "2022-11-02T17:34:01.425Z" },
]

[[package]]
name = "frozenlist"
version = "1.8.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/2d/f5/c831fac6cc817d26fd54c7eaccd04ef7e0288806943f7cc5bbf69f3ac1f0/frozenlist-1.8.0.tar.gz", hash = "sha256:3ede829ed8d842f6cd48fc7081d7a41001a56f1f38603f9d49bf3020d59a31ad", size = 45875, upload-time = "2025-10-06T05:38:17.865Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/69/29/948b9aa87e75820a38650af445d2ef2b6b8a6fab1a23b6bb9e4ef0be2d59/frozenlist-1.8.0-cp312-cp312-macosx_10_13_universal2.whl", hash = "sha256:78f7b9e5d6f2fdb88cdde9440dc147259b62b9d3b019924def9f6478be254ac1", size = 87782, upload-time = "2025-10-06T05:36:06.649Z" },
    { url = "https://files.pythonhosted.org/packages/64/80/4f6e318ee2a7c0750ed724fa33a4bdf1eacdc5a39a7a24e818a773cd91af/frozenlist-1.8.0-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:229bf37d2e4acdaf808fd3f06e854a4a7a3661e871b10dc1f8f1896a3b05f18b", size = 50594, upload-time = "2025-10-06T05:36:07.69Z" },
    { url = "https://files.pythonhosted.org/packages/2b/94/5c8a2b50a496b11dd519f4a24cb5496cf125681dd99e94c604ccdea9419a/frozenlist-1.8.0-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:f833670942247a14eafbb675458b4e61c82e002a148f49e68257b79296e865c4", size = 50448, upload-time = "2025-10-06T05:36:08.78Z" },
    { url = "https://files.pythonhosted.org/packages/6a/bd/d91c5e39f490a49df14320f4e8c80161cfcce09f1e2cde1edd16a551abb3/frozenlist-1.8.0-cp312-cp312-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:494a5952b1c597ba44e0e78113a7266e656b9794eec897b19ead706bd7074383", size = 242411, upload-time = "2025-10-06T05:36:09.801Z" },
    { url = "https://files.pythonhosted.org/packages/8f/83/f61505a05109ef3293dfb1ff594d13d64a2324ac3482be2cedc2be818256/frozenlist-1.8.0-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:96f423a119f4777a4a056b66ce11527366a8bb92f54e541ade21f2374433f6d4", size = 243014, upload-time = "2025-10-06T05:36:11.394Z" },
    { url = "https://files.pythonhosted.org/packages/d8/cb/cb6c7b0f7d4023ddda30cf56b8b17494eb3a79e3fda666bf735f63118b35/frozenlist-1.8.0-cp312-cp312-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:3462dd9475af2025c31cc61be6652dfa25cbfb56cbbf52f4ccfe029f38decaf8", size = 234909, upload-time = "2025-10-06T05:36:12.598Z" },
    { url = "https://files.pythonhosted.org/packages/31/c5/cd7a1f3b8b34af009fb17d4123c5a778b44ae2804e3ad6b86204255f9ec5/frozenlist-1.8.0-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:c4c800524c9cd9bac5166cd6f55285957fcfc907db323e193f2afcd4d9abd69b", size = 250049, upload-time = "2025-10-06T05:36:14.065Z" },
    { url = "https://files.pythonhosted.org/packages/c0/01/2f95d3b416c584a1e7f0e1d6d31998c4a795f7544069ee2e0962a4b60740/frozenlist-1.8.0-cp312-cp312-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:d6a5df73acd3399d893dafc71663ad22534b5aa4f94e8a2fabfe856c3c1b6a52", size = 256485, upload-time = "2025-10-06T05:36:15.39Z" },
    { url = "https://files.pythonhosted.org/packages/ce/03/024bf7720b3abaebcff6d0793d73c154237b85bdf67b7ed55e5e9596dc9a/frozenlist-1.8.0-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:405e8fe955c2280ce66428b3ca55e12b3c4e9c336fb2103a4937e891c69a4a29", size = 237619, upload-time = "2025-10-06T05:36:16.558Z" },
    { url = "https://files.pythonhosted.org/packages/69/fa/f8abdfe7d76b731f5d8bd217827cf6764d4f1d9763407e42717b4bed50a0/frozenlist-1.8.0-cp312-cp312-musllinux_1_2_armv7l.whl", hash = "sha256:908bd3f6439f2fef9e85031b59fd4f1297af54415fb60e4254a95f75b3cab3f3", size = 250320, upload-time = "2025-10-06T05:36:17.821Z" },
    { url = "https://files.pythonhosted.org/packages/f5/3c/b051329f718b463b22613e269ad72138cc256c540f78a6de89452803a47d/frozenlist-1.8.0-cp312-cp312-musllinux_1_2_ppc64le.whl", hash = "sha256:294e487f9ec720bd8ffcebc99d575f7eff3568a08a253d1ee1a0378754b74143", size = 246820, upload-time = "2025-10-06T05:36:19.046Z" },
    { url = "https://files.pythonhosted.org/packages/0f/ae/58282e8f98e444b3f4dd42448ff36fa38bef29e40d40f330b22e7108f565/frozenlist-1.8.0-cp312-cp312-musllinux_1_2_s390x.whl", hash = "sha256:74c51543498289c0c43656701be6b077f4b265868fa7f8a8859c197006efb608", size = 250518, upload-time = "2025-10-06T05:36:20.763Z" },
    { url = "https://files.pythonhosted.org/packages/8f/96/007e5944694d66123183845a106547a15944fbbb7154788cbf7272789536/frozenlist-1.8.0-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:776f352e8329135506a1d6bf16ac3f87bc25b28e765949282dcc627af36123aa", size = 239096, upload-time = "2025-10-06T05:36:22.129Z" },
    { url = "https://files.pythonhosted.org/packages/66/bb/852b9d6db2fa40be96f29c0d1205c306288f0684df8fd26ca1951d461a56/frozenlist-1.8.0-cp312-cp312-win32.whl", hash = "sha256:433403ae80709741ce34038da08511d4a77062aa924baf411ef73d1146e74faf", size = 39985, upload-time = "2025-10-06T05:36:23.661Z" },
    { url = "https://files.pythonhosted.org/packages/b8/af/38e51a553dd66eb064cdf193841f16f077585d4d28394c2fa6235cb41765/frozenlist-1.8.0-cp312-cp312-win_amd64.whl", hash = "sha256:34187385b08f866104f0c0617404c8eb08165ab1272e884abc89c112e9c00746", size = 44591, upload-time = "2025-10-06T05:36:24.958Z" },
    { url = "https://files.pythonhosted.org/packages/a7/06/1dc65480ab147339fecc70797e9c2f69d9cea9cf38934ce08df070fdb9cb/frozenlist-1.8.0-cp312-cp312-win_arm64.whl", hash = "sha256:fe3c58d2f5db5fbd18c2987cba06d51b0529f52bc3a6cdc33d3f4eab725104bd", size = 40102, upload-time = "2025-10-06T05:36:26.333Z" },
    { url = "https://files.pythonhosted.org/packages/2d/40/0832c31a37d60f60ed79e9dfb5a92e1e2af4f40a16a29abcc7992af9edff/frozenlist-1.8.0-cp313-cp313-macosx_10_13_universal2.whl", hash = "sha256:8d92f1a84bb12d9e56f818b3a746f3efba93c1b63c8387a73dde655e1e42282a", size = 85717, upload-time = "2025-10-06T05:36:27.341Z" },
    { url = "https://files.pythonhosted.org/packages/30/ba/b0b3de23f40bc55a7057bd38434e25c34fa48e17f20ee273bbde5e0650f3/frozenlist-1.8.0-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:96153e77a591c8adc2ee805756c61f59fef4cf4073a9275ee86fe8cba41241f7", size = 49651, upload-time = "2025-10-06T05:36:28.855Z" },
    { url = "https://files.pythonhosted.org/packages/0c/ab/6e5080ee374f875296c4243c381bbdef97a9ac39c6e3ce1d5f7d42cb78d6/frozenlist-1.8.0-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:f21f00a91358803399890ab167098c131ec2ddd5f8f5fd5fe9c9f2c6fcd91e40", size = 49417, upload-time = "2025-10-06T05:36:29.877Z" },
    { url = "https://files.pythonhosted.org/packages/d5/4e/e4691508f9477ce67da2015d8c00acd751e6287739123113a9fca6f1604e/frozenlist-1.8.0-cp313-cp313-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:fb30f9626572a76dfe4293c7194a09fb1fe93ba94c7d4f720dfae3b646b45027", size = 234391, upload-time = "2025-10-06T05:36:31.301Z" },
    { url = "https://files.pythonhosted.org/packages/40/76/c202df58e3acdf12969a7895fd6f3bc016c642e6726aa63bd3025e0fc71c/frozenlist-1.8.0-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:eaa352d7047a31d87dafcacbabe89df0aa506abb5b1b85a2fb91bc3faa02d822", size = 233048, upload-time = "2025-10-06T05:36:32.531Z" },
    { url = "https://files.pythonhosted.org/packages/f9/c0/8746afb90f17b73ca5979c7a3958116e105ff796e718575175319b5bb4ce/frozenlist-1.8.0-cp313-cp313-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:03ae967b4e297f58f8c774c7eabcce57fe3c2434817d4385c50661845a058121", size = 226549, upload-time = "2025-10-06T05:36:33.706Z" },
    { url = "https://files.pythonhosted.org/packages/7e/eb/4c7eefc718ff72f9b6c4893291abaae5fbc0c82226a32dcd8ef4f7a5dbef/frozenlist-1.8.0-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:f6292f1de555ffcc675941d65fffffb0a5bcd992905015f85d0592201793e0e5", size = 239833, upload-time = "2025-10-06T05:36:34.947Z" },
    { url = "https://files.pythonhosted.org/packages/c2/4e/e5c02187cf704224f8b21bee886f3d713ca379535f16893233b9d672ea71/frozenlist-1.8.0-cp313-cp313-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:29548f9b5b5e3460ce7378144c3010363d8035cea44bc0bf02d57f5a685e084e", size = 245363, upload-time = "2025-10-06T05:36:36.534Z" },
    { url = "https://files.pythonhosted.org/packages/1f/96/cb85ec608464472e82ad37a17f844889c36100eed57bea094518bf270692/frozenlist-1.8.0-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:ec3cc8c5d4084591b4237c0a272cc4f50a5b03396a47d9caaf76f5d7b38a4f11", size = 229314, upload-time = "2025-10-06T05:36:38.582Z" },
    { url = "https://files.pythonhosted.org/packages/5d/6f/4ae69c550e4cee66b57887daeebe006fe985917c01d0fff9caab9883f6d0/frozenlist-1.8.0-cp313-cp313-musllinux_1_2_armv7l.whl", hash = "sha256:517279f58009d0b1f2e7c1b130b377a349405da3f7621ed6bfae50b10adf20c1", size = 243365, upload-time = "2025-10-06T05:36:40.152Z" },
    { url = "https://files.pythonhosted.org/packages/7a/58/afd56de246cf11780a40a2c28dc7cbabbf06337cc8ddb1c780a2d97e88d8/frozenlist-1.8.0-cp313-cp313-musllinux_1_2_ppc64le.whl", hash = "sha256:db1e72ede2d0d7ccb213f218df6a078a9c09a7de257c2fe8fcef16d5925230b1", size = 237763, upload-time = "2025-10-06T05:36:41.355Z" },
    { url = "https://files.pythonhosted.org/packages/cb/36/cdfaf6ed42e2644740d4a10452d8e97fa1c062e2a8006e4b09f1b5fd7d63/frozenlist-1.8.0-cp313-cp313-musllinux_1_2_s390x.whl", hash = "sha256:b4dec9482a65c54a5044486847b8a66bf10c9cb4926d42927ec4e8fd5db7fed8", size = 240110, upload-time = "2025-10-06T05:36:42.716Z" },
    { url = "https://files.pythonhosted.org/packages/03/a8/9ea226fbefad669f11b52e864c55f0bd57d3c8d7eb07e9f2e9a0b39502e1/frozenlist-1.8.0-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:21900c48ae04d13d416f0e1e0c4d81f7931f73a9dfa0b7a8746fb2fe7dd970ed", size = 233717, upload-time = "2025-10-06T05:36:44.251Z" },
    { url = "https://files.pythonhosted.org/packages/1e/0b/1b5531611e83ba7d13ccc9988967ea1b51186af64c42b7a7af465dcc9568/frozenlist-1.8.0-cp313-cp313-win32.whl", hash = "sha256:8b7b94a067d1c504ee0b16def57ad5738701e4ba10cec90529f13fa03c833496", size = 39628, upload-time = "2025-10-06T05:36:45.423Z" },
    { url = "https://files.pythonhosted.org/packages/d8/cf/174c91dbc9cc49bc7b7aab74d8b734e974d1faa8f191c74af9b7e80848e6/frozenlist-1.8.0-cp313-cp313-win_amd64.whl", hash = "sha256:878be833caa6a3821caf85eb39c5ba92d28e85df26d57afb06b35b2efd937231", size = 43882, upload-time = "2025-10-06T05:36:46.796Z" },
    { url = "https://files.pythonhosted.org/packages/c1/17/502cd212cbfa96eb1388614fe39a3fc9ab87dbbe042b66f97acb57474834/frozenlist-1.8.0-cp313-cp313-win_arm64.whl", hash = "sha256:44389d135b3ff43ba8cc89ff7f51f5a0bb6b63d829c8300f79a2fe4fe61bcc62", size = 39676, upload-time = "2025-10-06T05:36:47.8Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/d2/5c/3bbfaa920dfab09e76946a5d2833a7cbdf7b9b4a91c714666ac4855b88b4/frozenlist-1.8.0-cp313-cp313t-macosx_10_13_universal2.whl", hash = "sha256:e25ac20a2ef37e91c1b39938b591457666a0fa835c7783c3a8f33ea42870db94", size = 89235, upload-time = "2025-10-06T05:36:48.78Z" },
    { url = "https://files.pythonhosted.org/packages/d2/d6/f03961ef72166cec1687e84e8925838442b615bd0b8854b54923ce5b7b8a/frozenlist-1.8.0-cp313-cp313t-macosx_10_13_x86_64.whl", hash = "sha256:07cdca25a91a4386d2e76ad992916a85038a9b97561bf7a3fd12d5d9ce31870c", size = 50742, upload-time = "2025-10-06T05:36:49.837Z" },
    { url = "https://files.pythonhosted.org/packages/1e/bb/a6d12b7ba4c3337667d0e421f7181c82dda448ce4e7ad7ecd249a16fa806/frozenlist-1.8.0-cp313-cp313t-macosx_11_0_arm64.whl", hash = "sha256:4e0c11f2cc6717e0a741f84a527c52616140741cd812a50422f83dc31749fb52", size = 51725, upload-time = "2025-10-06T05:36:50.851Z" },
    { url = "https://files.pythonhosted.org/packages/bc/71/d1fed0ffe2c2ccd70b43714c6cab0f4188f09f8a67a7914a6b46ee30f274/frozenlist-1.8.0-cp313-cp313t-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:b3210649ee28062ea6099cfda39e147fa1bc039583c8ee4481cb7811e2448c51", size = 284533, upload-time = "2025-10-06T05:36:51.898Z" },
    { url = "https://files.pythonhosted.org/packages/c9/1f/fb1685a7b009d89f9bf78a42d94461bc06581f6e718c39344754a5d9bada/frozenlist-1.8.0-cp313-cp313t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:581ef5194c48035a7de2aefc72ac6539823bb71508189e5de01d60c9dcd5fa65", size = 292506, upload-time = "2025-10-06T05:36:53.101Z" },
    { url = "https://files.pythonhosted.org/packages/e6/3b/b991fe1612703f7e0d05c0cf734c1b77aaf7c7d321df4572e8d36e7048c8/frozenlist-1.8.0-cp313-cp313t-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:3ef2d026f16a2b1866e1d86fc4e1291e1ed8a387b2c333809419a2f8b3a77b82", size = 274161, upload-time = "2025-10-06T05:36:54.309Z" },
    { url = "https://files.pythonhosted.org/packages/ca/ec/c5c618767bcdf66e88945ec0157d7f6c4a1322f1473392319b7a2501ded7/frozenlist-1.8.0-cp313-cp313t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:5500ef82073f599ac84d888e3a8c1f77ac831183244bfd7f11eaa0289fb30714", size = 294676, upload-time = "2025-10-06T05:36:55.566Z" },
    { url = "https://files.pythonhosted.org/packages/7c/ce/3934758637d8f8a88d11f0585d6495ef54b2044ed6ec84492a91fa3b27aa/frozenlist-1.8.0-cp313-cp313t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:50066c3997d0091c411a66e710f4e11752251e6d2d73d70d8d5d4c76442a199d", size = 300638, upload-time = "2025-10-06T05:36:56.758Z" },
    { url = "https://files.pythonhosted.org/packages/fc/4f/a7e4d0d467298f42de4b41cbc7ddaf19d3cfeabaf9ff97c20c6c7ee409f9/frozenlist-1.8.0-cp313-cp313t-musllinux_1_2_aarch64.whl", hash = "sha256:5c1c8e78426e59b3f8005e9b19f6ff46e5845895adbde20ece9218319eca6506", size = 283067, upload-time = "2025-10-06T05:36:57.965Z" },
    { url = "https://files.pythonhosted.org/packages/dc/48/c7b163063d55a83772b268e6d1affb960771b0e203b632cfe09522d67ea5/frozenlist-1.8.0-cp313-cp313t-musllinux_1_2_armv7l.whl", hash = "sha256:eefdba20de0d938cec6a89bd4d70f346a03108a19b9df4248d3cf0d88f1b0f51", size = 292101, upload-time = "2025-10-06T05:36:59.237Z" },
    { url = "https://files.pythonhosted.org/packages/9f/d0/2366d3c4ecdc2fd391e0afa6e11500bfba0ea772764d631bbf82f0136c9d/frozenlist-1.8.0-cp313-cp313t-musllinux_1_2_ppc64le.whl", hash = "sha256:cf253e0e1c3ceb4aaff6df637ce033ff6535fb8c70a764a8f46aafd3d6ab798e", size = 289901, upload-time = "2025-10-06T05:37:00.811Z" },
    { url = "https://files.pythonhosted.org/packages/b8/94/daff920e82c1b70e3618a2ac39fbc01ae3e2ff6124e80739ce5d71c9b920/frozenlist-1.8.0-cp313-cp313t-musllinux_1_2_s390x.whl", hash = "sha256:032efa2674356903cd0261c4317a561a6850f3ac864a63fc1583147fb05a79b0", size = 289395, upload-time = "2025-10-06T05:37:02.115Z" },
    { url = "https://files.pythonhosted.org/packages/e3/20/bba307ab4235a09fdcd3cc5508dbabd17c4634a1af4b96e0f69bfe551ebd/frozenlist-1.8.0-cp313-cp313t-musllinux_1_2_x86_64.whl", hash = "sha256:6da155091429aeba16851ecb10a9104a108bcd32f6c1642867eadaee401c1c41", size = 283659, upload-time = "2025-10-06T05:37:03.711Z" },
    { url = "https://files.pythonhosted.org/packages/fd/00/04ca1c3a7a124b6de4f8a9a17cc2fcad138b4608e7a3fc5877804b8715d7/frozenlist-1.8.0-cp313-cp313t-win32.whl", hash = "sha256:0f96534f8bfebc1a394209427d0f8a63d343c9779cda6fc25e8e121b5fd8555b", size = 43492, upload-time = "2025-10-06T05:37:04.915Z" },
    { url = "https://files.pythonhosted.org/packages/59/5e/c69f733a86a94ab10f68e496dc6b7e8bc078ebb415281d5698313e3af3a1/frozenlist-1.8.0-cp313-cp313t-win_amd64.whl", hash = "sha256:5d63a068f978fc69421fb0e6eb91a9603187527c86b7cd3f534a5b77a592b888", size = 48034, upload-time = "2025-10-06T05:37:06.343Z" },
    { url = "https://files.pythonhosted.org/packages/16/6c/be9d79775d8abe79b05fa6d23da99ad6e7763a1d080fbae7290b286093fd/frozenlist-1.8.0-cp313-cp313t-win_arm64.whl", hash = "sha256:bf0a7e10b077bf5fb9380ad3ae8ce20ef919a6ad93b4552896419ac7e1d8e042", size = 41749, upload-time = "2025-10-06T05:37:07.431Z" },
    { url = "https://files.pythonhosted.org/packages/f1/c8/85da824b7e7b9b6e7f7705b2ecaf9591ba6f79c1177f324c2735e41d36a2/frozenlist-1.8.0-cp314-cp314-macosx_10_13_universal2.whl", hash = "sha256:cee686f1f4cadeb2136007ddedd0aaf928ab95216e7691c63e50a8ec066336d0", size = 86127, upload-time = "2025-10-06T05:37:08.438Z" },
    { url = "https://files.pythonhosted.org/packages/8e/e8/a1185e236ec66c20afd72399522f142c3724c785789255202d27ae992818/frozenlist-1.8.0-cp314-cp314-macosx_10_13_x86_64.whl", hash = "sha256:119fb2a1bd47307e899c2fac7f28e85b9a543864df47aa7ec9d3c1b4545f096f", size = 49698, upload-time = "2025-10-06T05:37:09.48Z" },
    { url = "https://files.pythonhosted.org/packages/a1/93/72b1736d68f03fda5fdf0f2180fb6caaae3894f1b854d006ac61ecc727ee/frozenlist-1.8.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:4970ece02dbc8c3a92fcc5228e36a3e933a01a999f7094ff7c23fbd2beeaa67c", size = 49749, upload-time = "2025-10-06T05:37:10.569Z" },
    { url = "https://files.pythonhosted.org/packages/a7/b2/fabede9fafd976b991e9f1b9c8c873ed86f202889b864756f240ce6dd855/frozenlist-1.8.0-cp314-cp314-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:cba69cb73723c3f329622e34bdbf5ce1f80c21c290ff04256cff1cd3c2036ed2", size = 231298, upload-time = "2025-10-06T05:37:11.993Z" },
    { url = "https://files.pythonhosted.org/packages/3a/3b/d9b1e0b0eed36e70477ffb8360c49c85c8ca8ef9700a4e6711f39a6e8b45/frozenlist-1.8.0-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:778a11b15673f6f1df23d9586f83c4846c471a8af693a22e066508b77d201ec8", size = 232015, upload-time = "2025-10-06T05:37:13.194Z" },
    { url = "https://files.pythonhosted.org/packages/dc/94/be719d2766c1138148564a3960fc2c06eb688da592bdc25adcf856101be7/frozenlist-1.8.0-cp314-cp314-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:0325024fe97f94c41c08872db482cf8ac4800d80e79222c6b0b7b162d5b13686", size = 225038, upload-time = "2025-10-06T05:37:14.577Z" },
    { url = "https://files.pythonhosted.org/packages/e4/09/6712b6c5465f083f52f50cf74167b92d4ea2f50e46a9eea0523d658454ae/frozenlist-1.8.0-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:97260ff46b207a82a7567b581ab4190bd4dfa09f4db8a8b49d1a958f6aa4940e", size = 240130, upload-time = "2025-10-06T05:37:15.781Z" },
    { url = "https://files.pythonhosted.org/packages/f8/d4/cd065cdcf21550b54f3ce6a22e143ac9e4836ca42a0de1022da8498eac89/frozenlist-1.8.0-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:54b2077180eb7f83dd52c40b2750d0a9f175e06a42e3213ce047219de902717a", size = 242845, upload-time = "2025-10-06T05:37:17.037Z" },
    { url = "https://files.pythonhosted.org/packages/62/c3/f57a5c8c70cd1ead3d5d5f776f89d33110b1addae0ab010ad774d9a44fb9/frozenlist-1.8.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:2f05983daecab868a31e1da44462873306d3cbfd76d1f0b5b69c473d21dbb128", size = 229131, upload-time = "2025-10-06T05:37:18.221Z" },
    { url = "https://files.pythonhosted.org/packages/6c/52/232476fe9cb64f0742f3fde2b7d26c1dac18b6d62071c74d4ded55e0ef94/frozenlist-1.8.0-cp314-cp314-musllinux_1_2_armv7l.whl", hash = "sha256:33f48f51a446114bc5d251fb2954ab0164d5be02ad3382abcbfe07e2531d650f", size = 240542, upload-time = "2025-10-06T05:37:19.771Z" },
    { url = "https://files.pythonhosted.org/packages/5f/85/07bf3f5d0fb5414aee5f47d33c6f5c77bfe49aac680bfece33d4fdf6a246/frozenlist-1.8.0-cp314-cp314-musllinux_1_2_ppc64le.whl", hash = "sha256:154e55ec0655291b5dd1b8731c637ecdb50975a2ae70c606d100750a540082f7", size = 237308, upload-time = "2025-10-06T05:37:20.969Z" },
    { url = "https://files.pythonhosted.org/packages/11/99/ae3a33d5befd41ac0ca2cc7fd3aa707c9c324de2e89db0e0f45db9a64c26/frozenlist-1.8.0-cp314-cp314-musllinux_1_2_s390x.whl", hash = "sha256:4314debad13beb564b708b4a496020e5306c7333fa9a3ab90374169a20ffab30", size = 238210, upload-time = "2025-10-06T05:37:22.252Z" },
    { url = "https://files.pythonhosted.org/packages/b2/60/b1d2da22f4970e7a155f0adde9b1435712ece01b3cd45ba63702aea33938/frozenlist-1.8.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:073f8bf8becba60aa931eb3bc420b217bb7d5b8f4750e6f8b3be7f3da85d38b7", size = 231972, upload-time = "2025-10-06T05:37:23.5Z" },
    { url = "https://files.pythonhosted.org/packages/3f/ab/945b2f32de889993b9c9133216c068b7fcf257d8595a0ac420ac8677cab0/frozenlist-1.8.0-cp314-cp314-win32.whl", hash = "sha256:bac9c42ba2ac65ddc115d930c78d24ab8d4f465fd3fc473cdedfccadb9429806", size = 40536, upload-time = "2025-10-06T05:37:25.581Z" },
    { url = "https://files.pythonhosted.org/packages/59/ad/9caa9b9c836d9ad6f067157a531ac48b7d36499f5036d4141ce78c230b1b/frozenlist-1.8.0-cp314-cp314-win_amd64.whl", hash = "sha256:3e0761f4d1a44f1d1a47996511752cf3dcec5bbdd9cc2b4fe595caf97754b7a0", size = 44330, upload-time = "2025-10-06T05:37:26.928Z" },
    { url = "https://files.pythonhosted.org/packages/82/13/e6950121764f2676f43534c555249f57030150260aee9dcf7d64efda11dd/frozenlist-1.8.0-cp314-cp314-win_arm64.whl", hash = "sha256:d1eaff1d00c7751b7c6662e9c5ba6eb2c17a2306ba5e2a37f24ddf3cc953402b", size = 40627, upload-time = "2025-10-06T05:37:28.075Z" },
    { url = "https://files.pythonhosted.org/packages/c0/c7/43200656ecc4e02d3f8bc248df68256cd9572b3f0017f0a0c4e93440ae23/frozenlist-1.8.0-cp314-cp314t-macosx_10_13_universal2.whl", hash = "sha256:d3bb933317c52d7ea5004a1c442eef86f426886fba134ef8cf4226ea6ee1821d", size = 89238, upload-time = "2025-10-06T05:37:29.373Z" },
    { url = "https://files.pythonhosted.org/packages/d1/29/55c5f0689b9c0fb765055629f472c0de484dcaf0acee2f7707266ae3583c/frozenlist-1.8.0-cp314-cp314t-macosx_10_13_x86_64.whl", hash = "sha256:8009897cdef112072f93a0efdce29cd819e717fd2f649ee3016efd3cd885a7ed", size = 50738, upload-time = "2025-10-06T05:37:30.792Z" },
    { url = "https://files.pythonhosted.org/packages/ba/7d/b7282a445956506fa11da8c2db7d276adcbf2b17d8bb8407a47685263f90/frozenlist-1.8.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:2c5dcbbc55383e5883246d11fd179782a9d07a986c40f49abe89ddf865913930", size = 51739, upload-time = "2025-10-06T05:37:32.127Z" },
    { url = "https://files.pythonhosted.org/packages/62/1c/3d8622e60d0b767a5510d1d3cf21065b9db874696a51ea6d7a43180a259c/frozenlist-1.8.0-cp314-cp314t-
```

```
manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:39ecbc32f1390387d2aa4f5a995e465e9e2f79ba3adcac92d68e3e0afae6657c", s
ize = 284186, upload-time = "2025-10-06T05:37:33.21Z" },
    { url = "https://files.pythonhosted.org/packages/2d/14/aa36d5f85a89679a85a1d44cd7a6657e0b1c75f61e7cad987b203d2daca8/frozenlist-1.8.0-cp314-cp314t-
manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:92db2bf818d5cc8d9c1f1fc56b897662e24ea5adb36ad1f1d82875bd64e03
c24", size = 292196, upload-time = "2025-10-06T05:37:36.107Z" },
    { url = "https://files.pythonhosted.org/packages/05/23/6bde59eb55abd407d34f77d39a5126fb7b4f109a3f611d3929f14b700c66/frozenlist-1.8.0-cp314-cp314t-
manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:2dc43a022e555de94c3b68a4ef0b11c4f747d12c024a520c7101709a2144fb37
", size = 273830, upload-time = "2025-10-06T05:37:37.663Z" },
    { url = "https://files.pythonhosted.org/packages/d2/3f/22cff331bfad7a8afa616289000ba793347fcd7bc275f3b28ecea2a27909/frozenlist-1.8.0-cp314-cp314t-
manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:cb89a7f2de3602cfed448095bab3f178399646ab7c61454315089787df077
33a", size = 294289, upload-time = "2025-10-06T05:37:39.261Z" },
    { url = "https://files.pythonhosted.org/packages/a4/89/5b057c799de4838b6c69aa82b79705f2027615e01be996d2486a69ca99c4/frozenlist-1.8.0-cp314-cp314t-
manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:33139dc858c580ea50e7e60a1b0ea003efa1fd42e6ec7fdbad78fff65fad2fd2",
size = 300318, upload-time = "2025-10-06T05:37:43.213Z" },
    { url = "https://files.pythonhosted.org/packages/30/de/2c22ab3eb2a8af6d69dc799e48455813bab3690c760de58e1bf43b36da3e/frozenlist-1.8.0-cp314-cp314t-
musllinux_1_2_aarch64.whl", hash = "sha256:168c0969a329b416119507ba30b9ea13688fafffac1b7822802537569a1cb0ef", size = 282814, upload-time = "2025-10-06
T05:37:45.337Z" },
    { url = "https://files.pythonhosted.org/packages/59/f7/970141a6a8dbd7f556d94977858cfb36fa9b66e0892c6dd780d2219d8cd8/frozenlist-1.8.0-cp314-cp314t-
musllinux_1_2_armv7l.whl", hash = "sha256:28bd570e8e189d7f7b001966435f9dac6718324b5be2990ac496cf1ea9ddb7fe", size = 291762, upload-time = "2025-10-06T
05:37:46.657Z" },
    { url = "https://files.pythonhosted.org/packages/c1/15/ca1adae83a719f82df9116d66f5bb28bb95557b3951903d39135620ef157/frozenlist-1.8.0-cp314-cp314t-
musllinux_1_2_ppc64le.whl", hash = "sha256:b2a095d45c5d46e5e79ba1e5b9cb787f541a8dee0433836cea4b96a2c439dcd8", size = 289470, upload-time = "2025-10-06
T05:37:47.946Z" },
    { url = "https://files.pythonhosted.org/packages/ac/83/dca6dc53bf657d371fbc88ddeb21b79891e747189c5de990b9dfff2ccba1/frozenlist-1.8.0-cp314-cp314t-
musllinux_1_2_s390x.whl", hash = "sha256:eab8145831a0d56ec9c4139b6c3e594c7a83c2c8be25d5bcf2d86136a532287a", size = 289042, upload-time = "2025-10-06T0
5:37:49.499Z" },
    { url = "https://files.pythonhosted.org/packages/96/52/abddd34ca99be142f354398700536c5bd315880ed0a213812bc491cff5e4/frozenlist-1.8.0-cp314-cp314t-
musllinux_1_2_x86_64.whl", hash = "sha256:974b28cf63cc99dfb2188d8d222bc6843656188164848c4f679e63dae4b0708e", size = 283148, upload-time = "2025-10-06T
05:37:50.745Z" },
    { url = "https://files.pythonhosted.org/packages/af/d3/76bd4ed4317e7119c2b7f57c3f6934aba26d277acc6309f873341640e21f/frozenlist-1.8.0-cp314-cp314t-
win32.whl", hash = "sha256:342c97bf697ac5480c0a7ec73cd700ecfa5a8a40ac923bd035484616efecc2df", size = 44676, upload-time = "2025-10-06T05:37:52.222Z" }
,
    { url = "https://files.pythonhosted.org/packages/89/76/c615883b7b521ead2944bb3480398cbb07e12b7b4e4d073d3752eb721558/frozenlist-1.8.0-cp314-cp314t-
win_amd64.whl", hash = "sha256:06be8f67f39c8b1dc671f5d83aaefd3358ae5cdcf8314552c57e7ed3e6475bdd", size = 49451, upload-time = "2025-10-06T05:37:53.425
Z" },
    { url = "https://files.pythonhosted.org/packages/e0/a3/5982da14e113d07b325230f95060e2169f5311b1017ea8af2a29b374c289/frozenlist-1.8.0-cp314-cp314t-
win_arm64.whl", hash = "sha256:102e6314ca4da683dca92e3b1355490fed5f313b768500084fbe6371fddfdb79", size = 42507, upload-time = "2025-10-06T05:37:54.513
Z" },
    { url = "https://files.pythonhosted.org/packages/9a/9a/e35b4a917281c0b8419d4207f4334c8e8c5dbf4f3f5f9ada73958d937dcc/frozenlist-1.8.0-py3-none-any.
whl", hash = "sha256:0c18a16eab41e82c295618a77502e17b195883241c563b00f0aa5106fc4eaa0d", size = 13409, upload-time = "2025-10-06T05:38:16.721Z" },
]

[[package]]
name = "fsspec"
version = "2026.7.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/00/78/f34251dadb8f3921264a1d9b8946f5e542014ee2614b285261b4e40e6775/fsspec-2026.7.0.tar.gz", h
ash = "sha256:c803c40f4cf860b49dea58ee3e1c33cb9c790520e233537e1340049f89b82a88", size = 317040, upload-time = "2026-07-28T16:34:51.052Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/fd/3c/6a2bf344106328fd04963664a60b9bb6496fc25df8e962fcdc1367285fb9/fsspec-2026.7.0-py3-none-any.w
hl", hash = "sha256:b57ddbafedfaef7018c1ecab32aa200a9d7ca26b77965f64e48b70061249d279", size = 206583, upload-time = "2026-07-28T16:34:49.538Z" },
]

[[package]]
name = "genbi-backend"
version = "0.1.0"
source = { virtual = "." }
dependencies = [
    { name = "alembic" },
    { name = "asyncpg" },
    { name = "bcrypt" },
    { name = "fastapi" },
    { name = "gunicorn" },
    { name = "httpx" },
    { name = "langchain" },
    { name = "langchain-anthropic" },
    { name = "langfuse" },
    { name = "langgraph" },
    { name = "llama-index" },
    { name = "loguru" },
    { name = "mcp" },
    { name = "numpy" },
    { name = "openai" },
    { name = "opentelemetry-api" },
    { name = "opentelemetry-instrumentation-fastapi" },
    { name = "opentelemetry-sdk" },
    { name = "pandas" },
    { name = "pgvector" },
    { name = "prometheus-client" },
    { name = "pydantic" },
    { name = "pydantic-settings" },
    { name = "python-jose", extra = ["cryptography"] },
    { name = "python-multipart" },
    { name = "pyyaml" },
    { name = "redis" },
    { name = "sqlalchemy", extra = ["asyncio"] },
    { name = "uvicorn", extra = ["standard"] },
]

[package.optional-dependencies]
dev = [
    { name = "httpx" },
    { name = "mypy" },
    { name = "pytest" },
    { name = "pytest-asyncio" },
    { name = "pytest-cov" },
    { name = "ruff" },
]

[package.metadata]
requires-dist = [
    { name = "alembic", specifier = ">=1.14" },
    { name = "asyncpg", specifier = ">=0.30" },
    { name = "bcrypt", specifier = ">=5.0" },
    { name = "fastapi", specifier = ">=0.115" },
```

```
    { name = "gunicorn", specifier = ">=23.0" },
    { name = "httpx", specifier = ">=0.28" },
    { name = "httpx", marker = "extra == 'dev'", specifier = ">=0.28" },
    { name = "langchain", specifier = ">=0.3" },
    { name = "langchain-anthropic", specifier = ">=0.3" },
    { name = "langfuse", specifier = ">=2.55" },
    { name = "langgraph", specifier = ">=0.2" },
    { name = "llama-index", specifier = ">=0.12" },
    { name = "loguru", specifier = ">=0.7" },
    { name = "mcp", specifier = ">=1.5" },
    { name = "mypy", marker = "extra == 'dev'", specifier = ">=1.11" },
    { name = "numpy", specifier = ">=2.1" },
    { name = "openai", specifier = ">=1.58" },
    { name = "opentelemetry-api", specifier = ">=1.28" },
    { name = "opentelemetry-instrumentation-fastapi", specifier = ">=0.49" },
    { name = "opentelemetry-sdk", specifier = ">=1.28" },
    { name = "pandas", specifier = ">=2.2" },
    { name = "pgvector", specifier = ">=0.3" },
    { name = "prometheus-client", specifier = ">=0.21" },
    { name = "pydantic", specifier = ">=2.9" },
    { name = "pydantic-settings", specifier = ">=2.6" },
    { name = "pytest", marker = "extra == 'dev'", specifier = ">=8.3" },
    { name = "pytest-asyncio", marker = "extra == 'dev'", specifier = ">=0.24" },
    { name = "pytest-cov", marker = "extra == 'dev'", specifier = ">=5.0" },
    { name = "python-jose", extras = ["cryptography"], specifier = ">=3.3" },
    { name = "python-multipart", specifier = ">=0.0.17" },
    { name = "pyyaml", specifier = ">=6.0" },
    { name = "redis", specifier = ">=5.2" },
    { name = "ruff", marker = "extra == 'dev'", specifier = ">=0.6" },
    { name = "sqlalchemy", extras = ["asyncio"], specifier = ">=2.0.35" },
    { name = "uvicorn", extras = ["standard"], specifier = ">=0.32" },
]
provides-extras = ["dev"]

[[package]]
name = "googleapis-common-protos"
version = "1.75.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "protobuf" },
]
sdist = { url = "https://files.pythonhosted.org/packages/b5/c8/f439cffde755cffa462bfbb156278fa6f9d09119719af9814b858fd4f81f/googleapis_common_protos-1.75.0.tar.gz", hash = "sha256:53a062ff3c32552fbd62c11fe23768b78e4ddf0494d5e5fd97d3f4689c75fbbd", size = 151035, upload-time = "2026-05-07T08:04:49.423Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e7/c8/e2645aa8ed02fd4c7a2f59d68783b65b1f3cbdfe39a6308e156509d1fee8/googleapis_common_protos-1.75.0-py3-none-any.whl", hash = "sha256:961ed60399c457ceb0ee8f285a84c870aabc9c6a832b9d37bb281b5bebde43ed", size = 300631, upload-time = "2026-05-07T08:03:30.345Z" },
]

[[package]]
name = "greenlet"
version = "3.5.4"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a3/74/b13368064b09053253555d3f2839cc2684d22d5aed0d2ccffbf7a6736558/greenlet-3.5.4.tar.gz", hash = "sha256:0232ae1de90a8e07867bb127d7a6ba2301e859145489f25cda8a6096dabe1d20", size = 206538, upload-time = "2026-07-22T12:47:14.468Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/f3/04/81bd731d6d1e3a469d9a4c36f5eb069bcf0cbb2d5d342c9fec22245b91fc/greenlet-3.5.4-cp312-cp312-macosx_11_0_universal2.whl", hash = "sha256:3d66250e8b09f182ede05490998c818b5961f7a3640332d44c4927caec7bbfe4", size = 295909, upload-time = "2026-07-22T11:38:09.261Z" },
    { url = "https://files.pythonhosted.org/packages/cc/dd/f5f22903a6ae70f5ea328ed0beaec92ad903f0e3b7d2845133b354abc4b8/greenlet-3.5.4-cp312-cp312-manylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:c90e930c9c192e5b3ee9fb8bcd920ea3926155e2e3ded39fc697323addecee17", size = 612011, upload-time = "2026-07-22T12:26:40.69Z" },
    { url = "https://files.pythonhosted.org/packages/8e/10/92a4a88d12b915d74ea5b6d288e4afefda4771647caa34442c156f7a454f/greenlet-3.5.4-cp312-cp312-manylinux_2_24_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:791fdfeeb9c6e0c7b10fa151bf110d2a6974866f13dcb5b1c7efae698245893a", size = 624299, upload-time = "2026-07-22T12:29:02.089Z" },
    { url = "https://files.pythonhosted.org/packages/6c/f9/03e26be3487c5238e81f2b84714959a86ea8515a869828cf41f4fc54b34e/greenlet-3.5.4-cp312-cp312-manylinux_2_24_s390x.manylinux_2_28_s390x.whl", hash = "sha256:b7c895310363f310361e0fe2072af85269d2a2a285cd04c0c59e79a5e3670dcf", size = 629603, upload-time = "2026-07-22T12:43:43.456Z" },
    { url = "https://files.pythonhosted.org/packages/50/6d/0b14bb9db2989f32cd9fe7f76afedea01ee8bee3f87c07e69f24adfe7e63/greenlet-3.5.4-cp312-cp312-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:f88193799d43dbf8c8a806d6405c9c52fe2af40bf75072a606357b33cc336c7f", size = 621541, upload-time = "2026-07-22T11:51:09.464Z" },
    { url = "https://files.pythonhosted.org/packages/57/6b/7c55ca72ef80d57c16c4a55210f82582622462dc4485799a30f4ec6f3372/greenlet-3.5.4-cp312-cp312-manylinux_2_39_riscv64.whl", hash = "sha256:13b980043cb1b3134e81ea469da1250ddcc6bfe6d245bbaa59168d9cdc8f228f", size = 432554, upload-time = "2026-07-22T12:39:51.379Z" },
    { url = "https://files.pythonhosted.org/packages/48/3d/25e9a2d9eb6b2e8b7ca4e80a3a26cb887cce6c8e0a87c921164f11bc5574/greenlet-3.5.4-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:b7a5f095767c4493afcd06067f2bb3b8716e3f3f9e92b99c88e7e99f885b3d4d", size = 1581444, upload-time = "2026-07-22T12:25:03.818Z" },
    { url = "https://files.pythonhosted.org/packages/b9/96/4c9bf2e2c408dcc0556edce69efa9f802e82223573c53240136a086821f1/greenlet-3.5.4-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:42afdc1ab5f66da8c586c32af9224a74a706b4f0ea0dc3a4188a0860a09c65c9", size = 1645842, upload-time = "2026-07-22T11:51:12.295Z" },
    { url = "https://files.pythonhosted.org/packages/b5/41/303ecb26a3a56122c0f4d4073ee078881847bd6b6f463ae0ec57ec20223b/greenlet-3.5.4-cp312-cp312-win_amd64.whl", hash = "sha256:60149df8f462d1b230038e6590c23c3b4768bb5d6c022b3b6e82532b34b0b8a3", size = 247169, upload-time = "2026-07-22T11:38:19.893Z" },
    { url = "https://files.pythonhosted.org/packages/a4/e3/ef56864b4c35fcb3eb3b41b869f6cc46f4cd3f5e2c68e74acde8ac433951/greenlet-3.5.4-cp312-cp312-win_arm64.whl", hash = "sha256:77d6ce04fed0d9aeed42e0f37923cc43eba9b027bdd9c34546bb4ccd143d0fe0", size = 245565, upload-time = "2026-07-22T11:38:27.061Z" },
    { url = "https://files.pythonhosted.org/packages/c0/9a/e51225dcd58713f16ccbdcc501a8da21098ea14515b7870f1f94459e5ff5/greenlet-3.5.4-cp313-cp313-macosx_11_0_universal2.whl", hash = "sha256:24e61b88cb7e1b1d794b32a10cc346ac779681d6d74ff137a3e0a444d2bf1f02", size = 294831, upload-time = "2026-07-22T11:38:53.389Z" },
    { url = "https://files.pythonhosted.org/packages/9f/ea/de50a50fadf979713ab18b46f22ad5ff5f2dcfc637a3ebdecf669801e1a5/greenlet-3.5.4-cp313-cp313-manylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:870d730fec833f5a06906a32596cc099b9161594642a92a520b7a88911c95356", size = 614619, upload-time = "2026-07-22T12:26:42.282Z" },
    { url = "https://files.pythonhosted.org/packages/db/c7/2aae27fea41205b8650294c301f042a2a4bb6155eea48c995b890a92f2c1/greenlet-3.5.4-cp313-cp313-manylinux_2_24_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:ec5ff0d1878df6af3bf9b638a5a92a7d5693291de77c91bff10fa48519c604ef", size = 627021, upload-time = "2026-07-22T12:29:03.445Z" },
    { url = "https://files.pythonhosted.org/packages/1b/80/fb4d4788bbc8e54761f1fc88533af9523a6e86299fa113d6e8a8503ed9fc/greenlet-3.5.4-cp313-cp313-manylinux_2_24_s390x.manylinux_2_28_s390x.whl", hash = "sha256:07bd44616608d873d06735b63ef1a88191d6ca57c8d291d6559c71bc14c0893c", size = 632845, upload-time = "2026-07-22T12:43:45.19Z" },
    { url = "https://files.pythonhosted.org/packages/eb/56/79fd826f9ccaae0b84e1b4ef68dabba5e105bb044ffcd448a0b782fcba9a/greenlet-3.5.4-cp313-cp313-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:d84d993f6e575c950d91a23c1345d18fe1a4310d447bf630849d7809196b52f0", size = 624002, upload-time = "2026-07-22T11:51:11.391Z" },
    { url = "https://files.pythonhosted.org/packages/42/e3/6086fa578ebb72772722cdc4bcd628459814b42e0c2db1e3cbd6552b3271/greenlet-3.5.4-cp313-cp313-manylinux_2_39_riscv64.whl", hash = "sha256:3529a8a933582ad19e224792cac7372489526576b75b4c124e8e4f29948f4861", size = 435053, upload-time = "2026-07-22T12:39:52.715Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/0a/1a/27319f97e731298513dcba1a2e91b63e9d8811d9de22130f960b129b1bf1/greenlet-3.5.4-cp313-cp313-mus
llinux_1_2_aarch64.whl", hash = "sha256:58023945f421093de5e6fa108c0985a8659d43f49e0216da25099369a121bcbd", size = 1581533, upload-time = "2026-07-22T1
2:25:05.322Z" },
    { url = "https://files.pythonhosted.org/packages/b1/6d/24240bf562e9786dd2799ee0a4a4dadb4ded22510f41b20245099159ac8c/greenlet-3.5.4-cp313-cp313-mus
llinux_1_2_x86_64.whl", hash = "sha256:bae2728e1897aa8df8cb1af38cd48b3a743aefe29372de7b8b7a9f532501e69f", size = 1645781, upload-time = "2026-07-22T11
:51:14.805Z" },
    { url = "https://files.pythonhosted.org/packages/c1/5a/442ab1a9ef7ca6bf7210e5397a95972206a91a31033a03c8900866a10039/greenlet-3.5.4-cp313-cp313-win
_amd64.whl", hash = "sha256:ca5726c0b08ca35ae873557266a78b2c3f3b2b7d7401aa5ff886c2045dd0111c", size = 247133, upload-time = "2026-07-22T11:39:20.661Z"
 },
    { url = "https://files.pythonhosted.org/packages/3e/e6/9160210222386b1a378ff94db846b9508ca24a121cf684991561fdb69280/greenlet-3.5.4-cp313-cp313-win
_arm64.whl", hash = "sha256:7c1303791d603080cac6fc3b34df51c3b75b723739c282c8029e48a0d241672f", size = 245500, upload-time = "2026-07-22T11:40:22.185Z"
 },
    { url = "https://files.pythonhosted.org/packages/a5/a7/6ab1d4f9cd548d15ab90da29947f2076100130bb179b0bde59f795a459e3/greenlet-3.5.4-cp314-cp314-mac
osx_11_0_universal2.whl", hash = "sha256:7e8afa5eac028f8140ceafe5ceec66e6aa127ddcb21452d2a564dcd2900b5f22", size = 295410, upload-time = "2026-07-22T1
1:40:35.747Z" },
    { url = "https://files.pythonhosted.org/packages/cd/7a/422f63b4715cbc0b24385305407adf38b48f6bb68b3e6b04090e994d0f5a/greenlet-3.5.4-cp314-cp314-man
ylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:73b37afe369021423ea53dd3123e04bffa7e93ac64429b9f50835b2e4fcae7cf", size = 661286, uplo
ad-time = "2026-07-22T12:26:43.8Z" },
    { url = "https://files.pythonhosted.org/packages/d0/31/5a1cac663bf5582190c5a714ef81364f03cde232227f39748f8ae4c11da5/greenlet-3.5.4-cp314-cp314-man
ylinux_2_24_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:3ef964f56dfcb6f9bbef2a190d9126795eac408716aeae47b5e7c73c32aafca9", size = 673517, uplo
ad-time = "2026-07-22T12:29:04.815Z" },
    { url = "https://files.pythonhosted.org/packages/9c/bf/250c2921c7b585dde12f5239e313ca2dcbc464d161ecca36e4e6ef21762d/greenlet-3.5.4-cp314-cp314-man
ylinux_2_24_s390x.manylinux_2_28_s390x.whl", hash = "sha256:cef589bc65fae02d10bca2ac341191c5b33acc2967892ebf4fcbd10eabb7a74c", size = 677968, upload-t
ime = "2026-07-22T12:43:46.788Z" },
    { url = "https://files.pythonhosted.org/packages/15/4a/2a82a1e3f8aaca020853ac8d12211280ca2b231aa08ea39f636f1060c319/greenlet-3.5.4-cp314-cp314-man
ylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9c53ff01a5c53a40f2c16820ebc56d7c61a77f5fbe009dadd96292d5682f80f8", size = 670917, upload
-time = "2026-07-22T11:51:13.589Z" },
    { url = "https://files.pythonhosted.org/packages/18/40/10bfcf6513558d82f7b95dd728001c63bd388259fe27d3e30ae01f103430/greenlet-3.5.4-cp314-cp314-man
ylinux_2_39_riscv64.whl", hash = "sha256:dfc41ae893d9ceaf22c824f2153a88b30651b20e8758c2cd9ac143f23640563c", size = 480643, upload-time = "2026-07-22T1
2:39:54.149Z" },
    { url = "https://files.pythonhosted.org/packages/68/b0/e379a152b17bfdfa95795af4049e37c0fd1b4d81f020d426db104ed07c77/greenlet-3.5.4-cp314-cp314-mus
llinux_1_2_aarch64.whl", hash = "sha256:ecca4d80d55a01ad6b23b33262662956149fbb7b2c6be2910f1705921958cbf3", size = 1628478, upload-time = "2026-07-22T1
2:25:06.678Z" },
    { url = "https://files.pythonhosted.org/packages/5e/43/bffdfa64f7317f954c5c1230b5dd5922676ce198689a68c1ac1ed4b1b1a5/greenlet-3.5.4-cp314-cp314-mus
llinux_1_2_x86_64.whl", hash = "sha256:2ffbc533e0eaf8e80d8471411646ab88fe58f641d508c0b02b24494479f4d9ec", size = 1692021, upload-time = "2026-07-22T11
:51:17.008Z" },
    { url = "https://files.pythonhosted.org/packages/d0/11/f799f9637e2c6e9b0b716015e339040598b058cf7654dfc0d67468b177ed/greenlet-3.5.4-cp314-cp314-win
_amd64.whl", hash = "sha256:305f69e6c4523d7f6979ed001cff4e5853c063e5da0488029660 3aa0227e544c", size = 248031, upload-time = "2026-07-22T11:40:11.007Z"
 },
    { url = "https://files.pythonhosted.org/packages/05/75/625bcdd74d5e6b2dca1ecba3c3ac77bcf8a026c21a649a46cef23e421f97/greenlet-3.5.4-cp314-cp314-win
_arm64.whl", hash = "sha256:f260930bbbbcf9caee661211235a5111c86dfe5832fdf6ae4570da1e0995320f", size = 246892, upload-time = "2026-07-22T11:40:27.357Z"
 },
    { url = "https://files.pythonhosted.org/packages/ec/69/35c62ed49c320cb4d98e14698ccca5467d3bfe683984172be9cb564d9ce3/greenlet-3.5.4-cp314-cp314t-ma
cosx_11_0_universal2.whl", hash = "sha256:41ddab54e4b238f4a6c323f39b4e59e176affd5a94d461a9fb7583dac74240a3", size = 305571, upload-time = "2026-07-22T
11:40:31.659Z" },
    { url = "https://files.pythonhosted.org/packages/5b/6c/64d60216b3640dcb0b62d913dd9e0d80030c09115bb2e4ba70c95d10ca45/greenlet-3.5.4-cp314-cp314t-ma
nylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:b3dabe3e2809013052c68bdf0b7fa5f5f2859c43a80803131ad61af9cabd7867", size = 672568, upl
oad-time = "2026-07-22T12:26:45.298Z" },
    { url = "https://files.pythonhosted.org/packages/5c/de/ba3ab0a96292e53039530333b0d2ae18d9e508f3a325cd7bf15f8172944c/greenlet-3.5.4-cp314-cp314t-ma
nylinux_2_24_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:27d3f00718634d4520a3a150154ac5da36f257869d41321953375b90bfbbc72c", size = 680076, upl
oad-time = "2026-07-22T12:29:06.125Z" },
    { url = "https://files.pythonhosted.org/packages/ae/db/24a10af12bf8e639cec46c38b9ce1a282543ba42ff4fb0b31a970f1ab603/greenlet-3.5.4-cp314-cp314t-ma
nylinux_2_24_s390x.manylinux_2_28_s390x.whl", hash = "sha256:39169a11d87a6a263afda3e9a27d1df16d0f919d40a4837cc73986c9884c0dd8", size = 681690, upload-
time = "2026-07-22T12:43:48.109Z" },
    { url = "https://files.pythonhosted.org/packages/a5/be/aeada79083c6f1c15f45d77a332f9c441af263ee298e3eb17522cd337d22/greenlet-3.5.4-cp314-cp314t-ma
nylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:cbd60b5763c6543c1827e48faaf14ea9bfbad245f52b1a4d76a2a2d8884c6c66", size = 676733, uploa
d-time = "2026-07-22T11:51:16.027Z" },
    { url = "https://files.pythonhosted.org/packages/f4/60/44a2eca7b9fd71ae0fae7ff184da1cd3169d176652b97aa1cffcbb0ef961/greenlet-3.5.4-cp314-cp314t-ma
nylinux_2_39_riscv64.whl", hash = "sha256:bd3d1145f603b2db19feb9078c2e6855eb7c67e15580c010ed815cee519b86fd", size = 510263, upload-time = "2026-07-22T
12:39:55.678Z" },
    { url = "https://files.pythonhosted.org/packages/e9/10/2392fc3a98948652ef5fd1e7275c04f861dd13f74b78a2b4309f4ee4d090/greenlet-3.5.4-cp314-cp314t-mu
sllinux_1_2_aarch64.whl", hash = "sha256:f00f910f0e7b35416c63b23ad78b769aeccfc1775f712b43c4ee525624a2eef7", size = 1637327, upload-time = "2026-07-22T
12:25:07.879Z" },
    { url = "https://files.pythonhosted.org/packages/55/c6/e7237a3dfa1f205ed0d9ea1e46d70bd2811b32d516266399fb59d28ab90a/greenlet-3.5.4-cp314-cp314t-mu
sllinux_1_2_x86_64.whl", hash = "sha256:91c26423753b92caf41ab3f98fd547d7374d4d9fc2d85be041886c1579d9255e", size = 1697493, upload-time = "2026-07-22T1
1:51:19.214Z" },
    { url = "https://files.pythonhosted.org/packages/55/e3/4ba8154ba2a3d43729e499f72471b4b5c993f3826d3e24da81d5f06d6572/greenlet-3.5.4-cp314-cp314t-wi
n_amd64.whl", hash = "sha256:ee032b91fd8ec29ec6c4cea2b8c561b178435134bd0752c7334b94e9c736c132", size = 251637, upload-time = "2026-07-22T11:40:37.44Z"
 },
    { url = "https://files.pythonhosted.org/packages/90/03/e3f96dfc100261a29545ddc8270cafe58f9195b6651466910e820910de77/greenlet-3.5.4-cp315-cp315-mac
osx_11_0_universal2.whl", hash = "sha256:178111881dd7a6c946471fda85485ec796e1043c2b939f694b096e2ecf986809", size = 296076, upload-time = "2026-07-22T1
1:39:38.364Z" },
    { url = "https://files.pythonhosted.org/packages/a4/3d/da52d208e5c977bce8667e784729e584e38b5785f4c1ec0f4c836e9a1c42/greenlet-3.5.4-cp315-cp315-man
ylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:d92df08dd65fede97fc37aad36c2e9dcda3b31c467f8e0c2c096456cb818e927", size = 666870, uplo
ad-time = "2026-07-22T12:26:46.691Z" },
    { url = "https://files.pythonhosted.org/packages/2d/8a/7e6dee25cb8a8cf9b362c8e597cc269593378bd916f16c736c059a52e85a/greenlet-3.5.4-cp315-cp315-man
ylinux_2_24_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:99e8f8c4ebc4fd80aa26c1280ae9ad43a0976e786349703a181cf0bae60413e5", size = 677678, uplo
ad-time = "2026-07-22T12:29:07.508Z" },
    { url = "https://files.pythonhosted.org/packages/51/a7/dafc7415d430b0a43a16396eb49ecb3b62fd720877fb259cc4dcfaf5f31e/greenlet-3.5.4-cp315-cp315-man
ylinux_2_24_s390x.manylinux_2_28_s390x.whl", hash = "sha256:1f17e362d78e37559e0506c5a7d066bdd45073c36a0127a543e8a0df27242ff3", size = 681428, upload-t
ime = "2026-07-22T12:43:49.623Z" },
    { url = "https://files.pythonhosted.org/packages/6c/21/5a38699fa45de749e3857d93b8f07e4c20489e77c2d35d915a2e1c456606/greenlet-3.5.4-cp315-cp315-man
ylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:394de08dad5ffcb1f50c2159d93e398d9d2da3ed437645eaa54771fa720db9f0", size = 676067, upload
-time = "2026-07-22T11:51:18.163Z" },
    { url = "https://files.pythonhosted.org/packages/2e/d9/6298f3432de301d4718766cf934bd73c418c73f81fbb77247319364b0d96/greenlet-3.5.4-cp315-cp315-man
ylinux_2_39_riscv64.whl", hash = "sha256:cd320d998cbaa032932830448e39abf3c6a12901295e386e8114db926e10cffb", size = 487446, upload-time = "2026-07-22T1
2:39:57.044Z" },
    { url = "https://files.pythonhosted.org/packages/5e/ba/863116ab8ff1ca7a729e327800268939d182db47aa433db70e216e7d9194/greenlet-3.5.4-cp315-cp315-mus
llinux_1_2_aarch64.whl", hash = "sha256:c883d61f2282d72c767a14936641b3efcbde9d82f1080712aaea0b1d3126cb88", size = 1633489, upload-time = "2026-07-22T1
2:25:09.605Z" },
    { url = "https://files.pythonhosted.org/packages/fa/06/7466ced82818d6132462d7f26b3f83c66ea15d2b193a6c0088d558ed7d95/greenlet-3.5.4-cp315-cp315-mus
llinux_1_2_x86_64.whl", hash = "sha256:2a924f15d17957e252a810acefcb5942f5ca712298e8b6fcaed9a307d357522c", size = 1696584, upload-time = "2026-07-22T11
:51:21.304Z" },
    { url = "https://files.pythonhosted.org/packages/f9/4d/55b638489260065de9ffce606c8b5d04507bef705de4b212a0c3d6a1a0df/greenlet-3.5.4-cp315-cp315-win
_amd64.whl", hash = "sha256:ed17e5f3420360d5b459de8462efb52060399a5326a613d4cde31cef63ef95da", size = 248297, upload-time = "2026-07-22T11:42:04.055Z"
 },
    { url = "https://files.pythonhosted.org/packages/bb/08/9dd4ae635da93d41dc268bc34bd62a9d711ed8b8825c5d22ac910c7d6e6d/greenlet-3.5.4-cp315-cp315-win
_arm64.whl", hash = "sha256:f908898d6fa484ce4b6f447ce70ea99b52c503fee419e53cf74d60a16bc9e667", size = 247423, upload-time = "2026-07-22T11:44:00.764Z"
 },
    { url = "https://files.pythonhosted.org/packages/19/66/7c87ed9cdbf1d49c2c6cd1c7b9dd4d16c33b24235ca03972293a1876b30c/greenlet-3.5.4-cp315-cp315t-ma
cosx_11_0_universal2.whl", hash = "sha256:1833637f17d5e7472548a48575c394fe39f1b1890d676d162d86593610f44d8c", size = 306487, upload-time = "2026-07-22T
11:41:25.118Z" },
    { url = "https://files.pythonhosted.org/packages/5b/05/0a4201e7c0054866eefc05da234f236dd4c950d0fbf9ca0517141f01b269/greenlet-3.5.4-cp315-cp315t-ma
nylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:12cda9122e03341f1cb6b8207a19d7a9d375e52f1b4e9243918375f40fd7b4b9", size = 676479, upl
oad-time = "2026-07-22T12:26:48.129Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/3d/c0/4b6b8c5a3aec70f0649cd89662d120fdd6421e2bdc8e3b15c3ab5ec568d8/greenlet-3.5.4-cp315-cp315t-manylinux_2_24_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:d83ae0e32d14957ab7170785a20f582635c8474deab1bfbb552b17e769a6ce25", size = 684321, upload-time = "2026-07-22T12:29:08.925Z" },
    { url = "https://files.pythonhosted.org/packages/88/15/0b167aeea95285b0e654ddce651922f666c089363c2ec528ca8b9a9ba74f/greenlet-3.5.4-cp315-cp315t-manylinux_2_24_s390x.manylinux_2_28_s390x.whl", hash = "sha256:123aa379c962ed5fe90a880327e0c3066124ac64ec99e12a238be9fd8eb3db3d", size = 685995, upload-time = "2026-07-22T12:43:50.993Z" },
    { url = "https://files.pythonhosted.org/packages/24/c9/b49c31c9a972eee91e260445770e922244a7efc542697f51a012ec046d0f/greenlet-3.5.4-cp315-cp315t-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9f1467de1bb767f75db0aa34c195e3a496d8d1278c796e70c24ce205d3e99cde", size = 681293, upload-time = "2026-07-22T11:51:20.43Z" },
    { url = "https://files.pythonhosted.org/packages/de/90/c023ec337f32ff505be7db759c80d98f0532bb94d0c6fa13645efe9bee2e/greenlet-3.5.4-cp315-cp315t-manylinux_2_39_riscv64.whl", hash = "sha256:adf2244d7f69409925a8f22ed22cc5f93cdfe5c9dc87ff3476be2c2aaae61a05", size = 516928, upload-time = "2026-07-22T12:39:58.359Z" },
    { url = "https://files.pythonhosted.org/packages/95/6f/7f2d4653770500eee667866016d42d7a68e3d3462f80df6b8e3fcd48a0eb/greenlet-3.5.4-cp315-cp315t-musllinux_1_2_aarch64.whl", hash = "sha256:0fa53040b78b578120eecdc0265e3f1051487cc425d11a2b7c761daadf4feaa8", size = 1642474, upload-time = "2026-07-22T12:25:10.819Z" },
    { url = "https://files.pythonhosted.org/packages/5a/d8/8cba31036a4caae448087ba5d150660ab03b4a0f54d9150f6495a3be7262/greenlet-3.5.4-cp315-cp315t-musllinux_1_2_x86_64.whl", hash = "sha256:60e0bc961d367df506660e9ac0177a76bc6d81305300704b0977d1634f76efe2", size = 1701012, upload-time = "2026-07-22T11:51:23.17Z" },
    { url = "https://files.pythonhosted.org/packages/e3/cd/3f77a4cce3bae631b08eb52f53a82a976669600337e21dfdba811cb50267/greenlet-3.5.4-cp315-cp315t-win_amd64.whl", hash = "sha256:f680e549edb3eaf21eea4e7fe101e15ec180c74b7879ab46adc080f22d4015d2", size = 251977, upload-time = "2026-07-22T11:41:38.125Z" },
    { url = "https://files.pythonhosted.org/packages/93/e8/65e8707d00fe2a49bf12f609a9b2b39ba6dd23c2810eacad877c4fc94bfe/greenlet-3.5.4-cp315-cp315t-win_arm64.whl", hash = "sha256:08fc36de8442d5c3e95b044550dbea9bf144d31ec0cc58e36fb241cb6ef6a994", size = 250538, upload-time = "2026-07-22T11:40:17.985Z" },
]

[[package]]
name = "griffe"
version = "2.1.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "griffecli" },
    { name = "griffelib" },
]
sdist = { url = "https://files.pythonhosted.org/packages/00/44/63913c007814cab5ba9d36f25ad40dfc640c2e2931d195bd2d05f774a5d6/griffe-2.1.0.tar.gz", hash = "sha256:c58845df5a364feaabd05ee8c767b97b03e478da8aa18b9923553c812fb0d955", size = 244879, upload-time = "2026-06-19T12:05:41.262Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/6b/fb/3c65d392feae6c36dc2b55a14dd8270b7b35a3171c93b71a4d2ee4abf241/griffe-2.1.0-py3-none-any.whl", hash = "sha256:2ccdab17fb9cd76f278d7b5611cfc8f68cbe846d8d48df63dff80b62ecfa6f65", size = 5140, upload-time = "2026-06-19T12:05:39.913Z" },
]

[[package]]
name = "griffecli"
version = "2.1.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "colorama" },
    { name = "griffelib" },
]
sdist = { url = "https://files.pythonhosted.org/packages/8e/c6/90f85d47af96300d629b38c25b71aad9467a620cac964a39280e822efc8a/griffecli-2.1.0.tar.gz", hash = "sha256:2ff68dbee9395fdb668b10374c51683392d697b226ac60159798f4add1ee716c", size = 56913, upload-time = "2026-06-19T12:05:43.119Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/80/2f/513232ec1d5f5da182e4ce45a11427e37dd210844a9e2bca451fc9661fb3/griffecli-2.1.0-py3-none-any.whl", hash = "sha256:6e22b1423d562ddc510997b4be1fe89de59e19dcff78831c0f4bfc3b8134a718", size = 9500, upload-time = "2026-06-19T12:05:37.517Z" },
]

[[package]]
name = "griffelib"
version = "2.1.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/33/e4/8d187ea29c2e30b3a09505c567513077d6117861bde1fbd997a167f262ec/griffelib-2.1.0.tar.gz", hash = "sha256:762a186d2c6fd6794d4ea20d428d597ffb857cb56b66421651cbba15bdd5e813", size = 216234, upload-time = "2026-06-19T12:05:42.278Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e4/d3/5268aeabf2ad82658c4e2ff3a060648d0f02f3926cb53247c0e4d0dab49e/griffelib-2.1.0-py3-none-any.whl", hash = "sha256:cc7b3d2d2865ad0b909fcc38086e3f554b5ea7acbaa7bbb7ecaa3f5dfb7d9f00", size = 142560, upload-time = "2026-06-19T12:05:38.742Z" },
]

[[package]]
name = "gunicorn"
version = "26.0.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "packaging" },
]
sdist = { url = "https://files.pythonhosted.org/packages/6d/b7/a4a3f632f823e432ce6bc65f62961b7980c898c77f075a2f7118cb3846fe/gunicorn-26.0.0.tar.gz", hash = "sha256:ca9346f85e3a4aeeb64d491045c16b9a35647abd37ea15efe53080eb8b090baf", size = 727286, upload-time = "2026-05-05T06:38:25.529Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e6/40/9c2384fc2be4ad25dd4a49decd5ad9ea5a3639814c11bd40ab77cb9f0a14/gunicorn-26.0.0-py3-none-any.whl", hash = "sha256:40233d26a5f0d1872916188c276e21641155111c2853f0c2cd55260aec0d24fc", size = 212009, upload-time = "2026-05-05T06:38:23.007Z" },
]

[[package]]
name = "h11"
version = "0.16.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/01/ee/02a2c011bdab74c6fb3c75474d40b3052059d95df7e73351460c8588d963/h11-0.16.0.tar.gz", hash = "sha256:4e35b956cf45792e4caa5885e69fba00bdbc6ffafbfa020300e549b208ee5ff1", size = 101250, upload-time = "2025-04-24T03:35:25.427Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/04/4b/29cac41a4d98d144bf5f6d33995617b185d14b22401f75ca86f384e87ff1/h11-0.16.0-py3-none-any.whl", hash = "sha256:63cf8bbe7522de3bf65932fda1d9c2772064ffb3dae62d55932da54b31cb6c86", size = 37515, upload-time = "2025-04-24T03:35:24.344Z" },
]

[[package]]
name = "httpcore"
version = "1.0.9"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "certifi" },
    { name = "h11" },
]
sdist = { url = "https://files.pythonhosted.org/packages/06/94/82699a10bca87a5556c9c59b5963f2d039dbd239f25bc2a63907a05a14cb/httpcore-1.0.9.tar.gz", hash = "sha256:6e34463af53fd2ab5d807f399a9b45ea31c3dfa2276f15a2c3f00afff6e176e8", size = 85484, upload-time = "2025-04-24T22:06:22.219Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/7e/f5/f66802a942d491edb555dd61e3a9961140fd64c90bce1eafd741609d334d/httpcore-1.0.9-py3-none-any.whl", hash = "sha256:2d400746a40668fc9dec9810239072b40b4484b640a8c38fd654a024c7a1bf55", size = 78784, upload-time = "2025-04-24T22:06:20.566Z" },
```

```
]

[[package]]
name = "httpcore2"
version = "2.9.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "h11" },
    { name = "truststore" },
]
sdist = { url = "https://files.pythonhosted.org/packages/39/a8/20ed1ed79cbc2ecdf5301c0968ab7c85547212e2a7bd126ddd2d986e206e/httpcore2-2.9.1.tar.gz", h
ash = "sha256:4d8acbf8b306f48c9d6046591fd5ba4037d1b1b1000d140fc2c3eab1e9a0c0e2", size = 67089, upload-time = "2026-07-24T09:21:03.867Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/9f/fb/46c52b781975c335a2bcf1072c7bbc007cbdc8d674217f5ee1daba2c848b/httpcore2-2.9.1-py3-none-any.w
hl", hash = "sha256:6182472379e855fe4221246a2bb7ecede403bc61c6798062ae1787d051ccde26", size = 82809, upload-time = "2026-07-24T09:21:01.178Z" },
]

[[package]]
name = "httptools"
version = "0.8.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/43/e5/d471fcb0e14523fe1c3f4ba58ca52480e7bd70ad7109a3846bc75892f7fb/httptools-0.8.0.tar.gz", h
ash = "sha256:6b2a32f18d97e16e90827d7a819ffa8dbd8cc245fc4e1fa9d1095b54ef4bd999", size = 271342, upload-time = "2026-05-25T22:17:48.841Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/14/88/1d21a36da8f5cb0fa49eafd4b169eba5608d57e75bbcf61845cbc6243216/httptools-0.8.0-cp312-cp312-ma
cosx_10_13_universal2.whl", hash = "sha256:880490234c10f70a9830743097e8958d6e4b9f5a0ffc24515023afeef984054d", size = 208247, upload-time = "2026-05-25
T22:17:07.843Z" },
    { url = "https://files.pythonhosted.org/packages/a5/42/cc4feea2945cb3051038f090c9b36bd5b8a9d7f5a894a506a8983e33fd1c/httptools-0.8.0-cp312-cp312-ma
cosx_11_0_arm64.whl", hash = "sha256:5931891fb7b441b8a3853cf1b85c82c903defce084dd5f6771ca46e31bf862c5", size = 113064, upload-time = "2026-05-25T22:17
:09.136Z" },
    { url = "https://files.pythonhosted.org/packages/e3/a6/febbb8b8db0f58b38e44ad6cb946e6a255ae49b55f2e8543408fb7501ccd/httptools-0.8.0-cp312-cp312-ma
nylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:b15fc622b0f869d19207c4089a501d9bcc63ca5e071ffdd2f03f922df882dcb2", siz
e = 523851, upload-time = "2026-05-25T22:17:10.106Z" },
    { url = "https://files.pythonhosted.org/packages/b7/e4/f90a0df0b83beff265b7e3b65f2a4cefd95792d4be0ac3e16049f2acd3c2/httptools-0.8.0-cp312-cp312-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:425f83884fd6343828d8c565f046cb72b6d19063f6924093e11bcd8e1548cd0
9", size = 518842, upload-time = "2026-05-25T22:17:11.218Z" },
    { url = "https://files.pythonhosted.org/packages/9e/2d/0c9ac76dd2c893841fbf6498d6acec4f2442e1b7067f6e3e316a80e494e8/httptools-0.8.0-cp312-cp312-mu
sllinux_1_2_aarch64.whl", hash = "sha256:ef7c3c97f4311c7be57e2986629df89d49cb434dbff78eafcd48c2bff986b15a", size = 501238, upload-time = "2026-05-25T2
2:17:12.728Z" },
    { url = "https://files.pythonhosted.org/packages/ca/42/906adc91ae3a5fa9c59c0a2f21c139725bd7e5b41ae6acd485cd14123ebf/httptools-0.8.0-cp312-cp312-mu
sllinux_1_2_x86_64.whl", hash = "sha256:a1afd7c9fbff0d9f5d489c4ce2768bd09c84a46ddefc7161e6aa82ae35c85745", size = 509567, upload-time = "2026-05-25T22
:17:13.842Z" },
    { url = "https://files.pythonhosted.org/packages/05/0b/4240efeb672751ee5b9b380cb0e3fdc050bc05f68adc7a8aefc4fcd9a69a/httptools-0.8.0-cp312-cp312-wi
n_amd64.whl", hash = "sha256:cd96f29b4bab1d42fa6e3d008711c75e0f79e94e06827330160e3a304227f150", size = 90918, upload-time = "2026-05-25T22:17:15.155Z"
 },
    { url = "https://files.pythonhosted.org/packages/5e/e5/8cfcabc5546e8022f168be28bcdaa128a240a0befdd03b59d558b4f18bd6/httptools-0.8.0-cp313-cp313-ma
cosx_10_13_universal2.whl", hash = "sha256:614ceea8ea606848bece2338ac03b3ce5324bcb4be8dc7d377ed708012fa4db8", size = 205148, upload-time = "2026-05-25
T22:17:16.333Z" },
    { url = "https://files.pythonhosted.org/packages/2a/0e/0fb14848c19a686c8062ff9067c1a48793e3224b47bc5b201535b6036fce/httptools-0.8.0-cp313-cp313-ma
cosx_11_0_arm64.whl", hash = "sha256:2d689918c15a013c65ef52d9fd495d766893ab831a2c8d89f2ac5940a5df847c", size = 111368, upload-time = "2026-05-25T22:17
:17.586Z" },
    { url = "https://files.pythonhosted.org/packages/2e/1b/46f1cecf06b9bbde8e4b8c88034ac7908989e5ff7a3a388ef38392949c1f/httptools-0.8.0-cp313-cp313-ma
nylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:eb3028cca2fc0a6d720e52ef61d8ebb62fcbfeb1de56874546d858d3f25a26b7", siz
e = 486447, upload-time = "2026-05-25T22:17:18.564Z" },
    { url = "https://files.pythonhosted.org/packages/77/00/258bfc0837221f81d9725c45f9b948a6a6b2994a147a4fb66e85100c668f/httptools-0.8.0-cp313-cp313-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:88bdd940f2b5d487b4d032c6afa5489a7dc4694410d43de3c38c4fb3af0dc45
d", size = 482448, upload-time = "2026-05-25T22:17:19.912Z" },
    { url = "https://files.pythonhosted.org/packages/04/ab/d1cef3b5523f4d272a70f42a776c3169a2dddfe3a54de4b2ce4a36341528/httptools-0.8.0-cp313-cp313-mu
sllinux_1_2_aarch64.whl", hash = "sha256:6a43c9dd399758ccc0531acb0a3c4a6c299ee893ee9400e9c893b7bdcfae0681", size = 464460, upload-time = "2026-05-25T2
2:17:20.882Z" },
    { url = "https://files.pythonhosted.org/packages/ce/48/5d1d072442277bb2b3434e0e60690b8e8c23840ef7de8b6ea54040a536d3/httptools-0.8.0-cp313-cp313-mu
sllinux_1_2_x86_64.whl", hash = "sha256:0770728beb05094c809b98e814edff5fef69d26ad7d21185f2f6d5884a0ba683", size = 471312, upload-time = "2026-05-25T22
:17:22.085Z" },
    { url = "https://files.pythonhosted.org/packages/0d/66/b96623b27e51a68199ef4efdda0613cced9233fe3062ac74e50749c5ad37/httptools-0.8.0-cp313-cp313-wi
n_amd64.whl", hash = "sha256:7685df791fad561384bfb139e77fde27a1ffd93134e016f95a0db424ffbf77b1", size = 90117, upload-time = "2026-05-25T22:17:23.074Z"
 },
    { url = "https://files.pythonhosted.org/packages/1a/12/fa3fbf5f9517b273edea2dc982aa82a8c634091e67c590792b729017bc6f/httptools-0.8.0-cp314-cp314-ma
cosx_10_13_universal2.whl", hash = "sha256:de242a49b5d18e0a8776e654e9f6bf6d89f3875a5c35b425a0e7ce940feb3fd6", size = 206183, upload-time = "2026-05-25
T22:17:24.004Z" },
    { url = "https://files.pythonhosted.org/packages/30/fc/5e7c4cb443370f2090a3aba0453a07384d29ff66b7435bb90e77e1037599/httptools-0.8.0-cp314-cp314-ma
cosx_11_0_arm64.whl", hash = "sha256:159e9ab5f701ccd42e555a12f1ad8ff69702910fc1c996cf2bb66e5fcb7a231b", size = 112079, upload-time = "2026-05-25T22:17
:25.216Z" },
    { url = "https://files.pythonhosted.org/packages/ba/53/771bd891eb0f236f32145d6a1775777ec85745f3cc983a1f23d1a3b8ddfe/httptools-0.8.0-cp314-cp314-ma
nylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:c4a9f1707e4823d54dfec6c33fa3697d302aed536ed352a7ebb5a061ddb869d0", siz
e = 481596, upload-time = "2026-05-25T22:17:26.186Z" },
    { url = "https://files.pythonhosted.org/packages/62/42/94e15bc68ce3d423243c45d7f1b0c7561f13844f97dc52ae23182fb65628/httptools-0.8.0-cp314-cp314-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:d76ad7b951387e3632c8716a9bb03ac5b45c5f16119aa409db0459520887944
e", size = 480865, upload-time = "2026-05-25T22:17:27.542Z" },
    { url = "https://files.pythonhosted.org/packages/1c/7c/fe2980fc03723272e30f135b62360b075f513dfe7cc73aef36c7f04012bd/httptools-0.8.0-cp314-cp314-mu
sllinux_1_2_aarch64.whl", hash = "sha256:a3b7387147361c3fd47a0bde763c5c91b5b4cd4dc9989b8ece84ff436c99843b", size = 463189, upload-time = "2026-05-25T2
2:17:28.546Z" },
    { url = "https://files.pythonhosted.org/packages/15/1b/47fc5fff68acd1bfa20b4734059c9a06cadb88119dcd5258b5b0d21d91c8/httptools-0.8.0-cp314-cp314-mu
sllinux_1_2_x86_64.whl", hash = "sha256:f256d6ce930c52ca1cb2a960b7da03548c454e7d28b06059ad41bfe789036ce0", size = 466610, upload-time = "2026-05-25T22
:17:29.816Z" },
    { url = "https://files.pythonhosted.org/packages/60/bd/07b13c93ffd9bec9546e0d43f8e19378dd696dbd278511406bc07371ef1f/httptools-0.8.0-cp314-cp314-wi
n_amd64.whl", hash = "sha256:19d1ee275bb59ba2643ba9a3a1e51cc0c788caf2b8df506368e03f56fdd08527", size = 92705, upload-time = "2026-05-25T22:17:31.133Z"
 },
    { url = "https://files.pythonhosted.org/packages/fd/c4/121648f68ce066d7bd762d6b6d97e620847642d38d54f3d90ff11d947629/httptools-0.8.0-cp314-cp314t-m
acosx_10_13_universal2.whl", hash = "sha256:de1ed58a974e75d56560acc7e7fed01a454994429456f65209789992e41f2568", size = 215023, upload-time = "2026-05-2
5T22:17:32.401Z" },
    { url = "https://files.pythonhosted.org/packages/b9/b0/312a062ae741ae3e8baa8c8bf20be81b2e67337b259ab4349bebc7b6142e/httptools-0.8.0-cp314-cp314t-m
acosx_11_0_arm64.whl", hash = "sha256:e93c227b595c6926c1acee96891dd9da4be338cfbe82e5cd3bb9d8dd7dc4ac0b", size = 117405, upload-time = "2026-05-25T22:1
7:33.742Z" },
    { url = "https://files.pythonhosted.org/packages/fc/37/fccd705f795386bb05bf413012fecff2a33e5aa8c2f069096de3e9fd8702/httptools-0.8.0-cp314-cp314t-m
anylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:2a021c3a8e65cc125390d72f59b968afca3bdcaff25bd67965e0a055a14946ca", si
ze = 558497, upload-time = "2026-05-25T22:17:34.732Z" },
    { url = "https://files.pythonhosted.org/packages/bd/39/f172e8003576de35f5ba77ff417cf0e34429d35dc014deef15afa337a72c/httptools-0.8.0-cp314-cp314t-m
anylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:48774d39cbb70e2b1f71f88852a3087ae1d3a1eb80482bb48c13067ab080c1
4f", size = 571585, upload-time = "2026-05-25T22:17:35.813Z" },
    { url = "https://files.pythonhosted.org/packages/3e/b9/f5564760af99f3dbbf3f9104dc00e5da27e96cf433c6bdcf77617f70bf3f/httptools-0.8.0-cp314-cp314t-m
usllinux_1_2_aarch64.whl", hash = "sha256:88eead8ec8680a9f146c655bc88445a325bd7921cfd8194c7337e9467282427d", size = 543297, upload-time = "2026-05-25T
22:17:37.08Z" },
    { url = "https://files.pythonhosted.org/packages/99/67/8d9f2c313618e161b82f3873188e7196126da1d6e29688df40eb3997c77a/httptools-0.8.0-cp314-cp314t-m
usllinux_1_2_x86_64.whl", hash = "sha256:2c032fa028f46871ec7e1fc59fc15e8023eab3e6bbe6ece786a1611719a5d081", size = 539535, upload-time = "2026-05-25T2
2:17:38.032Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/48/63/b906c01e53f50d432c0defe43ce52764a111dc1bdd028bafbeb54dcfd008/httptools-0.8.0-cp314-cp314t-w
in_amd64.whl", hash = "sha256:384c17174464c8e873398b7af24f0b1f44d992c820328413951a625323155d77", size = 108209, upload-time = "2026-05-25T22:17:39.473
Z" },
]

[[package]]
name = "httpx"
version = "0.28.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "certifi" },
    { name = "httpcore" },
    { name = "idna" },
]
sdist = { url = "https://files.pythonhosted.org/packages/b1/df/48c586a5fe32a0f01324ee087459e112ebb7224f646c0b5023f5e79e9956/httpx-0.28.1.tar.gz", hash
 = "sha256:75e98c5f16b0f35b567856f597f06ff2270a374470a5c2392242528e3e3e42fc", size = 141406, upload-time = "2024-12-06T15:37:23.222Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/2a/39/e50c7c3a983047577ee07d2a9e53faf5a69493943ec3f6a384bdc792deb2/httpx-0.28.1-py3-none-any.whl"
, hash = "sha256:d909fcccc110f8c7faf814ca82a9a4d816bc5a6dbfea25d6591d6985b8ba59ad", size = 73517, upload-time = "2024-12-06T15:37:21.509Z" },
]

[[package]]
name = "httpx2"
version = "2.9.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "httpcore2" },
    { name = "idna" },
    { name = "truststore" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/21/14/38128fbafd7e0ed41d874df6c9a653d47c2d111cfe59e2b4ac95161b4abd/httpx2-2.9.1.tar.gz", hash
 = "sha256:1932a768737e3666291582833da748cc4e563c337cf96706fccc04fa6e58764a", size = 95458, upload-time = "2026-07-24T09:21:04.972Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/13/b8/cfd91c4ab9134d386d48f0b6ac662ff3d4be6efdee59ee1c67ebc3c0487c/httpx2-2.9.1-py3-none-any.whl"
, hash = "sha256:1820fe14a9ab1107bfeff39259987429450b070ec0ff38cc87eb0d8c97fdc71a", size = 91191, upload-time = "2026-07-24T09:21:02.6Z" },
]

[[package]]
name = "idna"
version = "3.18"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/cd/63/9496c57188a2ee585e0f1db071d75089a11e98aa86eb99d9d7618fc1edce/idna-3.18.tar.gz", hash =
"sha256:ffb385a7e039654cef1ab9ef32c6fafe283c0c0467bba1d9029738ce4a14a848", size = 196711, upload-time = "2026-06-02T14:34:07.794Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/1e/5e/d4e9f1a599fb8e573b7b87160658329fbf28d19eac2718f51fc3def3aa5a/idna-3.18-py3-none-any.whl", h
ash = "sha256:7f952cbe720b688055e3f87de14f5c3e5fdaa8bc3928985c4077ca689de849a2", size = 65455, upload-time = "2026-06-02T14:34:06.319Z" },
]

[[package]]
name = "iniconfig"
version = "2.3.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/72/34/14ca021ce8e5dfedc35312d08ba8bf51fdd999c576889fc2c24cb97f4f10/iniconfig-2.3.0.tar.gz", h
ash = "sha256:c76315c77db068650d49c5b56314774a7804df16fee4402c1f19d6d15d8c4730", size = 20503, upload-time = "2025-10-18T21:55:43.219Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/cb/b1/3846dd7f199d53cb17f49cba7e651e9ce294d8497c8c150530ed11865bb8/iniconfig-2.3.0-py3-none-any.w
hl", hash = "sha256:f631c04d2c48c52b84d0d0549c99ff3859c98df65b3101406327ecc7d53fbf12", size = 7484, upload-time = "2025-10-18T21:55:41.639Z" },
]

[[package]]
name = "jinja2"
version = "3.1.6"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "markupsafe" },
]
sdist = { url = "https://files.pythonhosted.org/packages/df/bf/f7da0350254c0ed7c72f3e33cef02e048281fec7ecec5f032d4aac52226b/jinja2-3.1.6.tar.gz", hash
 = "sha256:0137fb05990d35f1275a587e9aee6d56da821fc83491a0fb838183be43f66d6d", size = 245115, upload-time = "2025-03-05T20:05:02.478Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/62/a1/3d680cbfd5f4b8f15abc1d571870c5fc3e594bb582bc3b64ea099db13e56/jinja2-3.1.6-py3-none-any.whl"
, hash = "sha256:85ece4451f492d0c13c5dd7c13a64681a86afae63a5f347908daf103ce6d2f67", size = 134899, upload-time = "2025-03-05T20:05:00.369Z" },
]

[[package]]
name = "jiter"
version = "0.16.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/1d/1f/10936e16d8860c70698a1aa939a46aa0224813b782bce4e000e637da0b2d/jiter-0.16.0.tar.gz", hash
 = "sha256:7b24c3492c5f4f84a37946ad9cf504910cf6a782d6a4e0689b6673c5894b4a1c", size = 176431, upload-time = "2026-06-29T13:05:13.657Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/83/2b/52ace16ed031354f0539749a49e4bf33797d82bea5137910835fa4b09793/jiter-0.16.0-cp312-cp312-macos
x_10_12_x86_64.whl", hash = "sha256:67c3bc1760f8c99d805dcab4e644027142a53b1d5d861f18780ebdbd5d40b72a", size = 306943, upload-time = "2026-06-29T13:03:
14.035Z" },
    { url = "https://files.pythonhosted.org/packages/94/2e/34957c2c1b661c252ba9bcc60ae0bddc27e0f7202c6073326a13c5390eec/jiter-0.16.0-cp312-cp312-macos
x_11_0_arm64.whl", hash = "sha256:5af7780e4a26bd7d0d989592bf9ef12ebf806b74ab709223ecca37c749872ea9", size = 307779, upload-time = "2026-06-29T13:03:15
.418Z" },
    { url = "https://files.pythonhosted.org/packages/88/6c/59bd309cab4460c54cf1079f3eb7fe7af6a4c895c5c957a53378693bad2b/jiter-0.16.0-cp312-cp312-manyl
inux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:d5bf78d0e05e45cfdd66558893938d59afe3d1b1a824a202039b20e607d25a72", size = 335826, upload-
time = "2026-06-29T13:03:17.11Z" },
    { url = "https://files.pythonhosted.org/packages/3b/8c/f5ef7b65f0df47afa16596969defb281ebb86e96df346d62be6fd853d620/jiter-0.16.0-cp312-cp312-manyl
inux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:f4444a83f946605990c98f625cdd3d2725bfb818158760c5748c653170a20e0e", size = 362573, upload-ti
me = "2026-06-29T13:03:18.781Z" },
    { url = "https://files.pythonhosted.org/packages/2b/0b/ace4354da061ee38844a0c27dc2c21eecd27aea119e8da324bea987522d0/jiter-0.16.0-cp312-cp312-manyl
inux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:3a23f0e4f957e1be65752d2dfac9a5a06b1917af8dc85deb639c3b9d02e31290", size = 457979, upload-
time = "2026-06-29T13:03:20.293Z" },
    { url = "https://files.pythonhosted.org/packages/55/40/c0253d3772eb9dcd8e6606ee9b2d53ec8e5b814589c47f140aa585f21eaa/jiter-0.16.0-cp312-cp312-manyl
inux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:c22a488f7b9218e245a0025a9ba6b100e2e54700831cf4cf16833a27fba3ad01", size = 372302, upload-time
 = "2026-06-29T13:03:21.739Z" },
    { url = "https://files.pythonhosted.org/packages/a8/d2/4839422241aa12860ce597b20068727094ba0bc480723c74924ca5bad483/jiter-0.16.0-cp312-cp312-manyl
inux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:46add52f4ad47a08bfb1219f3e673da972191489a33016edefdb5ea55bfa8c48", size = 343805, upload-ti
me = "2026-06-29T13:03:23.384Z" },
    { url = "https://files.pythonhosted.org/packages/e2/59/e196888a05befdda7dbe299b722d56f2f6eec65402bc34c0a3306d595feb/jiter-0.16.0-cp312-cp312-manyl
```

inux_2_31_riscv64.whl", hash = "sha256:9c8a956fd72c2cf1e730d01ea080341f13aa0a97a4a33b51abebe725b7ae9ca9", size = 351107, upload-time = "2026-06-29T13:03:24.815Z" },
    { url = "https://files.pythonhosted.org/packages/ec/74/4cd9e0fca65232136400354b630fbfcd2de634e22ccbb96567725981b548/jiter-0.16.0-cp312-cp312-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:561926e0573ffe4a32498420a76d64b16c513e1ab413b9d28158a8764ac701e5", size = 388441, upload-time = "2026-06-29T13:03:26.266Z" },
    { url = "https://files.pythonhosted.org/packages/d9/8c/554691e48bc711299c0a293dd8a6179e24b2d66a54dc295421fcf64569c0/jiter-0.16.0-cp312-cp312-musllinux_1_1_aarch64.whl", hash = "sha256:44d019fa8cdaf89bf29c71b39e3712143fdd0ac76725c6ef954f9957a5ea8730", size = 516354, upload-time = "2026-06-29T13:03:28.02Z" },
    { url = "https://files.pythonhosted.org/packages/a4/cb/01e9d69dc2cc6759d4f91e230b34489c4fdb2518992650633f9e20bece89/jiter-0.16.0-cp312-cp312-musllinux_1_1_x86_64.whl", hash = "sha256:0df91907609837f33341b8e6fe73b95991fdaa57caf1a0fbd343dffe826f386f", size = 547880, upload-time = "2026-06-29T13:03:29.534Z" },
    { url = "https://files.pythonhosted.org/packages/79/70/2953195f1c6ad00f49fa67e13df7e60acb3dd4f387101bc15abccddd905e/jiter-0.16.0-cp312-cp312-win32.whl", hash = "sha256:51d7b836acb0108d7c77df1742332cac2a1fa04a74d6dacec46e7091f0e91274", size = 203473, upload-time = "2026-06-29T13:03:31.025Z" },
    { url = "https://files.pythonhosted.org/packages/2d/05/2909a8b10699a4d560f8c502b6b2c5f3991b682b1922c1eedda242b225bd/jiter-0.16.0-cp312-cp312-win_amd64.whl", hash = "sha256:1878349266f8ee36ecb1375cc5ba2f115f35fd9f0a1a4119e725e379126647f7", size = 196905, upload-time = "2026-06-29T13:03:32.472Z" },
    { url = "https://files.pythonhosted.org/packages/e9/a9/6b82bb1c8d7790d602489b967b982a909e5d092875a6c2ade96444c8dfc5/jiter-0.16.0-cp312-cp312-win_arm64.whl", hash = "sha256:2ed5738ae4af18271a51a528b8811b0cbfa4a1858de9d83359e4169855d6a331", size = 190618, upload-time = "2026-06-29T13:03:34.672Z" },
    { url = "https://files.pythonhosted.org/packages/91/c0/555fc60473d30d66894ba825e63615e3be7524fac23858356afa7a38906c/jiter-0.16.0-cp313-cp313-macosx_10_12_x86_64.whl", hash = "sha256:41977aa5654023948c2dae2a81cbf9c43343954bef1cd59a154dd15a4d84c195", size = 306203, upload-time = "2026-06-29T13:03:36.243Z" },
    { url = "https://files.pythonhosted.org/packages/d0/2b/c3eaf16f5d7c9bad66ea32f40a95bd169b29a91217fcc7f081375157e99c/jiter-0.16.0-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:d28bb3c26762358dadf3e5bf0bccd29ae987d65e6988d2e6f49829c76b003c09", size = 306489, upload-time = "2026-06-29T13:03:37.846Z" },
    { url = "https://files.pythonhosted.org/packages/96/3f/02fdfc6705cad96127d883af5c34e4867f554f29ec7705ec1a46156400a9/jiter-0.16.0-cp313-cp313-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:0542a7189c26920778658fc8fcf2af8bae05bae9924577f71804acef37996536", size = 335453, upload-time = "2026-06-29T13:03:39.221Z" },
    { url = "https://files.pythonhosted.org/packages/b2/a6/e4bda5920d4b0d7c5dfb7174ce4a6b2e4d3e11c9162c452ef0eab4cdbdbd/jiter-0.16.0-cp313-cp313-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:8fb8de1e23a0cb2a7f53c335049c7b72b6db41aa6227cdcc0972a1de5cb39450", size = 361625, upload-time = "2026-06-29T13:03:40.597Z" },
    { url = "https://files.pythonhosted.org/packages/b7/97/4e6b59b2c6e55cbb3e183595f81ad65dcfb21c915fee5e19e335df21bc55/jiter-0.16.0-cp313-cp313-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:b72d0b2990ca754a9102779ac98d8597b7cb31678958562214a007f909eab78e", size = 456958, upload-time = "2026-06-29T13:03:42.074Z" },
    { url = "https://files.pythonhosted.org/packages/15/e0/97e9557686d2f94f4b93786eccb7eed28e9228ad132ea8237f44727314a7/jiter-0.16.0-cp313-cp313-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:d5f91b1c27fc22a57993d5a5cb8a627cb8ed4b10502716fac1ffbfe1d19d84e8", size = 372017, upload-time = "2026-06-29T13:03:43.658Z" },
    { url = "https://files.pythonhosted.org/packages/0f/94/db768b6938e0df35c86beeba3dfbbb025c9ee5c19e1aa271f2396e50864d/jiter-0.16.0-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:c682bea068a90b764577bdb78a60a4c1d1606daf9cd4c893832a37c7cc9d9026", size = 343320, upload-time = "2026-06-29T13:03:45.226Z" },
    { url = "https://files.pythonhosted.org/packages/c1/d6/5a59d938244a30735fe62d9433fd325f9021ea29d89780ea4596ea93bc89/jiter-0.16.0-cp313-cp313-manylinux_2_31_riscv64.whl", hash = "sha256:8d031aabecc4f1b6276adfb42e3aabb77c89d468bf616600e8d3a11328929053", size = 350520, upload-time = "2026-06-29T13:03:46.671Z" },
    { url = "https://files.pythonhosted.org/packages/67/f8/c4a857f49c9af125f6bbcac7e3eee7f7978ed89682833062e2dbf62576b1/jiter-0.16.0-cp313-cp313-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:eab2cd170150e70153de16896a1774e3a1dca80154c56b54d7a812c479a7165e", size = 387550, upload-time = "2026-06-29T13:03:48.361Z" },
    { url = "https://files.pythonhosted.org/packages/8b/d6/5fbc2f7d6b67b754caa61a993a2e626e815dec47ffc2f9e35f01adfebec7/jiter-0.16.0-cp313-cp313-musllinux_1_1_aarch64.whl", hash = "sha256:6edb63a46e65a82c26800a868e49b2cac30dd5a4218b88d74bc2c848c8ad60bb", size = 515424, upload-time = "2026-06-29T13:03:49.881Z" },
    { url = "https://files.pythonhosted.org/packages/ed/54/284f0164b64a5fed915fea6ba7e9ba9b3d8d37c67d59cf2e3bb99d45cdfe/jiter-0.16.0-cp313-cp313-musllinux_1_1_x86_64.whl", hash = "sha256:659039cc50b5addcc35fcc87ae2c1833b7c0a8e5326ef631a75e4478447bcf84", size = 546981, upload-time = "2026-06-29T13:03:51.363Z" },
    { url = "https://files.pythonhosted.org/packages/13/c5/2a467585a576594384e1d2c43e1224deaafc085f24e243529cf98beef8e1/jiter-0.16.0-cp313-cp313-win32.whl", hash = "sha256:c9c53be232c2e206ef9cdbad81a48bfa74c3d3f08bcf8124630a8a748aad993e", size = 202853, upload-time = "2026-06-29T13:03:53.015Z" },
    { url = "https://files.pythonhosted.org/packages/88/6a/de61d04b9eec69c71719968d2f716532a3bc121170c44a39e14979c6be81/jiter-0.16.0-cp313-cp313-win_amd64.whl", hash = "sha256:baad945ed47f163ad833314f8e3288c396118934f94e7bbb9e243ce4b341a4fd", size = 196160, upload-time = "2026-06-29T13:03:54.447Z" },
    { url = "https://files.pythonhosted.org/packages/19/4b/b390ed59bafb3f31d008d1218578f10327714484b334439947f7e5b11e7f/jiter-0.16.0-cp313-cp313-win_arm64.whl", hash = "sha256:3c1fd2dbe1b0af19e987f03fe66c5f5bd105a2229c1aff4ab14890b24f41d21a", size = 189862, upload-time = "2026-06-29T13:03:55.754Z" },
    { url = "https://files.pythonhosted.org/packages/a7/89/bc4f1b57d5da938fd344a466396541e586d161320d70bffd929aaafcd8f4/jiter-0.16.0-cp314-cp314-macosx_10_12_x86_64.whl", hash = "sha256:b2c61484666ad42726029af0c00ef4541f0f3b5cdc550221f56c2343208018ee", size = 308239, upload-time = "2026-06-29T13:03:57.205Z" },
    { url = "https://files.pythonhosted.org/packages/65/7a/c415453e5213001bf3b411ff65dec3d303b0e76a4a2cfea9768cd4960994/jiter-0.16.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:63efadc657488f45db1c676d81e704cac2abf3fdb892def1faea61db053127e2", size = 308928, upload-time = "2026-06-29T13:03:58.643Z" },
    { url = "https://files.pythonhosted.org/packages/11/fc/1f4fb7ebf9a724c7741994f4aae18fba1e2f3133df14521a79194952c34a/jiter-0.16.0-cp314-cp314-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:cf0d73f50e7b6935677854f6e8e31d499ca7064dd24734f703e060f5b237d883", size = 336998, upload-time = "2026-06-29T13:04:00.071Z" },
    { url = "https://files.pythonhosted.org/packages/a0/8d/72cadaac05ccfa7cc3a0a2232862e6c72443ca40cf300ba8b57f9f18b69b/jiter-0.16.0-cp314-cp314-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:bf3ea07d9bc8e7d03a9fbc051295462e6dbc295b894fd72457c3136e3e43d898", size = 362112, upload-time = "2026-06-29T13:04:01.52Z" },
    { url = "https://files.pythonhosted.org/packages/58/4a/c4b0d5f651fda90a24ffce9f8d56cde462a2e09d31ae3de3c68cef34c04e/jiter-0.16.0-cp314-cp314-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:26798522707abb47d767db536e4148ceac1b14446bf028ee85e579a2e043cfe5", size = 459807, upload-time = "2026-06-29T13:04:03.214Z" },
    { url = "https://files.pythonhosted.org/packages/80/58/ef77879ea9aa56b50824edc5a445e226422c7a8d211f3fd2a56bcb9493cf/jiter-0.16.0-cp314-cp314-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:bc837c1b9631be10abfe0191537fe8009838204cec7e44827401ace390ddb567", size = 373181, upload-time = "2026-06-29T13:04:04.629Z" },
    { url = "https://files.pythonhosted.org/packages/49/2e/ffbc3f254e4d8a66da3062c624a7df4b7c2b2cf9e1fe43cf394b3e104041/jiter-0.16.0-cp314-cp314-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:49060fd70737fad59d33ba9dcc0d83247dc9e77187de26053a19c16c9f32bd69", size = 344927, upload-time = "2026-06-29T13:04:06.067Z" },
    { url = "https://files.pythonhosted.org/packages/9a/f6/0be5dc6d64a89f80aa8fec984f94dedb2973e251edcae55841d60786d578/jiter-0.16.0-cp314-cp314-manylinux_2_31_riscv64.whl", hash = "sha256:adbb8edeadd431bc4477879d5d371ece7cb1334486584e0f252656dd7ffada29", size = 352754, upload-time = "2026-06-29T13:04:07.477Z" },
    { url = "https://files.pythonhosted.org/packages/da/6e/7d31243b3b91cd261dd19e9d3557fc3251a80883d3d8049c86174e7ab7af/jiter-0.16.0-cp314-cp314-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:31aaee5b80f672c1dc21272bcfb9cbdcfc1ea04ff50f00ed5af500b80c44fa93", size = 390553, upload-time = "2026-06-29T13:04:08.92Z" },
    { url = "https://files.pythonhosted.org/packages/25/33/51ae371fde3c88897520f62b4d5f8b27ad7103e2bb10812ff52195609853/jiter-0.16.0-cp314-cp314-musllinux_1_1_aarch64.whl", hash = "sha256:6722bcef4ffc86c835574b1b2fac6b33b9fb4a889c781e67950e891591f3c55a", size = 516900, upload-time = "2026-06-29T13:04:10.407Z" },
    { url = "https://files.pythonhosted.org/packages/a0/45/6449b3d123ea439ba79507c657288f461d55049e7bcbdc2cf8eb8210f491/jiter-0.16.0-cp314-cp314-musllinux_1_1_x86_64.whl", hash = "sha256:5ab4f50ff971b611d656554ea10b75f80097392c827bc32923c6eeb6386c8b00", size = 548754, upload-time = "2026-06-29T13:04:12.046Z" },
    { url = "https://files.pythonhosted.org/packages/9b/e7/fd2fb11ae3e2649333da3aa170d04d7b3000bbdc3b270f6513382fdf4e04/jiter-0.16.0-cp314-cp314-pyemscripten_2026_0_wasm32.whl", hash = "sha256:710cc51d4ebdcd3c1f70b232c1db1ea1344a075770422bbd4bede5708335acbe", size = 122381, upload-time = "2026-06-29T13:04:13.413Z" },
    { url = "https://files.pythonhosted.org/packages/26/80/f0b147a62c315a164ed2168908286ca302310824c218d3aae52b06c0c9a9/jiter-0.16.0-cp314-cp314-win32.whl", hash = "sha256:57b37fc887a32d44798e4d8ebfa7c9683ff3da1d5bf38f08d1bb3573ccb39106", size = 204578, upload-time = "2026-06-29T13:04:14.813Z" },
    { url = "https://files.pythonhosted.org/packages/5e/e6/4758a14304b4523a6f5adb2419340086aa3593bd4327c2b25b5948a90548/jiter-0.16.0-cp314-cp314-win_amd64.whl", hash = "sha256:cbd18dd5e2df96b580487b5745adf57ef64ad89ba2d9662fc3c19386acce7db8", size = 198154, upload-time = "2026-06-29T13:04:16.272Z" },
    { url = "https://files.pythonhosted.org/packages/26/be/41fa54a2e7ea41d6c99f1dc5b1f0fd4cb474680304b5d268dd518e81da3a/jiter-0.16.0-cp314-cp314-win_arm64.whl", hash = "sha256:a32d2027a9fa67f109ff245a3252ece3ccc32cc56703e1deab6cc846a59e0585", size = 191458, upload-time = "2026-06-29T13:04:17.707Z" },
    { url = "https://files.pythonhosted.org/packages/81/6b/59127338b86d9fe4d99418f5a15118bea778103ee0fe9d9dd7e0af174e95/jiter-0.16.0-cp314-cp314t-maco

```
sx_11_0_arm64.whl", hash = "sha256:2577196f4474ef3fc4779a088a23b0897bbf86f9ea3679c372d45b8383b43207", size = 316739, upload-time = "2026-06-29T13:04:19.663Z" },
    { url = "https://files.pythonhosted.org/packages/2d/95/49461034d5388196d3dabf98748935f017b7785d8f3f5349f834bcc4ed0d/jiter-0.16.0-cp314-cp314t-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:616e89e008a93c01104161c75b4988e58716b01d62307ebfe161e52a56d2a818", size = 340911, upload-time = "2026-06-29T13:04:21.257Z" },
    { url = "https://files.pythonhosted.org/packages/cd/97/a4369f2fb82cb3dda13b98622f31249b2e014b223fe64ee534413ad72294/jiter-0.16.0-cp314-cp314t-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:0e2e9efbe042210df657bade597f66d6d75723e3d8f45a12ea6d8167ff8bbce3", size = 361747, upload-time = "2026-06-29T13:04:22.677Z" },
    { url = "https://files.pythonhosted.org/packages/28/51/49b6ed456261646e1906016a6760367a28aacd3c24805e4e5fe64116c1db/jiter-0.16.0-cp314-cp314t-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:3f4d9e473a5ce7d27fef8b848df4dc16e283893d3f53b4a585e72c9595f3c284", size = 460225, upload-time = "2026-06-29T13:04:24.441Z" },
    { url = "https://files.pythonhosted.org/packages/33/b5/5689aff4f66c5b60be63106e591dbfcba2190df97d2c9c7cf052361ddb98/jiter-0.16.0-cp314-cp314t-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:8d30a4a1c87713060c8d1cc59a7b6c8fb6b8ef0a6900368014c76c87922a2929", size = 373169, upload-time = "2026-06-29T13:04:25.884Z" },
    { url = "https://files.pythonhosted.org/packages/a2/96/3ae1b85ee0d6d6cab254fb7f8da018272b932bbf2d69b07e98aa2a96c746/jiter-0.16.0-cp314-cp314t-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:bae96332410f866e5900d809298b1ed82735932986c672495f9701daacd80620", size = 350332, upload-time = "2026-06-29T13:04:27.302Z" },
    { url = "https://files.pythonhosted.org/packages/15/32/c99d7bafd78986556c95bf60ce84c6cc98786eac56066c12d7f828bb6747/jiter-0.16.0-cp314-cp314t-manylinux_2_31_riscv64.whl", hash = "sha256:da3d7ec75dc83bb18bca888b5edfae0656a26849056c59e05a7728badd17e7af", size = 353377, upload-time = "2026-06-29T13:04:28.731Z" },
    { url = "https://files.pythonhosted.org/packages/0e/4b/f99a8e571287c3dec766bcc18528bbe8e8fb5365522ab5e6d64c93e87066/jiter-0.16.0-cp314-cp314t-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:ee6162b77d49a9939229df666dfa8af3e656b6701b54c4c84966d740e189264e", size = 387746, upload-time = "2026-06-29T13:04:30.319Z" },
    { url = "https://files.pythonhosted.org/packages/75/69/c78a5b3f71040e34eb5917df26fb7ae9a2174cad1ccbf277512507c53a6e/jiter-0.16.0-cp314-cp314t-musllinux_1_1_aarch64.whl", hash = "sha256:63ffdbdae7d4499f4cda14eadc12ddcabef0fc0c081191bdc2247489cb698077", size = 517292, upload-time = "2026-06-29T13:04:31.709Z" },
    { url = "https://files.pythonhosted.org/packages/c2/f7/095b38eda4c70d03651c403f29a5590f16d12ddc5d544aac9f9cddf72277/jiter-0.16.0-cp314-cp314t-musllinux_1_1_x86_64.whl", hash = "sha256:a111256a7193bea0759267b10385e5870949c239ed7b6ddbaaf57573edb38734", size = 549259, upload-time = "2026-06-29T13:04:33.721Z" },
    { url = "https://files.pythonhosted.org/packages/2e/c5/6a0207d90e5f656d95af98ebd0934f382d37674416f215aeda2ff8063e51/jiter-0.16.0-cp314-cp314t-win32.whl", hash = "sha256:de5ba8763e56b793561f43bed197c9ea55776daa5e9a6b91eed68a909bc9cdbf", size = 206523, upload-time = "2026-06-29T13:04:35.068Z" },
    { url = "https://files.pythonhosted.org/packages/a5/31/c757d5f30a8980fd945ce7b98be10be9e4ff59c7c42f5fd86804c2e87db8/jiter-0.16.0-cp314-cp314t-win_amd64.whl", hash = "sha256:b8a3f9a6008048fe9def7bf465180564a6e458047d2ce499149cfbe73c3ae9db", size = 200366, upload-time = "2026-06-29T13:04:36.61Z" },
    { url = "https://files.pythonhosted.org/packages/7c/a2/d88de6d313d734a544a7901353ad5db67cb38dcfcd91713b7979dafc345d/jiter-0.16.0-cp314-cp314t-win_arm64.whl", hash = "sha256:0fa25b09b13075c46f5bc174f2690525a925a4fc2f7c82969a2bbabff22386ce", size = 190516, upload-time = "2026-06-29T13:04:38.004Z" },
    { url = "https://files.pythonhosted.org/packages/98/ab/664fd8c4be028b2bedd3d2ff08769c4ede23d0dbc87a77c62384a0515b5d/jiter-0.16.0-graalpy312-graalpy250_312_native-macosx_10_12_x86_64.whl", hash = "sha256:f17d61a28b4b3e0e3e2ba98490c70501403b4d196f78732439160e7fd3678127", size = 303106, upload-time = "2026-06-29T13:05:07.118Z" },
    { url = "https://files.pythonhosted.org/packages/1a/07/421f1d5b65493a76e16027b848aba6a7d28073ae75944fa4289cc914d39f/jiter-0.16.0-graalpy312-graalpy250_312_native-macosx_11_0_arm64.whl", hash = "sha256:96e38eea538c8ddf853a35727c7be0741c76c13f04148ac5c116222f50ece3b3", size = 304658, upload-time = "2026-06-29T13:05:08.708Z" },
    { url = "https://files.pythonhosted.org/packages/0a/db/bba1155f01a01c3c37a89425d571da751bbedf5c54247b831a04cb971798/jiter-0.16.0-graalpy312-graalpy250_312_native-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:d284fb8d94d5855d60c44fefcab4bf966f1da6fada73992b01f6f0c9bc0c6702", size = 339719, upload-time = "2026-06-29T13:05:10.41Z" },
    { url = "https://files.pythonhosted.org/packages/78/f7/18a1afcd64f35314b68c1f23afcd9994d0bc13e65cc77517afff4e83986d/jiter-0.16.0-graalpy312-graalpy250_312_native-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:64d613743df53199b1aa256a7d328340da6d7078aac7705a7db9d7a791e9cfd2", size = 343885, upload-time = "2026-06-29T13:05:12.087Z" },
]

[[package]]
name = "joblib"
version = "1.5.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/41/f2/d34e8b3a08a9cc79a50b2208a93dce981fe615b64d5a4d4abee421d898df/joblib-1.5.3.tar.gz", hash = "sha256:8561a3269e6801106863fd0d6d84bb737be9e7631e33aaed3fb9ce5953688da3", size = 331603, upload-time = "2025-12-15T08:41:46.427Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/7b/91/984aca2ec129e2757d1e4e3c81c3fcda9d0f85b74670a094cc443d9ee949/joblib-1.5.3-py3-none-any.whl", hash = "sha256:5fc3c5039fc5ca8c0276333a188bbd59d6b7ab37fe6632daa76bc7f9ec18e713", size = 309071, upload-time = "2025-12-15T08:41:44.973Z" },
]

[[package]]
name = "jsonpatch"
version = "1.33"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "jsonpointer" },
]
sdist = { url = "https://files.pythonhosted.org/packages/42/78/18813351fe5d63acad16aec57f94ec2b70a09e53ca98145589e185423873/jsonpatch-1.33.tar.gz", hash = "sha256:9fcd4009c41e6d12348b4a0ff2563ba56a2923a7dfee731d004e212e1ee5030c", size = 21699, upload-time = "2023-06-26T12:07:29.144Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/73/07/02e16ed01e04a374e644b575638ec7987ae846d25ad97bcc9945a3ee4b0e/jsonpatch-1.33-py2.py3-none-any.whl", hash = "sha256:0ae28c0cd062bbd8b8ecc26d7d164fbbea9652a1a3693f3b956c1eae5145dade", size = 12898, upload-time = "2023-06-16T21:01:28.466Z" },
]

[[package]]
name = "jsonpointer"
version = "3.1.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/18/c7/af399a2e7a67fd18d63c40c5e62d3af4e67b836a2107468b6a5ea24c4304/jsonpointer-3.1.1.tar.gz", hash = "sha256:0b801c7db33a904024f6004d526dcc53bbb8a4a0f4e32bfd10beadf60adf1900", size = 9068, upload-time = "2026-03-23T22:32:32.458Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/9e/6a/a83720e953b1682d2d109d3c2dbb0bc9bf28cc1cbc205be4ef4be5da709d/jsonpointer-3.1.1-py3-none-any.whl", hash = "sha256:8ff8b95779d071ba472cf5bc913028df06031797532f08a7d5b602d8b2a488ca", size = 7659, upload-time = "2026-03-23T22:32:31.568Z" },
]

[[package]]
name = "jsonschema"
version = "4.26.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "attrs" },
    { name = "jsonschema-specifications" },
    { name = "referencing" },
    { name = "rpds-py" },
]
sdist = { url = "https://files.pythonhosted.org/packages/b3/fc/e067678238fa451312d4c62bf6e6cf5ec56375422aee02f9cb5f909b3047/jsonschema-4.26.0.tar.gz", hash = "sha256:0c26707e2efad8aa1bfc5b7ce170f3fccc2e4918ff85989ba9ffa9facb2be326", size = 366583, upload-time = "2026-01-07T13:41:07.246Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/69/90/f63fb5873511e014207a475e2bb4e8b2e570d655b00ac19a9a0ca0a385ee/jsonschema-4.26.0-py3-none-any.whl", hash = "sha256:d489f15263b8d200f8387e64b4c3a75f06629559fb73deb8fdfb525f2dab50ce", size = 90630, upload-time = "2026-01-07T13:41:05.306Z" },
]

[[package]]
```

```
name = "jsonschema-specifications"
version = "2025.9.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "referencing" },
]
sdist = { url = "https://files.pythonhosted.org/packages/19/74/a633ee74eb36c44aa6d1095e7cc5569bebf04342ee146178e2d36600708b/jsonschema_specifications-2025.9.1.tar.gz", hash = "sha256:b540987f239e745613c7a9176f3edb72b832a4ac465cf02712288397832b5e8d", size = 32855, upload-time = "2025-09-08T01:34:59.186Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/41/45/1a4ed80516f02155c51f51e8cedb3c1902296743db0bbc66608a0db2814f/jsonschema_specifications-2025.9.1-py3-none-any.whl", hash = "sha256:98802fee3a11ee76ecaca44429fda8a41bff98b00a0f2838151b113f210cc6fe", size = 18437, upload-time = "2025-09-08T01:34:57.871Z" },
]

[[package]]
name = "langchain"
version = "1.3.14"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "langchain-core" },
    { name = "langgraph" },
    { name = "pydantic" },
]
sdist = { url = "https://files.pythonhosted.org/packages/29/68/a6dbad9c22df4087a0f9e79ddd46226c442b30128bfeee538d5889492a73/langchain-1.3.14.tar.gz", hash = "sha256:1b6696c72ba3bbbce54d745e0180742c9f6ece8bbc59ed5a46c3e20b9a435929", size = 645181, upload-time = "2026-07-16T13:28:18.29Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a9/ec/0f942e78a621f8e3162ff1ed24284f469aaf51fb4607ee5831c626f2b2bc/langchain-1.3.14-py3-none-any.whl", hash = "sha256:4d10dbe91005952cddd56d0dc77aa108964da6bae90ab20063653957e901f782", size = 139560, upload-time = "2026-07-16T13:28:16.498Z" },
]

[[package]]
name = "langchain-anthropic"
version = "1.5.4"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anthropic" },
    { name = "langchain-core" },
    { name = "pydantic" },
]
sdist = { url = "https://files.pythonhosted.org/packages/d8/9d/d56f68ec9a2c8aba3b7639e5d7a502d82cbf71c31e178a06c0955d789792/langchain_anthropic-1.5.4.tar.gz", hash = "sha256:113cb9bdac3169f2da65ea395dedb290e30dc6f899d01c205ba88aa88a2a02c2", size = 718860, upload-time = "2026-08-05T19:03:16.848Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/23/d8/47fb549e91f55a54ce829b5034a21360d9383342a3bf4986188e254e89e4/langchain_anthropic-1.5.4-py3-none-any.whl", hash = "sha256:730a9cb1ad384c9f1642840469c1ebbf20237066b1635f5d4fa9876e365fceaf", size = 56025, upload-time = "2026-08-05T19:03:15.579Z" },
]

[[package]]
name = "langchain-core"
version = "1.5.3"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "jsonpatch" },
    { name = "langchain-protocol" },
    { name = "langsmith" },
    { name = "packaging" },
    { name = "pydantic" },
    { name = "pyyaml" },
    { name = "tenacity" },
    { name = "typing-extensions" },
    { name = "uuid-utils" },
]
sdist = { url = "https://files.pythonhosted.org/packages/65/3e/63af6b9d76d9be907c7c524d6ec18a2efed7e0e2d123fea0230d78dbd73f/langchain_core-1.5.3.tar.gz", hash = "sha256:a56457ac444fef41e9404443c187f0ecea708d36e816ea4ba9573c027f7d1a2d", size = 972461, upload-time = "2026-07-30T14:55:55.833Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/36/e6/c7c39efe0bc7e1b7c3d8f54f85846e04c901913c3d3e99068b218558c6f1/langchain_core-1.5.3-py3-none-any.whl", hash = "sha256:48b56fa580277209594dd7baf837f5b9a2a3651613f34ff9fb1728b429df015f", size = 561687, upload-time = "2026-07-30T14:55:54.419Z" },
]

[[package]]
name = "langchain-protocol"
version = "0.0.18"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/d2/59/b5959aea96faa9146e2e49a7a22882b3528c62efafe9a6a95beab30c2305/langchain_protocol-0.0.18.tar.gz", hash = "sha256:ec3e11782f1ed0c9db38e5a9ed01b0e7a0d3fba406faa8aef6594b73c56a63e6", size = 6150, upload-time = "2026-06-18T17:08:26.959Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/99/2e/d82db9eec13ad0f72e7aaad5c4bc730ab111934fdc83c85523206eb9b0a0/langchain_protocol-0.0.18-py3-none-any.whl", hash = "sha256:70b53a86fbf9cedc863555effe44da192ab02d556ddbf2cf95b8873adcf41b5a", size = 7221, upload-time = "2026-06-18T17:08:25.996Z"
 },
]

[[package]]
name = "langfuse"
version = "4.14.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "backoff" },
    { name = "httpx" },
    { name = "opentelemetry-api" },
    { name = "opentelemetry-exporter-otlp-proto-http" },
    { name = "opentelemetry-sdk" },
    { name = "packaging" },
    { name = "pydantic" },
    { name = "wrapt" },
]
sdist = { url = "https://files.pythonhosted.org/packages/36/7c/4cdf76a100f169a0855d4a7e69598a3bbe19bd2f048c5de3c298af1aa2c0/langfuse-4.14.2.tar.gz", hash = "sha256:76eb3f2c2106ddd63271154d1a33f8b3d4036f449a9f9b48403c36bd5e7c1bba", size = 389507, upload-time = "2026-07-30T11:54:48.46Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/8b/b5/1499bcedb6872e8acc4e7043c522c9f3a4fe2eafd78a6f37e73308d18753/langfuse-4.14.2-py3-none-any.whl", hash = "sha256:27879cbaea570e94323c6ced60a0997feedf5294f18b51436009947c42f9f92e", size = 688132, upload-time = "2026-07-30T11:54:49.553Z" },
]
```

```
[[package]]
name = "langgraph"
version = "1.2.10"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "langchain-core" },
    { name = "langgraph-checkpoint" },
    { name = "langgraph-prebuilt" },
    { name = "langgraph-sdk" },
    { name = "pydantic" },
    { name = "xxhash" },
]
sdist = { url = "https://files.pythonhosted.org/packages/70/1d/a32f3caf4b3d60651656c0d64976b48d168653e81c71bb7512e9a31541aa/langgraph-1.2.10.tar.gz",
hash = "sha256:05a183a746ed570a06c7c1b879920163509a75df9e44e92dd2238218d677fd37", size = 723404, upload-time = "2026-07-28T18:33:51.441Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e2/4d/3fc3e2535ee2c731130d71371848ebc6d4a9d2e8ae6060b11987ba134951/langgraph-1.2.10-py3-none-any.
whl", hash = "sha256:52c48bd42fa31a1de0e1c0f0ebfe342e11ca2957b8b3563f83dbd60d8e30f921", size = 247753, upload-time = "2026-07-28T18:33:50.028Z" },
]

[[package]]
name = "langgraph-checkpoint"
version = "4.1.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "langchain-core" },
    { name = "ormsgpack" },
]
sdist = { url = "https://files.pythonhosted.org/packages/83/47/886af6f886f0bff2273164a45f008694e48a96ff3cd25ff0228f2aa9480e/langgraph_checkpoint-4.1.1
.tar.gz", hash = "sha256:6c2bdb530c91f91d7d9c1bd100925d0fc4f498d418c17f3587d1526279482a25", size = 184020, upload-time = "2026-05-22T16:57:38.503Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/bd/b4/71425e3e38be92611300b9cc5e46a5bf98ab23f5ea8a75b73d02a2f1413c/langgraph_checkpoint-4.1.1-py3
-none-any.whl", hash = "sha256:25d29144b082827218e7bc3f1e9b0566a4bb007895cd6cc26f66a8428739f56e", size = 56212, upload-time = "2026-05-22T16:57:37.203
Z" },
]

[[package]]
name = "langgraph-prebuilt"
version = "1.1.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "langchain-core" },
    { name = "langgraph-checkpoint" },
]
sdist = { url = "https://files.pythonhosted.org/packages/29/66/ed9b93f56bc17ef22d551892f0ac2b225a97fe0fcf23a511b857f70d590b/langgraph_prebuilt-1.1.0.t
ar.gz", hash = "sha256:3c579cf6eed2d17f9c157c2d0fcaddcd8688524e7022d3b22b37a3bf4589d528", size = 178833, upload-time = "2026-05-12T03:37:49.332Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e9/43/3fe1a700b8490ed02679cdbbc8c915eb23a092faf496c9c1118abcd10be3/langgraph_prebuilt-1.1.0-py3-n
one-any.whl", hash = "sha256:51e311747d755b751d5c6b39b0c1446124d3a7643d2515017e6714b323508fc9", size = 41043, upload-time = "2026-05-12T03:37:48.007Z"
 },
]

[[package]]
name = "langgraph-sdk"
version = "0.4.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "httpx" },
    { name = "langchain-core" },
    { name = "langchain-protocol" },
    { name = "orjson" },
    { name = "websockets" },
]
sdist = { url = "https://files.pythonhosted.org/packages/b4/2b/bd8ac26d4e97f6df88ef05ce5b6a38945a3903e1025d926f4752aa88aa97/langgraph_sdk-0.4.2.tar.gz
", hash = "sha256:b88f0f5f6328ac0680d6790614a905b2bcfa257f2276dba4e38f0e86db0aa738", size = 348327, upload-time = "2026-06-01T17:51:19.856Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a0/05/aac507337cceae773c2cc9ab91eb6301963af7aeeb55b4217a00e15aff17/langgraph_sdk-0.4.2-py3-none-a
ny.whl", hash = "sha256:75fa5096c1177ce39c847096a8fe3745ffd480ddb412995f836e9f5f884c43dd", size = 160521, upload-time = "2026-06-01T17:51:18.849Z" },
]

[[package]]
name = "langsmith"
version = "0.10.16"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "distro" },
    { name = "httpx" },
    { name = "orjson", marker = "platform_python_implementation != 'PyPy'" },
    { name = "packaging" },
    { name = "pydantic" },
    { name = "requests" },
    { name = "requests-toolbelt" },
    { name = "sniffio" },
    { name = "typing-extensions" },
    { name = "uuid-utils" },
    { name = "websockets" },
    { name = "xxhash" },
    { name = "zstandard" },
]
sdist = { url = "https://files.pythonhosted.org/packages/b0/f1/ee288a685ae666121a5196575e56886e853b8fde9bbc5b796279cf85b2ba/langsmith-0.10.16.tar.gz",
 hash = "sha256:7dc5ff6477d50101b4944a985773bf9a3e2a9374e5d71ed2549ada929ca500b2", size = 4797058, upload-time = "2026-08-05T16:59:31.13Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/95/0b/cf0a51c19f659b40351dab76b4d3f0a3554807966a912111fda3156c96fe/langsmith-0.10.16-py3-none-any
.whl", hash = "sha256:f0b3882e0d2d69bda596c7b452a7857943aefb2c531c047677470b0a394cc490", size = 734191, upload-time = "2026-08-05T16:59:28.021Z" },
]

[[package]]
name = "librt"
version = "0.13.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/dc/2f/3908645ddddab7120b46295e541ead308109fa48dbec7d67d7a778870d60/librt-0.13.0.tar.gz", hash
 = "sha256:1d2a610c14ac0d0750ee0a3ab8548e83155258387891caaca04def4bf7289781", size = 211402, upload-time = "2026-07-08T12:26:29.834Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/f0/f4/b2933ddae222dac338476abb872641169a5cfed2c2bb5444a5b07b32b0c3/librt-0.13.0-cp312-cp312-macos
```

x_10_13_x86_64.whl", hash = "sha256:30536798f4504c0fad0885b1d371b0539abb081e4570c9d7c641cb51141b49f0", size = 150990, upload-time = "2026-07-08T12:25:02.422Z" },
    { url = "https://files.pythonhosted.org/packages/90/ef/db98f744ca50e6efc9c95c70ee49b77aefac31f6a3fc7c83754a42d6a74f/librt-0.13.0-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:93d24ebb82aa4420b1409c389e7857bc35bd0b668007ac8172427d5c73cc8cc5", size = 155238, upload-time = "2026-07-08T12:25:03.681Z" },
    { url = "https://files.pythonhosted.org/packages/03/e7/a197e7bc72baf2c61ce7fdc6906a5054dc05bd8da0819aa894e4857bf87e/librt-0.13.0-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:cb8a1adce42d8b75485a5d56a9623a50bcab995b6079f1dac59fc44034dd93d9", size = 503073, upload-time = "2026-07-08T12:25:05.049Z" },
    { url = "https://files.pythonhosted.org/packages/f8/e7/7887712e27da7c1ab80fcabb1de6eb24243964f6557cae530d4b70706dbd/librt-0.13.0-cp312-cp312-manylinux2014_i686.manylinux_2_17_i686.manylinux_2_28_i686.whl", hash = "sha256:0763ca2ab66058174f9dee426dc64f5e0a89c24a7df8d3fe3f1836c04e25de4b", size = 496528, upload-time = "2026-07-08T12:25:06.26Z" },
    { url = "https://files.pythonhosted.org/packages/94/f0/f2283385bb6b950b26a1410f4ce51ec27231e0b3a4b925c46366d218b198/librt-0.13.0-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:b222493da6e7b6199db9bd79502436cf5a27da3c1f7fa83c7e285444fc93fd03", size = 531786, upload-time = "2026-07-08T12:25:07.658Z" },
    { url = "https://files.pythonhosted.org/packages/36/11/69ac3b54766ffba5fd7e5acebfb048d66dbe1f9f2d14516c2b3edc59cf87/librt-0.13.0-cp312-cp312-manylinux_2_34_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:fadc63331f4388c3dc90090448f682a7e9feafc11481391c1e94f2f907a3976e", size = 524393, upload-time = "2026-07-08T12:25:09.121Z" },
    { url = "https://files.pythonhosted.org/packages/61/5f/d72f95fd444a926a3c14b4e24979474116988dd57a45be242077c45d3c22/librt-0.13.0-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:70d9c62a4cffd9f23396cd5ef93fc5d11b31596b9b7d6306074abe3d5fcf09bd", size = 543026, upload-time = "2026-07-08T12:25:10.459Z" },
    { url = "https://files.pythonhosted.org/packages/c4/08/dcd9993ad192737a004ba263d549f8ea605b326b952e7d6205c7d4170b76/librt-0.13.0-cp312-cp312-musllinux_1_2_i686.whl", hash = "sha256:66c0e7e6b02a155576df2c77ec933a70b72da726e248c494abf690923e624348", size = 546829, upload-time = "2026-07-08T12:25:11.716Z" },
    { url = "https://files.pythonhosted.org/packages/96/d5/6d9bb2f54e4109a956b7128836529653eb9d740f784bc47ed10a02c1000e/librt-0.13.0-cp312-cp312-musllinux_1_2_riscv64.whl", hash = "sha256:ac04bcd3328eb91d99dfedf6a60d9c1f15d3434e6f6daf922f0420f7d90b85c7", size = 535700, upload-time = "2026-07-08T12:25:13.144Z" },
    { url = "https://files.pythonhosted.org/packages/8c/f2/10946922503858a359492fa27f13e86228bde702116a740ac7b3cd185f24/librt-0.13.0-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:db327e7271e653c32040b85ae6188059c924b57d7e1e29f935523fa017cd4e82", size = 573566, upload-time = "2026-07-08T12:25:14.336Z" },
    { url = "https://files.pythonhosted.org/packages/48/a8/94f00e3c99479a18088af3685ea016c42f3c7d5d1964d8dbb40c08d7f1aa/librt-0.13.0-cp312-cp312-win32.whl", hash = "sha256:860bd1d8ba48456ce08feaf8d343a8aaeb2fa086f2bcaa2a923fa3f7a3ff9aa3", size = 106099, upload-time = "2026-07-08T12:25:16.159Z" },
    { url = "https://files.pythonhosted.org/packages/c9/7b/2da9c74c1ed25a89cc4e1c8e007ea2eb4a0f1fafa3e70d757fe3242c5c5c/librt-0.13.0-cp312-cp312-win_amd64.whl", hash = "sha256:e54a315caf843c8d77e388cadc56ea9ded569935ee2d2347d7ea94992e5aa6fa", size = 126934, upload-time = "2026-07-08T12:25:17.275Z" },
    { url = "https://files.pythonhosted.org/packages/d0/65/aead61bbf3b5358593f9d4779d2a0e88eaf6ec191a6342dde36dd1df6371/librt-0.13.0-cp312-cp312-win_arm64.whl", hash = "sha256:c718e99a0992127af84385378460db624103b559ab260435abcfe77a4e4ed1c1", size = 112236, upload-time = "2026-07-08T12:25:18.425Z" },
    { url = "https://files.pythonhosted.org/packages/67/3b/18e7b63255297a2bdc9c25c8d6d4ca8eca9f63aceb1252c0f7427ac7099e/librt-0.13.0-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:a468951af16155824e88bdd8326ebe5bdb371f3ec0ac04642994b98201d914f3", size = 151027, upload-time = "2026-07-08T12:25:19.638Z" },
    { url = "https://files.pythonhosted.org/packages/4d/68/e2248452c00d1a03b45fee1752cdc8f790a476efd2402b75181da88a9e61/librt-0.13.0-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:ae01d8512cc17079e53425635327dbf3f7ff57a42c00dec348bf79791c56444c", size = 155152, upload-time = "2026-07-08T12:25:20.851Z" },
    { url = "https://files.pythonhosted.org/packages/0e/16/52b1c99bf19057a062aac39c900cbb81499f6f75d6c537c14463d247ba78/librt-0.13.0-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:32c26893cd085c1efe83219e78d866da23fb20a066101b8f68210004361d224c", size = 502499, upload-time = "2026-07-08T12:25:22.055Z" },
    { url = "https://files.pythonhosted.org/packages/9f/54/b811151805c795f55e0dedee6ec687b75f9982a8105d240ea3910737a77b/librt-0.13.0-cp313-cp313-manylinux2014_i686.manylinux_2_17_i686.manylinux_2_28_i686.whl", hash = "sha256:5929da1981a46bcf4b28b1b9499905f0ff58e2419da402a048234e9783acbc4b", size = 496108, upload-time = "2026-07-08T12:25:23.296Z" },
    { url = "https://files.pythonhosted.org/packages/8f/f8/094d6b2bd93f3fdaa54db54cc788c4a365333bddad65ab02e04da0b1d004/librt-0.13.0-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:94b85d664d777bab6c0d709416cb42938251fda9e221b79e3a2215d85df5f4f9", size = 531576, upload-time = "2026-07-08T12:25:24.648Z" },
    { url = "https://files.pythonhosted.org/packages/2e/40/541733d5755824f968f7ec39d78ffbd75d145964157ae5e69a09ec6d7326/librt-0.13.0-cp313-cp313-manylinux_2_34_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:531b2df3e9fe96b1fcf73a6d165921e4656be5f58d631d384ebce344298368db", size = 524390, upload-time = "2026-07-08T12:25:25.898Z" },
    { url = "https://files.pythonhosted.org/packages/c6/b5/255673cfdbf5ba663339d36cd863c897289ab4337577e19f9405ce059f36/librt-0.13.0-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:109b84a9edf69ad89dc1f66358659e14a031baca95e3e5b0060bd903ede8efd6", size = 543053, upload-time = "2026-07-08T12:25:27.436Z" },
    { url = "https://files.pythonhosted.org/packages/9e/11/ab5005e9c9850710f21e354201bf090646349d3fabf5f951eaf70235729e/librt-0.13.0-cp313-cp313-musllinux_1_2_i686.whl", hash = "sha256:1304368a3e7ffc3e9db986796cc5326fdb5943a3567ecc137cff318e4240c0e7", size = 546387, upload-time = "2026-07-08T12:25:28.65Z" },
    { url = "https://files.pythonhosted.org/packages/a2/04/a5d7ce1d1df1afd15ca283dcdf7530ac073e12d69ae8c40879dda96f7868/librt-0.13.0-cp313-cp313-musllinux_1_2_riscv64.whl", hash = "sha256:e4f9b472e7d308d94b62c801982065661158c6ed02790d6c7ddb4337cea0f9c1", size = 535970, upload-time = "2026-07-08T12:25:30.171Z" },
    { url = "https://files.pythonhosted.org/packages/5a/76/927e267a6daa290174ac281b23c9804c8829b042ade9c6f24a065f540958/librt-0.13.0-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:9f836c37478f167a81200d8c8b2c920a22224564bed2c23d7aeec760965c367a", size = 573582, upload-time = "2026-07-08T12:25:31.507Z" },
    { url = "https://files.pythonhosted.org/packages/10/24/b6c5213efe39c19f9e13605644d0cf063b4ddaa33ac2e45b088e23a70e2e/librt-0.13.0-cp313-cp313-pyemscripten_2025_0_wasm32.whl", hash = "sha256:4000d961ff9598ac6ea603c6c836a5ed49bc205ade5fc378b998dfe1e2c36628", size = 82189, upload-time = "2026-07-08T12:25:32.675Z" },
    { url = "https://files.pythonhosted.org/packages/4c/00/d29736be177a906ac0b84a5b04b4fbfa22c776dc2f366de4172b0f968c08/librt-0.13.0-cp313-cp313-win32.whl", hash = "sha256:79e44cff71750d299d61a678e49995b0d5935a9cda238c2574daeca3ba536927", size = 106193, upload-time = "2026-07-08T12:25:33.692Z" },
    { url = "https://files.pythonhosted.org/packages/c8/ac/aff6fb45393cb8912f39dfb156ef6b2d1cadb207ff465fc8f66141054be8/librt-0.13.0-cp313-cp313-win_amd64.whl", hash = "sha256:54dab44a847d5ad1acd05c8a83fe518ae685516ecf4d3f7cc6e3df2a66767650", size = 126962, upload-time = "2026-07-08T12:25:34.769Z" },
    { url = "https://files.pythonhosted.org/packages/d9/3a/d68cb2b334d53fd30fac81d3a489ce4ba0d9506f4df43fcf676b68352b19/librt-0.13.0-cp313-cp313-win_arm64.whl", hash = "sha256:d4cb6fbfdf874340ab5e51450753c0f817b6958a3621125ee695bbc3de866566", size = 112127, upload-time = "2026-07-08T12:25:35.981Z" },
    { url = "https://files.pythonhosted.org/packages/7b/66/f49ae0d592bd45b6941e9a8bafcb6a87cddcd501ee7874707e767f01b585/librt-0.13.0-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:25218d94b1d2cbc0ba1d8a3f9dc9af578d9646e5ed16443a70cde1dfdcce6d71", size = 149818, upload-time = "2026-07-08T12:25:37.203Z" },
    { url = "https://files.pythonhosted.org/packages/3d/50/51c76d74014d04fb95b6506d286808984b78a2f7a41039094e6b2194ac48/librt-0.13.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:f26629539d4893c2957a16c41bb058e1e135c1f150f6a2e25ed047f64cf3f5c6", size = 154071, upload-time = "2026-07-08T12:25:39.399Z" },
    { url = "https://files.pythonhosted.org/packages/b8/fe/f19b0f5f82d5a1f2da736586bc840abd00ce07d6388136ae80b7333883fc/librt-0.13.0-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:a4517d47b2b8af26975a406fba7d314de9696d864252e0257c6ea90238cfe27f", size = 494168, upload-time = "2026-07-08T12:25:40.641Z" },
    { url = "https://files.pythonhosted.org/packages/94/bc/b8550c75775127fd31a5f20e8775997f7b527ad661fc8ddccd7497c064f7/librt-0.13.0-cp314-cp314-manylinux2014_i686.manylinux_2_17_i686.manylinux_2_28_i686.whl", hash = "sha256:f19e181de5b3a1148bb3420b8c4b0b0ea0fce6950099724ad151d6cea5acc180", size = 491054, upload-time = "2026-07-08T12:25:41.905Z" },
    { url = "https://files.pythonhosted.org/packages/30/14/4d0204867623df3f33f86efd3d3692ba5e01321443f4d6eab35a22697618/librt-0.13.0-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:22034924f5b42d5a56371cf271771bfeaabf235a7a8b6264bef2d20013f786c6", size = 523006, upload-time = "2026-07-08T12:25:43.327Z" },
    { url = "https://files.pythonhosted.org/packages/19/0a/c45fc9a260934696bace1ac5df1e148ac92bd71767aee3bf7cd7a4534f4c/librt-0.13.0-cp314-cp314-manylinux_2_34_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:c7897db4e95e22468bdda33d8e012ceacd0182abf001e6389d763f0def6286b9", size = 515058, upload-time = "2026-07-08T12:25:44.541Z" },
    { url = "https://files.pythonhosted.org/packages/13/0a/50c5ce45b326854ef8fa6ae4c36cf5142e5c55315eaf9e51d0ae73ac4da3/librt-0.13.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:1ce61b3746545029d4f5c17d6bd74b676254ad98433086c846ffb5e8fa73f007", size = 534025, upload-time = "2026-07-08T12:25:45.825Z" },
    { url = "https://files.pythonhosted.org/packages/89/2d/08c413c8f93fc13b8103624fce38e5caa86cd08cbbc8465870ab287af54b/librt-0.13.0-cp314-cp314-musllinux_1_2_i686.whl", hash = "sha256:46c330e82565962c761dbce7941be2cff7db674ee807455a8d0cadc5f9b759b0", size = 540557, upload-time = "2026-07-08T12:25:47.059Z" },
    { url = "https://files.pythonhosted.org/packages/b3/c1/93af71fb4a364952210051811dd4e40174e79656b050c89cacac18af3330/librt-0.13.0-cp314-cp314-musllinux_1_2_riscv64.whl", hash = "sha256:375f5af8f99cbaa99dd293af986e3d57caabc9ba81a5d3f021603764854197a1", size = 523201, upload-time = "2026-07-08T12:25:48.392Z" },

```
    { url = "https://files.pythonhosted.org/packages/c1/6e/9766f07b676a4889d9f8bc2864e9ba5fff165653143ef4dda7df6aa34d16/librt-0.13.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:9320d34c3376ae204b2cd176e8d4883a013934e0aef822f1aed9c536490c275d", size = 565740, upload-time = "2026-07-08T12:25:49.678Z" },
    { url = "https://files.pythonhosted.org/packages/a2/1e/664e3472ce2b6e10e9b83f29d4a36eb982ff6b5a169ae7567bba3a4c4ff5/librt-0.13.0-cp314-cp314-pyemscripten_2026_0_wasm32.whl", hash = "sha256:9af313c66157a69dc69ea0059a66961692250e0dc95af9c385a48ffb770a0d16", size = 81611, upload-time = "2026-07-08T12:25:50.857Z" },
    { url = "https://files.pythonhosted.org/packages/2f/d4/8582a4d65e2234673685e07309d02c230b28a85724eb0acbf13f019b7f6e/librt-0.13.0-cp314-cp314-win32.whl", hash = "sha256:f2a7253458e34f33543551394ae4fe104b497ec2a65ac266074de64c1df82e37", size = 100106, upload-time = "2026-07-08T12:25:52.03Z" },
    { url = "https://files.pythonhosted.org/packages/63/ce/0cb99efe6086b46cd985dc26672166fae312a239690e75871f7fafbd3fc5/librt-0.13.0-cp314-cp314-win_amd64.whl", hash = "sha256:a3dfe4edf10e8ed7e55b026a8bfc2c2a8704218b659cd4bffdf604fab966dc39", size = 121209, upload-time = "2026-07-08T12:25:53.166Z" },
    { url = "https://files.pythonhosted.org/packages/26/85/4f3ccb083a3c9b0d42e223acdb3c3f507953324a59cdcab4826e8e2e3b89/librt-0.13.0-cp314-cp314-win_arm64.whl", hash = "sha256:68a5faee4bba381cb93b5961f684a514cf0053cb92308ff9c792c2fea0b174c6", size = 106404, upload-time = "2026-07-08T12:25:54.253Z" },
    { url = "https://files.pythonhosted.org/packages/b2/77/333191499538c8e8189de7a4cba8e6f49ee949fd6d6e6324b21fd1522466/librt-0.13.0-cp314-cp314t-macosx_10_15_x86_64.whl", hash = "sha256:a38fb81d8376dfa2f8963b265fec07637802b0d01e2a127c19c66cb070fb24f5", size = 159231, upload-time = "2026-07-08T12:25:55.432Z" },
    { url = "https://files.pythonhosted.org/packages/7a/9e/2aa83758f22c278b837a1d8025898434ce2b8bff36678d5330ecaef56dff/librt-0.13.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:d4c8d9bd5abce34b2e75edb3bf37ab0f34e49b1f915a40ae8468eb7c85bc5b46", size = 161300, upload-time = "2026-07-08T12:25:56.585Z" },
    { url = "https://files.pythonhosted.org/packages/bb/c0/86791e936553ca763d6b3c2fb4d31d596cd00e14fa631c283a40ba01559a/librt-0.13.0-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:387e2f1d27e89bffe0d3f520f0da0662c973fd607ca16c1808f8a5085419485e", size = 582056, upload-time = "2026-07-08T12:25:58.144Z" },
    { url = "https://files.pythonhosted.org/packages/a8/d3/a9ec15984a185e000c4d2a16ba28bd623124ad4c38a10974c7ff78e3a893/librt-0.13.0-cp314-cp314t-manylinux2014_i686.manylinux_2_17_i686.manylinux_2_28_i686.whl", hash = "sha256:4f6db193d2e5e0ed60359b9a5a682cd67205d0d3b1e459a867dd4b5c4e7eaa7a", size = 562758, upload-time = "2026-07-08T12:25:59.544Z" },
    { url = "https://files.pythonhosted.org/packages/3c/af/dbe36b78b19c06a55097f99305e4ea9458e2273e6ae16a3cbecaad7ee978/librt-0.13.0-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:0d38604854e8d22faadf683ec6c02bb0f886e2ba56ef981a1c36ee275f21ea22", size = 602095, upload-time = "2026-07-08T12:26:00.991Z" },
    { url = "https://files.pythonhosted.org/packages/2a/a8/2966891b4dd2830f5203fbee92ac2c4947653a2390ba73dfa44244fad025/librt-0.13.0-cp314-cp314t-manylinux_2_34_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:371f7ce73026815dafd51c50ce38416e91428b28c4b2ec97cd39271164b0045c", size = 593452, uploaded-time = "2026-07-08T12:26:02.352Z" },
    { url = "https://files.pythonhosted.org/packages/61/f5/4df8bfc8405ecf8c0d525b4d69636f694bdd8620b313ec8b76e54a5926cc/librt-0.13.0-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:3aaedf52171bee90860704c560bc798fe83b76247df47568e0197e9b13c735a0", size = 623729, upload-time = "2026-07-08T12:26:04.294Z" },
    { url = "https://files.pythonhosted.org/packages/d6/13/9ac202dffc8db06f75d06c08c2f9f6ff054be67d21272dcc078fa1cc0c57/librt-0.13.0-cp314-cp314t-musllinux_1_2_i686.whl", hash = "sha256:96bad8725a4f196a798366c25ce075d1f7543a4ec045ffc13e6a7ec095cdab04", size = 617077, upload-time = "2026-07-08T12:26:05.845Z" },
    { url = "https://files.pythonhosted.org/packages/6e/f0/ebe38610716aee5cb28efd95089bb9019209617980277938 1e1c5dcf239c/librt-0.13.0-cp314-cp314t-musllinux_1_2_riscv64.whl", hash = "sha256:6bf6a559ffe4a93bbea6cf31ddf01a7fd9ba342ef51f27beb178e318b74acd61", size = 599561, upload-time = "2026-07-08T12:26:07.21Z" },
    { url = "https://files.pythonhosted.org/packages/4f/5c/c2e72e236fff7abc716d5b1753b8b8cd3ea85ac46fe17d2e7c51d4e1c723/librt-0.13.0-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:301067672387902c55f94b51d5022304b36c966ea9fe1f21caab99a9bef487c9", size = 645511, upload-time = "2026-07-08T12:26:08.562Z" },
    { url = "https://files.pythonhosted.org/packages/0c/99/6203ce619dee940d6bfbe099ec3fe4be00a68e9d60f70abf906cf124fe66/librt-0.13.0-cp314-cp314t-win32.whl", hash = "sha256:5fdcf34f86de8fb66d7dc7589f96ba91c4aa46671200d400e6fd6f109a483f18", size = 104357, upload-time = "2026-07-08T12:26:09.828Z" },
    { url = "https://files.pythonhosted.org/packages/52/dd/843b6314087c41657c7036d7914d8f294bdf9b580aa8513ea0588c8e9a3d/librt-0.13.0-cp314-cp314t-win_amd64.whl", hash = "sha256:260c33e92263fa629b4f6d3c51967a1c2158fe6c33237aaa3ebeac586b085259", size = 126998, upload-time = "2026-07-08T12:26:10.975Z" },
    { url = "https://files.pythonhosted.org/packages/5f/5d/3dcec2884ba1b0806d1408612555c38dd5d68e90156b59f75f6e36435c3a/librt-0.13.0-cp314-cp314t-win_arm64.whl", hash = "sha256:2f281549a4c52ac7bb97997f14353f8bd0e53a34ca0dad1c905cfd0b4a58ae99", size = 110771, upload-time = "2026-07-08T12:26:12.303Z" },
]

[[package]]
name = "llama-index"
version = "0.14.23"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "llama-index-core" },
    { name = "llama-index-embeddings-openai" },
    { name = "llama-index-llms-openai" },
    { name = "nltk" },
]
sdist = { url = "https://files.pythonhosted.org/packages/cb/5c/1f9334c3c0edd880367d0734939256f8536ca94165c6a5fdb455a7bcf180/llama_index-0.14.23.tar.gz", hash = "sha256:eac2049816a7410ff4568490cce4bdff99cda3ab99d59f52f6227dad22cda44b", size = 8566, upload-time = "2026-06-24T19:36:38.241Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e0/92/da2a737bf712fe31c1aaec00f51745797d4570ecdf7e4f5d2b4b93b3d337/llama_index-0.14.23-py3-none-any.whl", hash = "sha256:c205de2442a7186b8e05096f0771f96fa6bdc9603fcad2e07eab1bc96dcf086a", size = 7114, upload-time = "2026-06-24T19:36:37.21Z" },
]

[[package]]
name = "llama-index-core"
version = "0.14.23"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "aiohttp" },
    { name = "aiosqlite" },
    { name = "banks" },
    { name = "dataclasses-json" },
    { name = "deprecated" },
    { name = "dirtyjson" },
    { name = "filetype" },
    { name = "fsspec" },
    { name = "httpx" },
    { name = "llama-index-workflows" },
    { name = "nest-asyncio" },
    { name = "networkx" },
    { name = "nltk" },
    { name = "numpy" },
    { name = "pillow" },
    { name = "platformdirs" },
    { name = "pydantic" },
    { name = "pyyaml" },
    { name = "requests" },
    { name = "setuptools" },
    { name = "sqlalchemy", extra = ["asyncio"] },
    { name = "tenacity" },
    { name = "tiktoken" },
    { name = "tinytag" },
    { name = "tqdm" },
    { name = "typing-extensions" },
    { name = "typing-inspect" },
    { name = "wrapt" },
]
```

```
sdist = { url = "https://files.pythonhosted.org/packages/5c/ac/f885ae14317af43a026c909ea4d2083fcee2f0d014f90426b5b9aa1f9912/llama_index_core-0.14.23.tar.gz", hash = "sha256:c4baf2f2ab4f84e95090fe7941e0c87d6c514304f7bd2a749b8fa22164c1822b", size = 11588373, upload-time = "2026-06-24T19:35:55.43Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/ec/d5/05d61f34c01c6578fb758d0a3ddef58d36c6ffa9a9f84a5c9a16262ad94d/llama_index_core-0.14.23-py3-none-any.whl", hash = "sha256:6a54d267826732a8507f81df40785b107f7592af20f451a39a59005147caf84c", size = 11924908, upload-time = "2026-06-24T19:35:52.833Z" },
]

[[package]]
name = "llama-index-embeddings-openai"
version = "0.6.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "llama-index-core" },
    { name = "openai" },
]
sdist = { url = "https://files.pythonhosted.org/packages/06/52/eb56a4887501651fb17400f7f571c1878109ff698efbe0bbac9165a5603d/llama_index_embeddings_openai-0.6.0.tar.gz", hash = "sha256:eb3e6606be81cb89125073e23c97c0a6119dabb4827adbd14697c2029ad73f29", size = 7629, upload-time = "2026-03-12T20:21:27.234Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/4e/d1/4bb0b80f4057903110060f617ef519197194b3ff5dd6153d850c8f5676fa/llama_index_embeddings_openai-0.6.0-py3-none-any.whl", hash = "sha256:039bb1007ad4267e25ddb89a206dfdab862bfb87d58da4271a3919e4f9df4d61", size = 7666, upload-time = "2026-03-12T20:21:28.079Z" },
]

[[package]]
name = "llama-index-instrumentation"
version = "0.5.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "deprecated" },
    { name = "pydantic" },
]
sdist = { url = "https://files.pythonhosted.org/packages/4e/d0/671b23ccff255c9bce132a84ffd5a6f4541ceefdeab9c1786b08c9722f2e/llama_index_instrumentation-0.5.0.tar.gz", hash = "sha256:eeb724648b25d149de882a5ac9e21c5acb1ce780da214bda2b075341af29ad8e", size = 43831, upload-time = "2026-03-12T20:17:06.742Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/c3/45/6dcaccef44e541ffa138e4b45e33e0d40ab2a7d845338483954fcf77bc75/llama_index_instrumentation-0.5.0-py3-none-any.whl", hash = "sha256:aaab83cddd9dd434278891012d8995f47a3bc7ed1736a371db90965348c56a21", size = 16444, upload-time = "2026-03-12T20:17:05.957Z" },
]

[[package]]
name = "llama-index-llms-openai"
version = "0.7.10"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "llama-index-core" },
    { name = "openai" },
]
sdist = { url = "https://files.pythonhosted.org/packages/74/6e/1ef83a852296c55fe8c529b4476073eccaff150d57aceb3d6319a87dcc89/llama_index_llms_openai-0.7.10.tar.gz", hash = "sha256:aa8dabf08ea0b9740fc9dae9677f05049254be5e929794a886692eaf26973def", size = 27586, upload-time = "2026-07-21T23:16:18.087Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/8b/89/854cb7b2e5be3b3c16561e93f3edd52d8a1b6e031be1aca4519b9fdd3917/llama_index_llms_openai-0.7.10-py3-none-any.whl", hash = "sha256:b76a0c3d4aaa9aba57dc0366ca6b2346c6819bed22973e334676d7a34704e46a", size = 28673, upload-time = "2026-07-21T23:16:17.062Z" },
]

[[package]]
name = "llama-index-workflows"
version = "2.23.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "llama-index-instrumentation" },
    { name = "pydantic" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/cd/6c/14113a4da3ed61f5f07ac87ad1a643ab5d5f2902678ecbf0cc1f8bee4ce7/llama_index_workflows-2.23.0.tar.gz", hash = "sha256:96056dfead58bd33c869d219c8ef56c4abf976c04226d7f6f113ea38db19e56d", size = 136758, upload-time = "2026-08-05T15:21:22.203Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/4a/26/da04a96662ff70f24c9ad4a4ca1a3f41dc521a61bc9c80414d390a860a87/llama_index_workflows-2.23.0-py3-none-any.whl", hash = "sha256:0a038c962294cddb441286075ce3f0a1d4cd21cd899545ab2ad5ef31a0ce5d00", size = 164623, upload-time = "2026-08-05T15:21:20.926Z" },
]

[[package]]
name = "loguru"
version = "0.7.3"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "colorama", marker = "sys_platform == 'win32'" },
    { name = "win32-setctime", marker = "sys_platform == 'win32'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/3a/05/a1dae3dffd1116099471c643b8924f5aa6524411dc6c63fdae648c4f1aca/loguru-0.7.3.tar.gz", hash = "sha256:19480589e77d47b8d85b2c827ad95d49bf31b0dcde16593892eb51dd18706eb6", size = 63559, upload-time = "2024-12-06T11:20:56.608Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/0c/29/0348de65b8cc732daa3e33e67806420b2ae89bdce2b04af740289c5c6c8c/loguru-0.7.3-py3-none-any.whl", hash = "sha256:31a33c10c8e1e10422bfd431aeb5d351c7cf7fa671e3c4df004162264b28220c", size = 61595, upload-time = "2024-12-06T11:20:54.538Z" },
]

[[package]]
name = "mako"
version = "1.4.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "markupsafe" },
]
sdist = { url = "https://files.pythonhosted.org/packages/2a/12/b5fa2353e2754cd67fb9f83793fa48ff42c213a5da7e719869d2301f6ab8/mako-1.4.1.tar.gz", hash = "sha256:d7904710b662996425a21627710c4777c45053146942cf8a7aebf757c92b8c27", size = 410165, upload-time = "2026-08-05T06:10:56.611Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a5/54/12ed58d458474aaab5c3d180173e745a4fe131bb330370596876d19ff60f/mako-1.4.1-py3-none-any.whl", hash = "sha256:a359d9a94a541213958742b2698d0a7757bb83551767bc468a74b9905aba9617", size = 80010, upload-time = "2026-08-05T06:10:58.248Z" },
]
```

```

[[package]]
name = "markupsafe"
version = "3.0.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/7e/99/7690b6d4034fffd95959cbe0c02de8deb3098cc577c67bb6a24fe5d7caa7/markupsafe-3.0.3.tar.gz",
hash = "sha256:722695808f4b6457b320fdc131280796bdceb04ab50fe1795cd540799ebe1698", size = 80313, upload-time = "2025-09-27T18:37:40.426Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/5a/72/147da192e38635ada20e0a2e1a51cf8823d2119ce8883f7053879c2199b5/markupsafe-3.0.3-cp312-cp312-m
acosx_10_13_x86_64.whl", hash = "sha256:d53197da72cc091b024dd97249dfc7794d6a56530370992a5e1a08983ad9230e", size = 11615, upload-time = "2025-09-27T18:
36:30.854Z" },
    { url = "https://files.pythonhosted.org/packages/9a/81/7e4e08678a1f98521201c3079f77db69fb552acd56067661f8c2f534a718/markupsafe-3.0.3-cp312-cp312-m
acosx_11_0_arm64.whl", hash = "sha256:1872df69a4de6aead3491198eaf13810b565bdbeec3ae2dc8780f14458ec73ce", size = 12020, upload-time = "2025-09-27T18:36
:31.971Z" },
    { url = "https://files.pythonhosted.org/packages/1e/2c/799f4742efc39633a1b54a92eec4082e4f815314869865d876824c257c1e/markupsafe-3.0.3-cp312-cp312-m
anylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:3a7e8ae81ae39e62a41ec302f972ba6ae23a5c5396c8e60113e9066ef893da
0d", size = 24332, upload-time = "2025-09-27T18:36:32.813Z" },
    { url = "https://files.pythonhosted.org/packages/3c/2e/8d0c2ab90a8c1d9a24f0399058ab8519a3279d1bd4289511d74e909f060e/markupsafe-3.0.3-cp312-cp312-m
anylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:d6dd0be5b5b189d31db7cda48b91d7e0a9795f31430b7f271219ab30f1d3ac9d"
, size = 22947, upload-time = "2025-09-27T18:36:33.86Z" },
    { url = "https://files.pythonhosted.org/packages/2c/54/887f3092a85238093a0b2154bd629c89444f395618842e8b0c41783898ea/markupsafe-3.0.3-cp312-cp312-m
anylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:94c6f0bb423f739146aec64595853541634bde58b2135f27f61c1ffd1cd4d16a", size = 21962, upl
oad-time = "2025-09-27T18:36:35.099Z" },
    { url = "https://files.pythonhosted.org/packages/c9/2f/336b8c7b6f4a4d95e91119dc8521402461b74a485558d8f238a68312f11c/markupsafe-3.0.3-cp312-cp312-m
usllinux_1_2_aarch64.whl", hash = "sha256:be8813b57049a7dc738189df53d69395eba14fb99345e0a5994914a3864c8a4b", size = 23760, upload-time = "2025-09-27T1
8:36:36.001Z" },
    { url = "https://files.pythonhosted.org/packages/32/43/67935f2b7e4982ffb50a4d169b724d74b62a3964bc1a9a527f5ac4f1ee2b/markupsafe-3.0.3-cp312-cp312-m
usllinux_1_2_riscv64.whl", hash = "sha256:83891d0e9fb81a825d9a6d61e3f07550ca70a076484292a70fde82c4b807286f", size = 21529, upload-time = "2025-09-27T1
8:36:36.906Z" },
    { url = "https://files.pythonhosted.org/packages/89/e0/4486f11e51bbba8b0c041098859e869e304d1c261e59244baa3d295d47b7/markupsafe-3.0.3-cp312-cp312-m
usllinux_1_2_x86_64.whl", hash = "sha256:77f0643abe7495da77fb436f50f8dab76dbc6e5fd25d39589a0f1fe6548bfa2b", size = 23015, upload-time = "2025-09-27T18
:36:37.868Z" },
    { url = "https://files.pythonhosted.org/packages/2f/e1/78ee7a023dac597a5825441ebd17170785a9dab23de95d2c7508ade94e0e/markupsafe-3.0.3-cp312-cp312-w
in32.whl", hash = "sha256:d88b440e37a16e651bda4c7c2b930eb586fd15ca7406cb39e211fcff3bf3017d", size = 14540, upload-time = "2025-09-27T18:36:38.761Z" },
    { url = "https://files.pythonhosted.org/packages/aa/5b/bec5aa9bbbb2c946ca2733ef9c4ca91c91b6a24580193e891b5f7dbe8e1e/markupsafe-3.0.3-cp312-cp312-w
in_amd64.whl", hash = "sha256:26a5784ded40c9e318cfc2bdb30fe164bdb8665ded9cd64d500a34fb42067b1c", size = 15105, upload-time = "2025-09-27T18:36:39.701Z
" },
    { url = "https://files.pythonhosted.org/packages/e5/f1/216fc1bbfd74011693a4fd837e7026152e89c4bcf3e77b6692fba9923123/markupsafe-3.0.3-cp312-cp312-w
in_arm64.whl", hash = "sha256:35add3b638a5d900e807944a078b51922212fb3dedb01633a8defc4b01a3c85f", size = 13906, upload-time = "2025-09-27T18:36:40.689Z
" },
    { url = "https://files.pythonhosted.org/packages/38/2f/907b9c7bbba283e68f20259574b13d005c121a0fa4c175f9bed27c4597ff/markupsafe-3.0.3-cp313-cp313-m
acosx_10_13_x86_64.whl", hash = "sha256:e1cf1972137e83c5d4c136c43ced9ac51d0e124706ee1c8aa8532c1287fa8795", size = 11622, upload-time = "2025-09-27T18:
36:41.777Z" },
    { url = "https://files.pythonhosted.org/packages/9c/d9/5f7756922cdd676869eca1c4e3c0cd0df60ed30199ffd775e319089cb3ed/markupsafe-3.0.3-cp313-cp313-m
acosx_11_0_arm64.whl", hash = "sha256:116bb52f642a37c115f517494ea5feb03889e04df47eeff5b130b1808ce7c219", size = 12029, upload-time = "2025-09-27T18:36
:43.257Z" },
    { url = "https://files.pythonhosted.org/packages/00/07/575a68c754943058c78f30db02ee03a64b3c638586fba6a6dd56830b30a3/markupsafe-3.0.3-cp313-cp313-m
anylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:133a43e73a802c5562be9bbcd03d090aa5a1fe899db609c29e8c8d815c5f6d
e6", size = 24374, upload-time = "2025-09-27T18:36:44.508Z" },
    { url = "https://files.pythonhosted.org/packages/a9/21/9b05698b46f218fc0e118e1f8168395c65c8a2c750ae2bab54fc4bd4e0e8/markupsafe-3.0.3-cp313-cp313-m
anylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:ccfcd093f13f0f0b7fdd0f198b90053bf7b2f02a3927a30e63f3ccc9df56b676"
, size = 22980, upload-time = "2025-09-27T18:36:45.385Z" },
    { url = "https://files.pythonhosted.org/packages/7f/71/544260864f893f18b6827315b988c146b559391e6e7e8f7252839b1b846a/markupsafe-3.0.3-cp313-cp313-m
anylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:509fa21c6deb7a7a273d629cf5ec029bc209d1a51178615ddf718f5918992ab9", size = 21990, upl
oad-time = "2025-09-27T18:36:46.916Z" },
    { url = "https://files.pythonhosted.org/packages/c2/28/b50fc2f74d1ad761af2f5dcce7492648b983d00a65b8c0e0cb457c82ebbe/markupsafe-3.0.3-cp313-cp313-m
usllinux_1_2_aarch64.whl", hash = "sha256:a4afe79fb3de0b7097d81da19090f4df4f8d3a2b3adaa8764138aac2e44f3af1", size = 23784, upload-time = "2025-09-27T1
8:36:47.884Z" },
    { url = "https://files.pythonhosted.org/packages/ed/76/104b2aa106a208da8b17a2fb72e033a5a9d7073c68f7e508b94916ed47a9/markupsafe-3.0.3-cp313-cp313-m
usllinux_1_2_riscv64.whl", hash = "sha256:795e7751525cae078558e679d646ae45574b47ed6e7771863fcc079a6171a0fc", size = 21588, upload-time = "2025-09-27T1
8:36:48.82Z" },
    { url = "https://files.pythonhosted.org/packages/b5/99/16a5eb2d140087ebd97180d95249b00a03aa87e29cc224056274f2e45fd6/markupsafe-3.0.3-cp313-cp313-m
usllinux_1_2_x86_64.whl", hash = "sha256:8485f406a96febb5140bfeca44a73e3ce5116b2501ac54fe953e488fb1d03b12", size = 23041, upload-time = "2025-09-27T18
:36:49.797Z" },
    { url = "https://files.pythonhosted.org/packages/19/bc/e7140ed90c5d61d77cea142eed9f9c303f4c4806f60a1044c13e3f1471d0/markupsafe-3.0.3-cp313-cp313-w
in32.whl", hash = "sha256:bdd37121970bfd8be76c5fb069c7751683bdf373db1ed6c010162b2a130248ed", size = 14543, upload-time = "2025-09-27T18:36:51.584Z" },
    { url = "https://files.pythonhosted.org/packages/05/73/c4abe620b841b6b791f2edc248f556900667a5a1cf023a6646967ae98335/markupsafe-3.0.3-cp313-cp313-w
in_amd64.whl", hash = "sha256:9a1abfdc021a164803f4d485104931fb8f8c1efd55bc6b748d2f5774e78b62c5", size = 15113, upload-time = "2025-09-27T18:36:52.537Z
" },
    { url = "https://files.pythonhosted.org/packages/f0/3a/fa34a0f7cfef23cf9500d68cb7c32dd64ffd58a12b09225fb03dd37d5b80/markupsafe-3.0.3-cp313-cp313-w
in_arm64.whl", hash = "sha256:7e68f88e5b8799aa49c85cd116c932a1ac15caaa3f5db09087854d218359e485", size = 13911, upload-time = "2025-09-27T18:36:53.513Z
" },
    { url = "https://files.pythonhosted.org/packages/e4/d7/e05cd7efe43a88a17a37b3ae96e79a19e846f3f456fe79c57ca61356ef01/markupsafe-3.0.3-cp313-cp313t-
macosx_10_13_x86_64.whl", hash = "sha256:218551f6df4868a8d527e3062d0fb968682fe92054e89978594c28e642c43a73", size = 11658, upload-time = "2025-09-27T18
:36:54.819Z" },
    { url = "https://files.pythonhosted.org/packages/99/9e/e412117548182ce2148bdeacdda3bb494260c0b0184360fe0d56389b523b/markupsafe-3.0.3-cp313-cp313t-
macosx_11_0_arm64.whl", hash = "sha256:3524b778fe5cfb3452a09d31e7b5adefeea8c5be1d43c4f810ba09f2ceb29d37", size = 12066, upload-time = "2025-09-27T18:3
6:55.714Z" },
    { url = "https://files.pythonhosted.org/packages/bc/e6/fa0ffcda717ef64a5108eaa7b4f5ed28d56122c9a6d70ab8b72f9f715c80/markupsafe-3.0.3-cp313-cp313t-
manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:4e885a3d1efa2eadc93c894a21770e4bc67899e3543680313b09f139e149a
b19", size = 25639, upload-time = "2025-09-27T18:36:56.908Z" },
    { url = "https://files.pythonhosted.org/packages/96/ec/2102e881fe9d25fc16cb4b25d5f5cde50970967ffa5dddafdb771237062d/markupsafe-3.0.3-cp313-cp313t-
manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:8709b08f4a89aa7586de0aadc8da56180242ee0ada3999749b183aa23df95025
", size = 23569, upload-time = "2025-09-27T18:36:57.913Z" },
    { url = "https://files.pythonhosted.org/packages/4b/30/6f2fce1f1f205fc9323255b216ca8a235b15860c34b6798f810f05828e32/markupsafe-3.0.3-cp313-cp313t-
manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:b8512a91625c9b3da6f127803b166b629725e68af71f8184ae7e7d54686a56d6", size = 23284, up
load-time = "2025-09-27T18:36:58.833Z" },
    { url = "https://files.pythonhosted.org/packages/58/47/4a0ccea4ab9f5dcb6f79c0236d954acb382202721e704223a8aafa38b5c8/markupsafe-3.0.3-cp313-cp313t-
musllinux_1_2_aarch64.whl", hash = "sha256:9b79b7a16f7fedff2495d684f2b59b0457c3b493778c9eed31111be64d58279f", size = 24801, upload-time = "2025-09-27T
18:36:59.739Z" },
    { url = "https://files.pythonhosted.org/packages/6a/70/3780e9b72180b6fecb83a4814d84c3bf4b4ae4bf0b19c27196104149734c/markupsafe-3.0.3-cp313-cp313t-
musllinux_1_2_riscv64.whl", hash = "sha256:12c63dfb4a98206f045aa9563db46507995f7ef6d83b2f68eda65c307c6829eb", size = 22769, upload-time = "2025-09-27T
18:37:00.719Z" },
    { url = "https://files.pythonhosted.org/packages/98/c5/c03c7f4125180fc215220c035beac6b9cb684bc7a067c84fc69414d315f5/markupsafe-3.0.3-cp313-cp313t-
musllinux_1_2_x86_64.whl", hash = "sha256:8f71bc33915be5186016f675cd83a1e08523649b0e33efdb898db577ef5bb009", size = 23642, upload-time = "2025-09-27T1
8:37:01.673Z" },
    { url = "https://files.pythonhosted.org/packages/80/d6/2d1b89f6ca4bff1036499b1e29a1d02d282259f3681540e16563f27ebc23/markupsafe-3.0.3-cp313-cp313t-
win32.whl", hash = "sha256:69c0b73548bc525c8cb9a251cddf1931d1db4d2258e9599c28c07ef3580ef354", size = 14612, upload-time = "2025-09-27T18:37:02.639Z" }
,
    { url = "https://files.pythonhosted.org/packages/2b/98/e48a4bfba0a0ffcf9925fe2d69240bfaa19c6f7507b8cd09c70684a53c1e/markupsafe-3.0.3-cp313-cp313t-
win_amd64.whl", hash = "sha256:1b4b79e8ebf6b55351f0d91fe80f893b4743f104bff22e90697db1590e47a218", size = 15200, upload-time = "2025-09-27T18:37:03.582
Z" },
    { url = "https://files.pythonhosted.org/packages/0e/72/e3cc540f351f316e9ed0f092757459afbc595824ca724cbc5a5d4263713f/markupsafe-3.0.3-cp313-cp313t-
win_arm64.whl", hash = "sha256:ad2cf8aa28b8c020ab2fc8287b0f823d0a7d8630784c31e9ee5edea20f406287", size = 13973, upload-time = "2025-09-27T18:37:04.929
Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/33/8a/8e42d4838cd89b7dde187011e97fe6c3af66d8c044997d2183fbd6d31352/markupsafe-3.0.3-cp314-cp314-macosx_10_13_x86_64.whl", hash = "sha256:eaa9599de571d72e2daf60164784109f19978b327a3910d3e9de8c97b5b70cfe", size = 11619, upload-time = "2025-09-27T18:37:06.342Z" },
    { url = "https://files.pythonhosted.org/packages/b5/64/7660f8a4a8e53c924d0fa05dc3a55c9cee10bbd82b11c5afb27d44b096ce/markupsafe-3.0.3-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:c47a551199eb8eb2121d4f0f15ae0f923d31350ab9280078d1e5f12b249e0026", size = 12029, upload-time = "2025-09-27T18:37:07.213Z" },
    { url = "https://files.pythonhosted.org/packages/da/ef/e648bfd021127bef5fa12e1720ffed0c6cbb8310c8d9bea7266337ff06de/markupsafe-3.0.3-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:f34c41761022dd093b4b6896d4810782ffbabe30f2d443ff5f083e0cbbb8c737", size = 24408, upload-time = "2025-09-27T18:37:09.572Z" },
    { url = "https://files.pythonhosted.org/packages/41/3c/a36c2450754618e62008bf7435ccb0f88053e07592e6028a34776213d877/markupsafe-3.0.3-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:457a69a9577064c05a97c41f4e65148652db078a3a509039e64d3467b9e7ef97", size = 23005, upload-time = "2025-09-27T18:37:10.58Z" },
    { url = "https://files.pythonhosted.org/packages/bc/20/b7fdf89a8456b099837cd1dc21974632a02a999ec9bf7ca3e490aacd98e7/markupsafe-3.0.3-cp314-cp314-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:e8afc3f2ccfa24215f8cb28dcf43f0113ac3c37c2f0f0806d8c70e4228c5cf4d", size = 22048, upload-time = "2025-09-27T18:37:11.547Z" },
    { url = "https://files.pythonhosted.org/packages/9a/a7/591f592afdc734f47db08a75793a55d7fbcc6902a723ae4cfbab61010cc5/markupsafe-3.0.3-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:ec15a59cf5af7be74194f7ab02d0f59a62bdcf1a537677ce67a2537c9b87fcda", size = 23821, upload-time = "2025-09-27T18:37:12.48Z" },
    { url = "https://files.pythonhosted.org/packages/7d/33/45b24e4f44195b26521bc6f1a82197118f74df348556594bd2262bda1038/markupsafe-3.0.3-cp314-cp314-musllinux_1_2_riscv64.whl", hash = "sha256:0eb9ff8191e8498cca014656ae6b8d61f39da5f95b488805da4bb029cccbfbaf", size = 21606, upload-time = "2025-09-27T18:37:13.485Z" },
    { url = "https://files.pythonhosted.org/packages/ff/0e/53dfaca23a69fbfbbf17a4b64072090e70717344c52eaaaa9c5ddff1e5f0/markupsafe-3.0.3-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:2713baf880df847f2bece4230d4d094280f4e67b1e813eec43b4c0e144a34ffe", size = 23043, upload-time = "2025-09-27T18:37:14.408Z" },
    { url = "https://files.pythonhosted.org/packages/46/11/f333a06fc16236d5238bfe74daccbca41459dcd8d1fa952e8fbd5dccfb70/markupsafe-3.0.3-cp314-cp314-win32.whl", hash = "sha256:729586769a26dbceff69f7a7dbbf59ab6572b99d94576a5592625d5b411576b9", size = 14747, upload-time = "2025-09-27T18:37:15.36Z" },
    { url = "https://files.pythonhosted.org/packages/28/52/182836104b33b444e400b14f797212f720cbc9ed6ba34c800639d154e821/markupsafe-3.0.3-cp314-cp314-win_amd64.whl", hash = "sha256:bdc919ead48f234740ad807933cdf545180bfbe9342c2bb451556db2ed958581", size = 15341, upload-time = "2025-09-27T18:37:16.496Z" },
    { url = "https://files.pythonhosted.org/packages/6f/18/acf23e91bd94fd7b3031558b1f013adfa21a8e407a3fdb32745538730382/markupsafe-3.0.3-cp314-cp314-win_arm64.whl", hash = "sha256:5a7d5dc5140555cf21a6fefbdbf8723f06fcd2f63ef108f2854de715e4422cb4", size = 14073, upload-time = "2025-09-27T18:37:17.476Z" },
    { url = "https://files.pythonhosted.org/packages/3c/f0/57689aa4076e1b43b15fdfa646b04653969d50cf30c32a102762be2485da/markupsafe-3.0.3-cp314-cp314t-macosx_10_13_x86_64.whl", hash = "sha256:1353ef0c1b138e1907ae78e2f6c63ff67501122006b0f9abad68fda5f4ffc6ab", size = 11661, upload-time = "2025-09-27T18:37:18.453Z" },
    { url = "https://files.pythonhosted.org/packages/89/c3/2e67a7ca217c6912985ec766c6393b636fb0c2344443ff9d91404dc4c79f/markupsafe-3.0.3-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:1085e7fbddd3be5f89cc898938f42c0b3c711fdcb37d75221de2666af647c175", size = 12069, upload-time = "2025-09-27T18:37:19.332Z" },
    { url = "https://files.pythonhosted.org/packages/f0/00/be561dce4e6ca66b15276e184ce4b8aec61fe83662cce2f7d72bd3249d28/markupsafe-3.0.3-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:1b52b4fb9df4eb9ae465f8d0c228a00624de2334f216f178a995ccdcf82c4634", size = 25670, upload-time = "2025-09-27T18:37:20.245Z" },
    { url = "https://files.pythonhosted.org/packages/50/09/c419f6f5a92e5fadde27efd190eca90f05e1261b10dbd8cbcb39cd8ea1dc/markupsafe-3.0.3-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:fed51ac40f757d41b7c48425901843666a6677e3e8eb0abcff09e4ba6e664f50", size = 23598, upload-time = "2025-09-27T18:37:21.177Z" },
    { url = "https://files.pythonhosted.org/packages/22/44/a0681611106e0b2921b3033fc19bc53323e0b50bc70cffdd19f7d679bb66/markupsafe-3.0.3-cp314-cp314t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:f190daf01f13c72eac4efd5c430a8de82489d9cff23c364c3ea822545032993e", size = 23261, upload-time = "2025-09-27T18:37:22.167Z" },
    { url = "https://files.pythonhosted.org/packages/5f/57/1b0b3f100259dc9fffe780cfb60d4be71375510e435efec3d116b6436d43/markupsafe-3.0.3-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:e56b7d45a839a697b5eb268c82a71bd8c7f6c94d6fd50c3d577fa39a9f1409f5", size = 24835, upload-time = "2025-09-27T18:37:23.296Z" },
    { url = "https://files.pythonhosted.org/packages/26/6a/4bf6d0c97c4920f1597cc14dd720705eca0bf7c787aebc6bb4d1bead5388/markupsafe-3.0.3-cp314-cp314t-musllinux_1_2_riscv64.whl", hash = "sha256:f3e98bb3798ead92273dc0e5fd0f31ade220f59a266ffd8a4f6065e0a3ce0523", size = 22733, upload-time = "2025-09-27T18:37:24.237Z" },
    { url = "https://files.pythonhosted.org/packages/14/c7/ca723101509b518797fedc2fdf79ba57f886b4aca8a7d31857ba3ee8281f/markupsafe-3.0.3-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:5678211cb9333a6468fb8d8be0305520aa073f50d17f089b5b4b477ea6e67fdc", size = 23672, upload-time = "2025-09-27T18:37:25.271Z" },
    { url = "https://files.pythonhosted.org/packages/fb/df/5bd7a48c256faecd1d36edc13133e51397e41b73bb77e1a69deab746ebac/markupsafe-3.0.3-cp314-cp314t-win32.whl", hash = "sha256:915c04ba3851909ce68ccc2b8e2cd691618c4dc4c4232fb7982bca3f41fd8c3d", size = 14819, upload-time = "2025-09-27T18:37:26.285Z" },
    { url = "https://files.pythonhosted.org/packages/1a/8a/0402ba61a2f16038b48b39bccca271134be00c5c9f0f623208399333c448/markupsafe-3.0.3-cp314-cp314t-win_amd64.whl", hash = "sha256:4faffd047e07c38848ce017e8725090413cd80cbc23d86e55c587bf979e579c9", size = 15426, upload-time = "2025-09-27T18:37:27.316Z" },
    { url = "https://files.pythonhosted.org/packages/70/bc/6f1c2f612465f5fa89b95bead1f44dcb607670fd42891d8fdcd5d039f4f4/markupsafe-3.0.3-cp314-cp314t-win_arm64.whl", hash = "sha256:32001d6a8fc98c8cb5c947787c5d08b0a50663d139f1305bac5885d98d9b40fa", size = 14146, upload-time = "2025-09-27T18:37:28.327Z" },
]

[[package]]
name = "marshmallow"
version = "3.26.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "packaging" },
]
sdist = { url = "https://files.pythonhosted.org/packages/55/79/de6c16cc902f4fc372236926b0ce2ab7845268dcc30fb2fbb7f71b418631/marshmallow-3.26.2.tar.gz", hash = "sha256:bbe2adb5a03e6e3571b573f42527c6fe926e17467833660bebd11593ab8dfd57", size = 222095, upload-time = "2025-12-22T06:53:53.309Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/be/2f/5108cb3ee4ba6501748c4908b908e55f42a5b66245b4cfe0c99326e1ef6e/marshmallow-3.26.2-py3-none-any.whl", hash = "sha256:013fa8a3c4c276c24d26d84ce934dc964e2aa794345a0f8c7e5a7191482c8a73", size = 50964, upload-time = "2025-12-22T06:53:51.801Z" },
]

[[package]]
name = "mcp"
version = "2.0.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "httpx2" },
    { name = "jsonschema" },
    { name = "mcp-types" },
    { name = "opentelemetry-api" },
    { name = "pydantic" },
    { name = "pyjwt", extra = ["crypto"] },
    { name = "python-multipart" },
    { name = "pywin32", marker = "sys_platform == 'win32'" },
    { name = "sse-starlette" },
    { name = "starlette" },
    { name = "typing-extensions" },
    { name = "typing-inspection" },
    { name = "uvicorn", marker = "sys_platform != 'emscripten'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/74/33/32d4dff2c95bb5d897c3ef4c83649a08996b17b58f0a326d2495d4c81179/mcp-2.0.0.tar.gz", hash = "sha256:0f440e735c13ece8bb19bc62cf0b86f4313448432fbb77d35e14034f4e050728", size = 1662284, upload-time = "2026-07-28T13:45:32.346Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/67/72/7d7897418912c1d12e87556630dfb7bf0eac71160e9bef8b447960804ee3/mcp-2.0.0-py3-none-any.whl", h
```

```
ash = "sha256:1cb4c75d2d2c7b8c1d756355e5d82a39f2822cc7f13e22a2051d7ca3592349d6", size = 349980, upload-time = "2026-07-28T13:45:28.853Z" },
]

[[package]]
name = "mcp-types"
version = "2.0.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "pydantic" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/bb/56/9b8e1c152f61f6c6b07c4b5896c88c7d0ae90bac6ee6306f852fcc5c1eb0/mcp_types-2.0.0.tar.gz", h
ash = "sha256:d7d939b9285c9961ae8866ba75ef85da34d12bafe276efbf4eb6a131786d8379", size = 66632, upload-time = "2026-07-28T13:45:33.804Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/f5/4c/c78d78c3d52b0ac594ad7cc8ef5972adfe070e3597a8a4c6ce0cd39196ea/mcp_types-2.0.0-py3-none-any.w
hl", hash = "sha256:6b2de797ca2797f568b79529e1b25948e34de511bcc0bd82fef1039a6d1b8eb0", size = 69649, upload-time = "2026-07-28T13:45:30.713Z" },
]

[[package]]
name = "multidict"
version = "6.7.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/1a/c2/c2d94cbe6ac1753f3fc980da97b3d930efe1da3af3c9f5125354436c073d/multidict-6.7.1.tar.gz", h
ash = "sha256:ec6652a1bee61c53a3e5776b6049172c53b6aaba34f18c9ad04f82712bac623d", size = 102010, upload-time = "2026-01-26T02:46:45.979Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/8d/9c/f20e0e2cf80e4b2e4b1c365bf5fe104ee633c751a724246262db8f1a0b13/multidict-6.7.1-cp312-cp312-ma
cosx_10_13_universal2.whl", hash = "sha256:a90f75c956e32891a4eda3639ce6dd86e87105271f43d43442a3aedf3cddf172", size = 76893, upload-time = "2026-01-26T
02:43:52.754Z" },
    { url = "https://files.pythonhosted.org/packages/fe/cf/18ef143a81610136d3da8193da9d80bfe1cb548a1e2d1c775f26b23d024a/multidict-6.7.1-cp312-cp312-ma
cosx_10_13_x86_64.whl", hash = "sha256:3fccb473e87eaa138268905 3e4a4618e7ba7b9b9b8d6adf2027ee474597128cd", size = 45456, upload-time = "2026-01-26T02:4
3:53.893Z" },
    { url = "https://files.pythonhosted.org/packages/a9/65/1caac9d4cd32e8433908683446eebc953e82d22b03d10d41a5f0fefe991b/multidict-6.7.1-cp312-cp312-ma
cosx_11_0_arm64.whl", hash = "sha256:b0fa96985700739c4c7853a43c0b3e169360d6855780021bfc6d0f1ce7c123e7", size = 43872, upload-time = "2026-01-26T02:43:
55.041Z" },
    { url = "https://files.pythonhosted.org/packages/cf/3b/d6bd75dc4f3ff7c73766e04e705b00ed6dbbaccf670d9e05a12b006f5a21/multidict-6.7.1-cp312-cp312-ma
nylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:cb2a55f408c3043e42b40cc8eecd575afa27b7e0b956dfb190de0f8499a57a53", size = 25
1018, upload-time = "2026-01-26T02:43:56.198Z" },
    { url = "https://files.pythonhosted.org/packages/fd/80/c959c5933adedb9ac15152e4067c702a808ea183a8b64cf8f31af8ad3155/multidict-6.7.1-cp312-cp312-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:eb0ce7b2a32d09892b3dd6cc44877a0d02a33241fafca5f25c8b6b62374f8b7
5", size = 258883, upload-time = "2026-01-26T02:43:57.499Z" },
    { url = "https://files.pythonhosted.org/packages/86/85/7ed40adafea3d4f1c8b916e3b5cc3a8e07dfcdcb9cd72800f4ed3ca1b387/multidict-6.7.1-cp312-cp312-ma
nylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:c3a32d23520ee37bf327d1e1a656fec76a2edd5c038bf43eddfa0572ec49c60b",
 size = 242413, upload-time = "2026-01-26T02:43:58.755Z" },
    { url = "https://files.pythonhosted.org/packages/d2/57/b8565ff533e48595503c785f8361ff9a4fde4d67de25c207cd0ba3befd03/multidict-6.7.1-cp312-cp312-ma
nylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:9c90fed18bffc0189ba814749fdcc102b536e83a9f738a9003e569acd540a73
3", size = 268404, upload-time = "2026-01-26T02:44:00.216Z" },
    { url = "https://files.pythonhosted.org/packages/e0/50/9810c5c29350f7258180dfdcb2e52783a0632862eb334c4896ac717cebcb/multidict-6.7.1-cp312-cp312-ma
nylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:da62917e6076f512dacc fbbde27f46fed1c98fee202f0559adec8ee0de67f71a", si
ze = 269456, upload-time = "2026-01-26T02:44:02.202Z" },
    { url = "https://files.pythonhosted.org/packages/f3/8d/5e5be3ced1d12966fefb5c4ea3b2a5b480afcea36406559442c6e31d4a48/multidict-6.7.1-cp312-cp312-ma
nylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:bfde23ef6ed9db7eaee6c37dcec08524cb43903c60b285b172b6c094711b3961",
 size = 256322, upload-time = "2026-01-26T02:44:03.56Z" },
    { url = "https://files.pythonhosted.org/packages/31/6e/d8a26d81ac166a5592782d208dd90dfdc0a7a218adaa52b45a672b46c122/multidict-6.7.1-cp312-cp312-mu
sllinux_1_2_aarch64.whl", hash = "sha256:3758692429e4e32f1ba0df23219cd0b4fc0a52f476726fff9337d1a57676a582", size = 253955, upload-time = "2026-01-26T0
2:44:04.845Z" },
    { url = "https://files.pythonhosted.org/packages/59/4c/7c672c8aad41534ba619bcd4ade7a0dc87ed6b8b5c06149b85d3dd03f0cd/multidict-6.7.1-cp312-cp312-mu
sllinux_1_2_armv7l.whl", hash = "sha256:398c1478926eca669f2fd6a5856b6de9c0acf23a2cb59a14c0ba5844fa38077e", size = 251254, upload-time = "2026-01-26T02
:44:06.133Z" },
    { url = "https://files.pythonhosted.org/packages/7b/bd/84c24de512cbafbdbc39439f74e967f19570ce7924e3007174a29c348916/multidict-6.7.1-cp312-cp312-mu
sllinux_1_2_i686.whl", hash = "sha256:c102791b1c4f3ab36ce4101154549105a53dc828f016356b3e3bcae2e3a039d3", size = 252059, upload-time = "2026-01-26T02:4
4:07.518Z" },
    { url = "https://files.pythonhosted.org/packages/fa/ba/f5449385510825b73d01c2d4087bf6d2fccc20a2d42ac34df93191d3dd03/multidict-6.7.1-cp312-cp312-mu
sllinux_1_2_ppc64le.whl", hash = "sha256:a088b62bd733e2ad12c50dad01b7d0166c30287c166e137433d3b410add807a6", size = 263588, upload-time = "2026-01-26T0
2:44:09.382Z" },
    { url = "https://files.pythonhosted.org/packages/d7/11/afc7c677f68f75c84a69fe37184f0f82fce13ce4b92f49f3db280b7e92b3/multidict-6.7.1-cp312-cp312-mu
sllinux_1_2_s390x.whl", hash = "sha256:3d51ff4785d58d3f6c91bdbffcb5e1f7ddfda557727043aa20d20ec4f65e324a", size = 259642, upload-time = "2026-01-26T02:
44:10.73Z" },
    { url = "https://files.pythonhosted.org/packages/2b/17/ebb9644da78c4ab36403739e0e6e0e30ebb135b9caf3440825001a0bddcb/multidict-6.7.1-cp312-cp312-mu
sllinux_1_2_x86_64.whl", hash = "sha256:fc5907494fccf3e7d3f94f95c91d6336b092b5fc83811720fae5e2765890dfba", size = 251377, upload-time = "2026-01-26T02
:44:12.042Z" },
    { url = "https://files.pythonhosted.org/packages/ca/a4/840f5b97339e27846c46307f2530a2805d9d537d8b8bd416af031cad7fa0/multidict-6.7.1-cp312-cp312-wi
n32.whl", hash = "sha256:28ca5ce2fd9716631133d0e9a9b9a745ad7f60bac2bccafb56aa380fc0b6c511", size = 41887, upload-time = "2026-01-26T02:44:14.245Z" },
    { url = "https://files.pythonhosted.org/packages/80/31/0b2517913687895f5904325c2069d6a3b78f66cc641a86a2baf75a05dcbb/multidict-6.7.1-cp312-cp312-wi
n_amd64.whl", hash = "sha256:fcee94dfbd638784645b066074b338bc9cc155d4b4bffa4adce1615c5a426c19", size = 46053, upload-time = "2026-01-26T02:44:15.371Z"
 },
    { url = "https://files.pythonhosted.org/packages/0c/5b/aba28e4ee4006ae4c7df8d327d31025d760ffa992ea23812a601d226e682/multidict-6.7.1-cp312-cp312-wi
n_arm64.whl", hash = "sha256:ba0a9fb644d0c1a2194cf7ffb043bd852cea63a57f66fbd33959f7dae18517bf", size = 43307, upload-time = "2026-01-26T02:44:16.852Z"
 },
    { url = "https://files.pythonhosted.org/packages/f2/22/929c141d6c0dba87d3e1d38fbdf1ba8baba86b7776469f2bc2d3227a1e67/multidict-6.7.1-cp313-cp313-ma
cosx_10_13_universal2.whl", hash = "sha256:2b41f5fed0ed563624f1c17630cb9941cf2309d4df00e494b551b5f3e3d67a23", size = 76174, upload-time = "2026-01-26T
02:44:18.509Z" },
    { url = "https://files.pythonhosted.org/packages/c7/75/bc704ae15fee974f8fccd871305e254754167dce5f9e42d88a2def741a1d/multidict-6.7.1-cp313-cp313-ma
cosx_10_13_x86_64.whl", hash = "sha256:84e61e3af5463c19b67ced91f6c634effb89ef8bfc5ca0267f954451ed4bb6a2", size = 45116, upload-time = "2026-01-26T02:4
4:19.745Z" },
    { url = "https://files.pythonhosted.org/packages/79/76/55cd7186f498ed080a18440c9013011eb548f77ae1b297206d030eb1180a/multidict-6.7.1-cp313-cp313-ma
cosx_11_0_arm64.whl", hash = "sha256:935434b9853c7c112eee7ac891bc4cb86455aa631269ae35442cb316790c1445", size = 43524, upload-time = "2026-01-26T02:44:
21.571Z" },
    { url = "https://files.pythonhosted.org/packages/e9/3c/414842ef8d5a1628d68edee29ba0e5bcf235dbfb3ccd3ea303a7fe8c72ff/multidict-6.7.1-cp313-cp313-ma
nylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:432feb25a1cb67fe82a9680b4d65fb542e4635cb3166cd9c01560651ad60f177", size = 24
9368, upload-time = "2026-01-26T02:44:22.803Z" },
    { url = "https://files.pythonhosted.org/packages/f6/32/befed7f74c458b4a525e60519fe8d87eef72bb1e99924fa2b0f9d97a221e/multidict-6.7.1-cp313-cp313-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:e82d14e3c948952a1a85503817e038cba5905a3352de76b9a465075d072fba2
3", size = 256952, upload-time = "2026-01-26T02:44:24.306Z" },
    { url = "https://files.pythonhosted.org/packages/03/d6/c878a44ba877f366630c860fdf74bfb203c33778f12b6ac274936853c451/multidict-6.7.1-cp313-cp313-ma
nylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:4cfb48c6ea66c83bcaaf7e4dfa7ec1b6bbcf751b7db85a328902796dfde4c060",
 size = 240317, upload-time = "2026-01-26T02:44:25.772Z" },
    { url = "https://files.pythonhosted.org/packages/68/49/57421b4d7ad2e9e60e25922b08ceb37e077b90444bde6ead629095327a6f/multidict-6.7.1-cp313-cp313-ma
nylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:1d540e51b7e8e170174555edecddbd5538105443754539193e3e1061864d444
d", size = 267132, upload-time = "2026-01-26T02:44:27.648Z" },
    { url = "https://files.pythonhosted.org/packages/b7/fe/ec0edd52ddbcea2a2e89e174f0206444a61440b40f39704e64dc807a70bd/multidict-6.7.1-cp313-cp313-ma
nylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:273d23f4b40f3dce4d6c8a821c741a86dec62cded82e1175ba3d99be128147ed", si
ze = 268140, upload-time = "2026-01-26T02:44:29.588Z" },
    { url = "https://files.pythonhosted.org/packages/b0/73/6e1b01cbeb458807aa0831742232dbdd1fa92bfa33f52a3f176b4ff3dc11/multidict-6.7.1-cp313-cp313-ma
nylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9d624335fd4fa1c08a53f8b4be7676ebde19cd092b3895c421045ca87895b429",
 size = 254277, upload-time = "2026-01-26T02:44:30.902Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/6a/b2/5fb8c124d7561a4974c342bc8c778b471ebbeb3cc17df696f034a7e9afe7/multidict-6.7.1-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:12fad252f8b267cc75b66e8fc51b3079604e8d43a75428ffe193cd9e2195dfd6", size = 252291, upload-time = "2026-01-26T02:44:32.31Z" },
    { url = "https://files.pythonhosted.org/packages/5a/96/51d4e4e06bcce92577fcd488e22600bd38e4fd59c20cb49434d054903bd2/multidict-6.7.1-cp313-cp313-musllinux_1_2_armv7l.whl", hash = "sha256:03ede2a6ffbe8ef936b92cb4529f27f42be7f56afcdab5ab739cd5f27fb1cbf9", size = 250156, upload-time = "2026-01-26T02:44:33.734Z" },
    { url = "https://files.pythonhosted.org/packages/db/6b/420e173eec5fba721a50e2a9f89eda89d9c98fded1124f8d5c675f7a0c0f/multidict-6.7.1-cp313-cp313-musllinux_1_2_i686.whl", hash = "sha256:90efbcf47dbe33dcf643a1e400d67d59abeac5db07dc3f27d6bdeae497a2198c", size = 249742, upload-time = "2026-01-26T02:44:35.222Z" },
    { url = "https://files.pythonhosted.org/packages/44/a3/ec5b5bd98f306bc2aa297b8c6f11a46714a56b1e6ef5ebda50a4f5d7c5fb/multidict-6.7.1-cp313-cp313-musllinux_1_2_ppc64le.whl", hash = "sha256:5c4b9bfc148f5a91be9244d6264c53035c8a0dcd2f51f1c3c6e30e30ebaa1c84", size = 262221, upload-time = "2026-01-26T02:44:36.604Z" },
    { url = "https://files.pythonhosted.org/packages/cd/f7/e8c0d0da0cd1e28d10e624604e1a36bcc3353aaebdfdc3a43c72bc683a12/multidict-6.7.1-cp313-cp313-musllinux_1_2_s390x.whl", hash = "sha256:401c5a650f3add2472d1d288c26deebc540f99e2fb83e9525007a74cd2116f1d", size = 258664, upload-time = "2026-01-26T02:44:38.008Z" },
    { url = "https://files.pythonhosted.org/packages/52/da/151a44e8016dd33feed44f730bd856a66257c1ee7aed4f44b649fb7edeb3/multidict-6.7.1-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:97891f3b1b3ffbded884e2916cacf3c6fc87b66bb0dde46f7357404750559f33", size = 249490, upload-time = "2026-01-26T02:44:39.386Z" },
    { url = "https://files.pythonhosted.org/packages/87/af/a3b86bf9630b732897f6fc3f4c4714b90aa4361983ccbdcd6c0339b21b0c/multidict-6.7.1-cp313-cp313-win32.whl", hash = "sha256:e1c5988359516095535c4301af38d8a8838534158f649c05dd1050222321bcb3", size = 41695, upload-time = "2026-01-26T02:44:41.318Z" },
    { url = "https://files.pythonhosted.org/packages/b2/35/e994121b0e90e46134673422dd564623f93304614f5d11886b1b3e06f503/multidict-6.7.1-cp313-cp313-win64.whl", hash = "sha256:960c83bf01a95b12b08fd54324a4eb1d5b52c88932b5cba5d6e712bb3ed12eb5", size = 45884, upload-time = "2026-01-26T02:44:42.488Z" },
    { url = "https://files.pythonhosted.org/packages/ca/61/42d3e5dbf661242a69c97ea363f2d7b46c567da8eadef8890022be6e2ab0/multidict-6.7.1-cp313-cp313-win64.whl", hash = "sha256:563fe25c678aaba333d5399408f5ec3c383ca5b663e7f774dd179a520b8144df", size = 43122, upload-time = "2026-01-26T02:44:43.664Z" },
    { url = "https://files.pythonhosted.org/packages/6d/b3/e6b21c6c4f314bb956016b0b3ef2162590a529b84cb831c257519e7fde44/multidict-6.7.1-cp313-cp313t-macosx_10_13_universal2.whl", hash = "sha256:c76c4bec1538375dad9d452d246ca5368ad6e1c9039dadcf007ae59c70619ea1", size = 83175, upload-time = "2026-01-26T02:44:44.894Z" },
    { url = "https://files.pythonhosted.org/packages/fb/76/23ecd2abfe0957b234f6c960f4ade497f55f2c16aeb684d4ecdbf1c95791/multidict-6.7.1-cp313-cp313t-macosx_10_13_x86_64.whl", hash = "sha256:57b46b24b5d5ebcc978da4ec23a819a9402b4228b8a90d9c656422b4bdd8a963", size = 48460, upload-time = "2026-01-26T02:44:46.106Z" },
    { url = "https://files.pythonhosted.org/packages/c4/57/a0ed92b23f3a042c36bc4227b72b97eca803f5f1801c1ab77c8a212d455e/multidict-6.7.1-cp313-cp313t-macosx_11_0_arm64.whl", hash = "sha256:e954b24433c768ce78ab7929e84ccf3422e46deb45a4dc9f93438f8217fa2d34", size = 46930, upload-time = "2026-01-26T02:44:47.278Z" },
    { url = "https://files.pythonhosted.org/packages/b5/66/02ec7ace29162e447f6382c495dc95826bf931d3818799bbef11e8f7df1a/multidict-6.7.1-cp313-cp313t-manylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:3bd231490fa7217cc832528e1cd8752a96f0125ddd2b5749390f7c3ec8721b65", size = 242582, upload-time = "2026-01-26T02:44:48.604Z" },
    { url = "https://files.pythonhosted.org/packages/58/18/64f5a795e7677670e872673aca23416251469627 4597b3708b2c0d276cce/multidict-6.7.1-cp313-cp313t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:253282d70d67885a15c8a7716f3a73edf2d635793ceda8173b9ecc21f2fb8292", size = 250031, upload-time = "2026-01-26T02:44:50.544Z" },
    { url = "https://files.pythonhosted.org/packages/c8/ed/e192291dbbe51a8290c5686f482084d31bcd9d09af24f63358c3d42fd284/multidict-6.7.1-cp313-cp313t-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:0b4c48648d7649c9335cf1927a8b87fa692de3dcb15faa676c6a6f1f1aabda43", size = 228596, upload-time = "2026-01-26T02:44:51.951Z" },
    { url = "https://files.pythonhosted.org/packages/1e/7e/3562a15a60cf747397e7f2180b0a11dc0c38d9175a650e75fa1b4d325e15/multidict-6.7.1-cp313-cp313t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:98bc624954ec4d2c7cb074b8eefc2b5d0ce7d482e410df446414355d158fe4ca", size = 257492, upload-time = "2026-01-26T02:44:53.902Z" },
    { url = "https://files.pythonhosted.org/packages/24/02/7d0f9eae92b5249bb50ac1595b295f10e263dd0078ebb55115c31e0eaccd/multidict-6.7.1-cp313-cp313t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:1b99af4d9eec0b49927b4402bcbb58dea89d3e0db8806a4086117019939ad3dd", size = 255899, upload-time = "2026-01-26T02:44:55.316Z" },
    { url = "https://files.pythonhosted.org/packages/00/e3/9b60ed9e23e64c73a5cde95269ef1330678e9c6e34dd4eb6b431b85b5a10/multidict-6.7.1-cp313-cp313t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:6aac4f16b472d5b7dc6f66a0d49dd57b0e0902090be16594dc9ebfd3d17c47e7", size = 247970, upload-time = "2026-01-26T02:44:56.783Z" },
    { url = "https://files.pythonhosted.org/packages/3e/06/538e58a63ed5cfb0bd4517e346b91da32fde409d839720f664e9a4ae4f9d/multidict-6.7.1-cp313-cp313t-musllinux_1_2_aarch64.whl", hash = "sha256:21f830fe223215dffd51f538e78c172ed7c7f60c9b96a2bf05c4848ad49921c3", size = 245060, upload-time = "2026-01-26T02:44:58.195Z" },
    { url = "https://files.pythonhosted.org/packages/b2/2f/d743a3045a97c895d401e9bd29aaa09b94f5cbdf1bd561609e5a6c431c70/multidict-6.7.1-cp313-cp313t-musllinux_1_2_armv7l.whl", hash = "sha256:f5dd81c45b05518b9aa4da4aa74e1c93d715efa234fd3e8a179df611cc85e5f4", size = 235888, upload-time = "2026-01-26T02:44:59.57Z" },
    { url = "https://files.pythonhosted.org/packages/38/83/5a325cac191ab28b63c52f14f1131f3b0a55ba3b9aa65a6d0bf2a9b921a0/multidict-6.7.1-cp313-cp313t-musllinux_1_2_i686.whl", hash = "sha256:eb304767bca2bb92fb9c5bd33cedc95baee5bb5f6c88e63706533a1c06ad08c8", size = 243554, upload-time = "2026-01-26T02:45:01.054Z" },
    { url = "https://files.pythonhosted.org/packages/20/1f/9d2327086bd15da2725ef6aae624208e2ef828ed99892b17f60c344e57ed/multidict-6.7.1-cp313-cp313t-musllinux_1_2_ppc64le.whl", hash = "sha256:c9035dde0f916702850ef66460bc4239d89d08df4d02023a5926e7446724212c", size = 252341, upload-time = "2026-01-26T02:45:02.484Z" },
    { url = "https://files.pythonhosted.org/packages/e8/2c/2a1aa0280cf579d0f6eed8ee5211c4f1730bd7e06c636ba2ee6aafda302e/multidict-6.7.1-cp313-cp313t-musllinux_1_2_s390x.whl", hash = "sha256:af959b9beeb66c822380f222f0e0a1889331597e81f1ded7f374f3ecb0fd6c52", size = 246391, upload-time = "2026-01-26T02:45:03.862Z" },
    { url = "https://files.pythonhosted.org/packages/e5/03/7ca022ffc36c5a3f6e03b179a5ceb829be9da5783e6fe395f347c0794680/multidict-6.7.1-cp313-cp313t-musllinux_1_2_x86_64.whl", hash = "sha256:41f2952231456154ee479651491e94118229844dd7226541788be783be2b5108", size = 243422, upload-time = "2026-01-26T02:45:05.296Z" },
    { url = "https://files.pythonhosted.org/packages/dc/1d/b31650eab6c5778aceed46ba735bd97f7c7d2f54b319fa916c0f96e7805b/multidict-6.7.1-cp313-cp313t-win32.whl", hash = "sha256:df9f19c28adcb40b6aae30bbaa1478c389efd50c28d541d76760199fc1037c32", size = 47770, upload-time = "2026-01-26T02:45:06.754Z" },
    { url = "https://files.pythonhosted.org/packages/ac/5b/2d2d1d522e51285bd61b1e20df8f47ae1a9d80839db0b24ea783b3832832/multidict-6.7.1-cp313-cp313t-win_amd64.whl", hash = "sha256:d54ecf9f301853f2c5e802da559604b3e95bb7a3b01a9c295c6ee591b9882de8", size = 53109, upload-time = "2026-01-26T02:45:08.044Z" },
    { url = "https://files.pythonhosted.org/packages/3d/a3/cc409ba012c83ca024a308516703cf339bdc4b696195644a7215a5164a24/multidict-6.7.1-cp313-cp313t-win_arm64.whl", hash = "sha256:5a37ca18e360377cfda1d62f5f382ff41f2b8c4ccb329ed974cc2e1643440118", size = 45573, upload-time = "2026-01-26T02:45:09.349Z" },
    { url = "https://files.pythonhosted.org/packages/91/cc/db74228a8be41884a567e88a62fd589a913708fcf180d029898c17a9a371/multidict-6.7.1-cp314-cp314-macosx_10_15_universal2.whl", hash = "sha256:8f333ec9c5eb1b7105e3b84b53141e66ca05a19a605368c55450b6ba208cb9ee", size = 75190, upload-time = "2026-01-26T02:45:10.651Z" },
    { url = "https://files.pythonhosted.org/packages/d5/22/492f2246bb5b534abd44804292e81eeaf835388901f0c574bac4eeec73c5/multidict-6.7.1-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:a407f13c188f804c759fc6a9f88286a565c242a76b27626594c133b82883b5c2", size = 44486, upload-time = "2026-01-26T02:45:11.938Z" },
    { url = "https://files.pythonhosted.org/packages/f1/4f/733c48f270565d78b4544f2baddc2fb2a245e5a8640254b12c36ac7ac68e/multidict-6.7.1-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:0e161ddf326db5577c3a4cc2d8648f81456e8a20d40415541587a71620d7a7d1", size = 43219, upload-time = "2026-01-26T02:45:14.346Z" },
    { url = "https://files.pythonhosted.org/packages/24/bb/2c0c2287963f4259c85e8bcbba9182ced8d7fca65c780c38e99e61629d11/multidict-6.7.1-cp314-cp314-manylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:1e3a8bb24342a8201d178c3b4984c26ba81a577c80d4d525727427460a50c22d", size = 245132, upload-time = "2026-01-26T02:45:15.712Z" },
    { url = "https://files.pythonhosted.org/packages/a7/f9/44d4b3064c65079d2467888794dea218d1601898ac50222ab8a9a8094460/multidict-6.7.1-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:97231140a50f5d447d3164f994b86a0bed7cd016e2682f8650d6a9158e14fd31", size = 252420, upload-time = "2026-01-26T02:45:17.293Z" },
    { url = "https://files.pythonhosted.org/packages/8b/13/78f7275e73fa17b24c9a51b0bd9d73ba64bb32d0ed51b02a746eb876abe7/multidict-6.7.1-cp314-cp314-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:6b10359683bd8806a200fd2909e7c8ca3a7b24ec1d8132e483d58e791d881048", size = 233510, upload-time = "2026-01-26T02:45:19.356Z" },
    { url = "https://files.pythonhosted.org/packages/4b/25/8167187f62ae3cbd52da7893f58cb036b47ea3fb67138787c76800158982/multidict-6.7.1-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:283ddac99f7ac25a4acadbf004cb5ae34480bbeb063520f70ce397b28185936 2", size = 264094, upload-time = "2026-01-26T02:45:20.834Z" },
    { url = "https://files.pythonhosted.org/packages/a1/e7/69a3a83b7b030cf283fb06ce074a05a02322359783424d7edf0f15fe5022/multidict-6.7.1-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:538cec1e18c067d0e6103aa9a74f9e832904c957adc260e61cd9d8cf0c3b3d37", size = 260786, upload-time = "2026-01-26T02:45:22.818Z" },
    { url = "https://files.pythonhosted.org/packages/fe/3b/8ec5074bcfc450fe84273713b4b0a0dd47c0249358f5d82eb8104ffe2520/multidict-6.7.1-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:7eee46ccb30ff48a1e35bb818cc90846c6be2b68240e42a78599166722cea709",
```

```
 size = 248483, upload-time = "2026-01-26T02:45:24.368Z" },
    { url = "https://files.pythonhosted.org/packages/48/5a/d5a99e3acbca0e29c5d9cba8f92ceb15dce78bab963b308ae692981e3a5d/multidict-6.7.1-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:fa263a02f4f2dd2d11a7b1bb4362aa7cb1049f84a9235d31adf63f30143469a0", size = 248403, upload-time = "2026-01-26T02:45:25.982Z" },
    { url = "https://files.pythonhosted.org/packages/35/48/e58cd31f6c7d5102f2a4bf89f96b9cf7e00b6c6f3d04ecc44417c00a5a3c/multidict-6.7.1-cp314-cp314-musllinux_1_2_armv7l.whl", hash = "sha256:2e1425e2f99ec5bd36c15a01b690a1a2456209c5deed58f95469ffb46039ccbb", size = 240315, upload-time = "2026-01-26T02:45:27.487Z" },
    { url = "https://files.pythonhosted.org/packages/94/33/1cd210229559cb90b6786c30676bb0c58249ff42f942765f88793b41fdce/multidict-6.7.1-cp314-cp314-musllinux_1_2_i686.whl", hash = "sha256:497394b3239fc6f0e13a78a3e1b61296e72bf1c5f94b4c4eb80b265c37a131cd", size = 245528, upload-time = "2026-01-26T02:45:28.991Z" },
    { url = "https://files.pythonhosted.org/packages/64/f2/6e1107d226278c876c783056b7db43d800bb64c6131cec9c8dfb6903698e/multidict-6.7.1-cp314-cp314-musllinux_1_2_ppc64le.whl", hash = "sha256:233b398c29d3f1b9676b4b6f75c518a06fcb2ea0b925119fb2c1bc35c05e1601", size = 258784, upload-time = "2026-01-26T02:45:30.503Z" },
    { url = "https://files.pythonhosted.org/packages/4d/c1/11f664f14d525e4a1b5327a82d4de61a1db604ab34c6603bb3c2cc63ad34/multidict-6.7.1-cp314-cp314-musllinux_1_2_s390x.whl", hash = "sha256:93b1818e4a6e0930454f0f2af7dfce69307ca03cdcfb3739bf4d91241967b6c1", size = 251980, upload-time = "2026-01-26T02:45:32.603Z" },
    { url = "https://files.pythonhosted.org/packages/e1/9f/75a9ac888121d0c5bbd4ecf4eead45668b1766f6baabfb3b7f66a410e231/multidict-6.7.1-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:f33dc2a3abe9249ea5d8360f969ec7f4142e7ac45ee7014d8f8d5acddf178b7b", size = 243602, upload-time = "2026-01-26T02:45:34.043Z" },
    { url = "https://files.pythonhosted.org/packages/9a/e7/50bf7b004cc8525d80dbbbedfdc7aed3e4c323810890be4413e589074032/multidict-6.7.1-cp314-cp314-win32.whl", hash = "sha256:3ab8b9d8b75aef9df299595d5388b14530839f6422333357af1339443cff777d", size = 40930, upload-time = "2026-01-26T02:45:36.278Z" },
    { url = "https://files.pythonhosted.org/packages/e0/bf/52f25716bbe93745595800f36fb17b73711f14da59ed0bb2eba141bc9f0f/multidict-6.7.1-cp314-cp314-win_amd64.whl", hash = "sha256:5e01429a929600e7dab7b166062d9bb54a5eed752384c7384c968c2afab8f50f", size = 45074, upload-time = "2026-01-26T02:45:37.546Z" },
    { url = "https://files.pythonhosted.org/packages/97/ab/22803b03285fa3a525f48217963da3a65ae40f6a1b6f6cf2768879e208f9/multidict-6.7.1-cp314-cp314-win_arm64.whl", hash = "sha256:4885cb0e817aef5d00a2e8451d4665c1808378dc27c2705f1bf4ef8505c0d2e5", size = 42471, upload-time = "2026-01-26T02:45:38.889Z" },
    { url = "https://files.pythonhosted.org/packages/e0/6d/f9293baa6146ba9507e360ea0292b6422b016907c393e2f63fc40ab7b7b5/multidict-6.7.1-cp314-cp314t-macosx_10_15_universal2.whl", hash = "sha256:0458c978acd8e6ea53c81eefaddbbee9c6c5e591f41b3f5e8e194780fe026581", size = 82401, upload-time = "2026-01-26T02:45:40.254Z" },
    { url = "https://files.pythonhosted.org/packages/7a/68/53b5494738d83558d87c3c71a486504d8373421c3e0dbb6d0db48ad42ee0/multidict-6.7.1-cp314-cp314t-macosx_10_15_x86_64.whl", hash = "sha256:c0abd12629b0af3cf590982c0b413b1e7395cd4ec026f30986818ab95bfaa94a", size = 48143, upload-time = "2026-01-26T02:45:41.635Z" },
    { url = "https://files.pythonhosted.org/packages/37/e8/5284c53310dcdc99ce5d66563f6e5773531a9b9fe9ec7a615e9bc306b05f/multidict-6.7.1-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:14525a5f61d7d0c94b368a42cff4c9a4e7ba2d52e2672a7b23d84dc86fb02b0c", size = 46507, upload-time = "2026-01-26T02:45:42.99Z" },
    { url = "https://files.pythonhosted.org/packages/e4/fc/6800d0e5b3875568b4083ecf5f310dcf91d86d52573160834fb4bfcf5e4f/multidict-6.7.1-cp314-cp314t-manylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:17307b22c217b4cf05033dabefe68255a534d637c6c9b0cc8382718f87be4262", size = 239358, upload-time = "2026-01-26T02:45:44.376Z" },
    { url = "https://files.pythonhosted.org/packages/41/75/4ad0973179361cdf3a113905e6e088173198349131be2b390f9fa4da5fc6/multidict-6.7.1-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:7a7e590ff876a3eaf1c02a4dfe0724b6e69a9e9de6d8f556816f29c496046e59", size = 246884, upload-time = "2026-01-26T02:45:47.167Z" },
    { url = "https://files.pythonhosted.org/packages/c3/9c/095bb28b5da139bd41fb9a5d5caff412584f377914bd8787c2aa98717130/multidict-6.7.1-cp314-cp314t-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:5fa6a95dfee63893d80a34758cd0e0c118a30b8dcb46372bf75106c591b77889", size = 225878, upload-time = "2026-01-26T02:45:48.698Z" },
    { url = "https://files.pythonhosted.org/packages/07/d0/c0a72000243756e8f5a277b6b514fa005f2c73d481b7d9e47cd4568aa2e4/multidict-6.7.1-cp314-cp314t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:a0543217a6a017692aa6ae5cc39adb75e587af0f3a82288b1492eb73dd6cc2a4", size = 253542, upload-time = "2026-01-26T02:45:50.164Z" },
    { url = "https://files.pythonhosted.org/packages/c0/6b/f69da15289e384ecf2a68837ec8b5ad8c33e973aa18b266f50fe55f24b8c/multidict-6.7.1-cp314-cp314t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:f99fe611c312b3c1c0ace793f92464d8cd263cc3b26b5721950d977b006b6c4d", size = 252403, upload-time = "2026-01-26T02:45:51.779Z" },
    { url = "https://files.pythonhosted.org/packages/a2/76/b9669547afa5a1a25cd93eaca91c0da1c095b06b6d2d8ec25b713588d3a1/multidict-6.7.1-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9004d8386d133b7e6135679424c91b0b854d2d164af6ea3f289f8f2761064609", size = 244889, upload-time = "2026-01-26T02:45:53.27Z" },
    { url = "https://files.pythonhosted.org/packages/7e/a9/a50d2669e506dad33cfc45b5d574a205587b7b8a5f426f2fbb2e90882588/multidict-6.7.1-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:e628ef0e6859ffd8273c69412a2465c4be4a9517d07261b33334b5ec6f3c7489", size = 241982, upload-time = "2026-01-26T02:45:54.919Z" },
    { url = "https://files.pythonhosted.org/packages/c5/bb/1609558ad8b456b4827d3c5a5b775c93b87878fd3117ed3db3423dfbce1b/multidict-6.7.1-cp314-cp314t-musllinux_1_2_armv7l.whl", hash = "sha256:841189848ba629c3552035a6a7f5bf3b02eb304e9fea7492ca220a8eda6b0e5c", size = 232415, upload-time = "2026-01-26T02:45:56.981Z" },
    { url = "https://files.pythonhosted.org/packages/d8/59/6f61039d2aa9261871e03ab9dc058a550d240f25859b05b67fd70f80d4b3/multidict-6.7.1-cp314-cp314t-musllinux_1_2_i686.whl", hash = "sha256:ce1bbd7d780bb5a0da032e095c951f7014d6b0a205f8318308140f1a6aba159e", size = 240337, upload-time = "2026-01-26T02:45:58.698Z" },
    { url = "https://files.pythonhosted.org/packages/a1/29/fdc6a43c203890dc2ae9249971ecd0c41deaedfe00d25cb6564b2edd99eb/multidict-6.7.1-cp314-cp314t-musllinux_1_2_ppc64le.whl", hash = "sha256:b26684587228afed0d50cf804cc71062cc9c1cdf55051c4c6345d372947b268c", size = 248788, upload-time = "2026-01-26T02:46:00.862Z" },
    { url = "https://files.pythonhosted.org/packages/a9/14/a153a06101323e4cf086ecee3faadba52ff71633d471f9685c42e3736163/multidict-6.7.1-cp314-cp314t-musllinux_1_2_s390x.whl", hash = "sha256:9f9af11306994335398293f9958071019e3ab95e9a707dc1383a35613f6abcb9", size = 242842, upload-time = "2026-01-26T02:46:02.824Z" },
    { url = "https://files.pythonhosted.org/packages/41/5f/604ae839e64a4a6efc80db94465348d3b328ee955e37acb24badbcd24d83/multidict-6.7.1-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:b4938326284c4f1224178a560987b6cf8b4d38458b113d9b8c1db1a836e640a2", size = 240237, upload-time = "2026-01-26T02:46:05.898Z" },
    { url = "https://files.pythonhosted.org/packages/5f/60/c3a5187bf66f6fb546ff4ab8fb5a077cbdd832d7b1908d4365c7f74a1917/multidict-6.7.1-cp314-cp314t-win32.whl", hash = "sha256:98655c737850c064a65e006a3df7c997cd3b220be4ec8fe26215760b9697d4d7", size = 48008, upload-time = "2026-01-26T02:46:07.468Z" },
    { url = "https://files.pythonhosted.org/packages/0c/f7/addf1087b860ac60e6f382240f64fb99f8bfb532bb06f7c542b83c29ca61/multidict-6.7.1-cp314-cp314t-win_amd64.whl", hash = "sha256:497bde6223c212ba11d462853cfa4f0ae6ef97465033e7dc9940cdb3ab5b48e5", size = 53542, upload-time = "2026-01-26T02:46:08.809Z" },
    { url = "https://files.pythonhosted.org/packages/4c/81/4629d0aa32302ef7b2ec65c75a728cc5ff4fa410c50096174c1632e70b3e/multidict-6.7.1-cp314-cp314t-win_arm64.whl", hash = "sha256:2bbd113e0d4af5db41d5ebfe9ccaff89de2120578164f86a5d17d5a576d1e5b2", size = 44719, upload-time = "2026-01-26T02:46:11.146Z" },
    { url = "https://files.pythonhosted.org/packages/81/08/7036c080d7117f28a4af526d794aab6a84463126db031b007717c1a6676e/multidict-6.7.1-py3-none-any.whl", hash = "sha256:55d97cc6dae627efa6a6e548885712d4864b81110ac76fa4e534c03819fa4a56", size = 12319, upload-time = "2026-01-26T02:46:44.004Z" },
]

[[package]]
name = "mypy"
version = "2.3.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "ast-serialize" },
    { name = "librt", marker = "platform_python_implementation != 'PyPy'" },
    { name = "mypy-extensions" },
    { name = "pathspec" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/12/af/4e516a05d3ca2eb9283e9ec45b2c02225c1514dd6da49fd3c9eaa6639370/mypy-2.3.0.tar.gz", hash = "sha256:465965d41cd9a2726694e983e8ce7113259327bec798115d1e1dfa2a52fb666e", size = 3988104, upload-time = "2026-07-13T11:34:53.387Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/dc/94/0e7e592619e2133596a47cdd642534b0456545c218430bd3b9d8fefdd1b1/mypy-2.3.0-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:2d53fc67b9d28a43c6199077f49fea0f05839e36cf6158500331c9549225e5a5", size = 15026523, upload-time = "2026-07-13T11:34:49.206Z" },
    { url = "https://files.pythonhosted.org/packages/f6/d2/1e1731df090a857df2807177a4626863e5ac0f0256513c35780efe53986f/mypy-2.3.0-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:fbc00cee7bdbb9291979ddc9d08034a29dfcda4932628c9bbc28c1edd589df0c", size = 14032189, upload-time = "2026-07-13T11:33:57.168Z" },
    { url = "https://files.pythonhosted.org/packages/44/95/cab921f4a806e171f34113e6181dd23c55358ccf6a80741269ef594a410e/mypy-2.3.0-cp312-cp312-manylin
```

```
ux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:04e617030eca5221909c8b7d8d7fd1c637948199aa2100b2ad9813feb07e1491", s
ize = 14198696, upload-time = "2026-07-13T11:32:12.767Z" },
    { url = "https://files.pythonhosted.org/packages/66/80/e6d008bb19fe446e3662d85e0e2717bf9f2d611a2164fb29d6e067dbf46c/mypy-2.3.0-cp312-cp312-manylin
ux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:56c184d2c20ca6b6378d58d1960270a767f41f5e44acbbd27f05effef4f4e1d7", size
 = 15286904, upload-time = "2026-07-13T11:34:27.594Z" },
    { url = "https://files.pythonhosted.org/packages/db/83/94397c9293608a364aa03e8084fb34ede4ae976a260384b9b52929308135/mypy-2.3.0-cp312-cp312-musllin
ux_1_2_x86_64.whl", hash = "sha256:3961a4a34b05f7c74b0f05aa51fbfe99a2d1e126038df40318d15c8f558b7ef3", size = 15528342, upload-time = "2026-07-13T11:34
:07.819Z" },
    { url = "https://files.pythonhosted.org/packages/cf/96/d8b37d819adec6cfccfb1fd3afc1735d94717ddeafb45536db9c6943e09b/mypy-2.3.0-cp312-cp312-win_amd
64.whl", hash = "sha256:b1942b9314d4c784b8ea1dbab4972603290e5dd5630f06675f13aec97526bc4c", size = 11218346, upload-time = "2026-07-13T11:28:27.745Z" }
,
    { url = "https://files.pythonhosted.org/packages/2b/cd/cd9f725b19b19e5b530a154cf9bcf9e94279c5d55b3c34fb42b3aa48ea1b/mypy-2.3.0-cp312-cp312-win_arm
64.whl", hash = "sha256:be51653d7669d7d7955d613b8d0bb57d5b652eaf71a873ddf65ac87254dd2595", size = 10204525, upload-time = "2026-07-13T11:31:02.552Z" }
,
    { url = "https://files.pythonhosted.org/packages/6e/ae/f7d056eb0294586a572d0d0d89580ec633c064db520f11d37d5a2fb833bd/mypy-2.3.0-cp313-cp313-macosx_
10_13_x86_64.whl", hash = "sha256:91ad22a52ae2c7e621c2f67c94d5a17f66b3209a4cff5cf8a573579835c69e97", size = 14947298, upload-time = "2026-07-13T11:27:
47.734Z" },
    { url = "https://files.pythonhosted.org/packages/32/d5/db3e7af01e7844d21662c6ddc1f7825ec7cb4053f0391ac02faf3638396f/mypy-2.3.0-cp313-cp313-macosx_
11_0_arm64.whl", hash = "sha256:99ac767cc5d3b64c8d0ae226ead10c96694f94e4e7da1668642225dcd4e75aac", size = 13950768, upload-time = "2026-07-13T11:27:57
.726Z" },
    { url = "https://files.pythonhosted.org/packages/d9/fb/43c031f0190513d1ec248ed037eceb742ddd2a4d74bbf406658a28173837/mypy-2.3.0-cp313-cp313-manylin
ux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:de6d2c484742a4d7b0ed6d07b143375624d3b899c5749c7b3c947f56261f48a6", s
ize = 14151586, upload-time = "2026-07-13T11:29:18.615Z" },
    { url = "https://files.pythonhosted.org/packages/ec/c3/f8b2ffc60883084da91be51af58e88a7ffd4ff9795acb7d902ff88d31eb1/mypy-2.3.0-cp313-cp313-manylin
ux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:7da939dd335cfd2ad788bdfd081c9f4e47634ab995e5a45eb15fd1e5bc052f8b", size
 = 15227411, upload-time = "2026-07-13T11:30:29.904Z" },
    { url = "https://files.pythonhosted.org/packages/83/2e/16b917fc7adcf03f1aadddfc93aab804ffb234b1ab09c0ffd6d92a5d34a2/mypy-2.3.0-cp313-cp313-musllin
ux_1_2_x86_64.whl", hash = "sha256:7247eb2824f996722a949530183394921ca71deb9680052a338cf53cff7925c2", size = 15478790, upload-time = "2026-07-13T11:33
:14.686Z" },
    { url = "https://files.pythonhosted.org/packages/c0/88/aaa65a93c73d0cdae7e42f8adb302bf6885bb281302084f99d0290a35347/mypy-2.3.0-cp313-cp313-win_amd
64.whl", hash = "sha256:75b0984bb3cbd76bb5c9291a8671f7ae66ca3b51c7584c358fc2e923259f0757", size = 11234919, upload-time = "2026-07-13T11:33:39.28Z" },
    { url = "https://files.pythonhosted.org/packages/35/19/b40de63f1a80e63bc2d40f0679a6a8dbd34e95176c8122119bdf406aa552/mypy-2.3.0-cp313-cp313-win_arm
64.whl", hash = "sha256:d78fcf900b59cb7e82cb7e3a235e31b462d9333d92285bd1e4952d355b8ffba1", size = 10201510, upload-time = "2026-07-13T11:31:52.619Z" }
,
    { url = "https://files.pythonhosted.org/packages/a4/58/fa0ae047da911f540284009b4f44b96fe09d83c076d7c103e9d645f46303/mypy-2.3.0-cp314-cp314-macosx_
10_15_x86_64.whl", hash = "sha256:ea317b060ce83e26050f8f9e4d7d6bf44ed7597c8ff9990bccffbb9d1d8522db", size = 14941909, upload-time = "2026-07-13T11:32:
34.332Z" },
    { url = "https://files.pythonhosted.org/packages/15/14/2ba1d61452d7c2a7fe12741e8d374e52b183476b07aa7f9e2a0d02b0720a/mypy-2.3.0-cp314-cp314-macosx_
11_0_arm64.whl", hash = "sha256:094af99f92638aa92852326188b85a89e50f4a472f44827c03362228482f0762", size = 13967581, upload-time = "2026-07-13T11:30:00
.587Z" },
    { url = "https://files.pythonhosted.org/packages/ed/5a/483fb9e5ffbbb1a28dccc7b0a13d141b17ac769b6c9f488c0a0c63698962/mypy-2.3.0-cp314-cp314-manylin
ux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:de121747278144fc9ae7caa2e978cf5df12aebc82933182f5b3b86081a30baef", s
ize = 14168807, upload-time = "2026-07-13T11:28:48.6Z" },
    { url = "https://files.pythonhosted.org/packages/ae/77/70d7a10732063beb74ad713682cf871e88f5c5fa39bfc8beff8a524bf9cb/mypy-2.3.0-cp314-cp314-manylin
ux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:37fa4de896a84e2dc9200d91e614c22563b43d1a266789d4bbac7b22ebe6192b", size
 = 15200144, upload-time = "2026-07-13T11:31:25.283Z" },
    { url = "https://files.pythonhosted.org/packages/56/72/766218ac783be4fdfcd699b90037b63017348a3e86fb2c1fbfb18302637d/mypy-2.3.0-cp314-cp314-musllin
ux_1_2_x86_64.whl", hash = "sha256:f1b3a98dfd21058bc759bb3337d5d1f61d0fdf9f3cf9c00f4291790fb5427bff", size = 15460389, upload-time = "2026-07-13T11:29
:29.077Z" },
    { url = "https://files.pythonhosted.org/packages/38/4e/8a9db7411ecb8ec0cb1fd05dba432f28bafffcd38b4e887714a4a0506689/mypy-2.3.0-cp314-cp314-pyemscr
ipten_2026_0_wasm32.whl", hash = "sha256:944c665d984157cb96a679dfb7a4a81dd1d36b24b9c284b699514e6e626b82d4", size = 7753664, upload-time = "2026-07-13T
11:29:08.147Z" },
    { url = "https://files.pythonhosted.org/packages/65/4c/c3f8bfd6ed0e5e38b5a244403b27f821d433443df5a15a278417c10a3a3c/mypy-2.3.0-cp314-cp314-win_amd
64.whl", hash = "sha256:4359424140d985192c778c1ce2c114a10c1ca58a381ed79cfa70d37df94b299f", size = 11417237, upload-time = "2026-07-13T11:33:47.467Z" }
,
    { url = "https://files.pythonhosted.org/packages/3c/00/89a32eaf5ccf174bc4f90db0eaea5d70636c01b8d49f384bdab2e8834390/mypy-2.3.0-cp314-cp314-win_arm
64.whl", hash = "sha256:3dd0bed92c4bdec57c42505b96416fb9e6a5aa7be84d2809bcd5f2ecec2860d7", size = 10389252, upload-time = "2026-07-13T11:31:43.81Z" },
    { url = "https://files.pythonhosted.org/packages/31/56/104f93d69aa9f339b6b9d3b0a7faa699b8b466c942cf3ae86cc2a2ec0915/mypy-2.3.0-cp314-cp314t-macosx
_10_15_x86_64.whl", hash = "sha256:691fdc37132b1ae628d834f672e74de83462d9fb4aff621835767fb43a8dd373", size = 16385495, upload-time = "2026-07-13T11:29
:49.818Z" },
    { url = "https://files.pythonhosted.org/packages/d2/03/f1d2123313f55efafdd27706960f43a771c62f1b68426c76043f3ab9ebf3/mypy-2.3.0-cp314-cp314t-macosx
_11_0_arm64.whl", hash = "sha256:aec15d465d477558fd842757b487849007311cf3897849cdda0e3162ac0ac556", size = 15098155, upload-time = "2026-07-13T11:30:4
0.301Z" },
    { url = "https://files.pythonhosted.org/packages/e5/5d/d5f9200399b445e81726c4f23becee33f233aee81c72680b1ef3a258b641/mypy-2.3.0-cp314-cp314t-manyli
nux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:b352b7e49f5e6576009e8df730e1ff4f915cb565b851b396d2ffe2f5a6f5da88",
size = 15514155, upload-time = "2026-07-13T11:34:38.569Z" },
    { url = "https://files.pythonhosted.org/packages/cd/ce/69977c555f08faa3190cfde44189b89dbd56861b1ab97aa18fc5f3a2e4a3/mypy-2.3.0-cp314-cp314t-manyli
nux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:1c6c6bf687b17f90dbfcad95b960d32eaa0154c00da45f03ab50bf8952e047fe", siz
e = 16766351, upload-time = "2026-07-13T11:33:29.195Z" },
    { url = "https://files.pythonhosted.org/packages/bc/92/6648b6caa3ab9e00f9ac0c2a78307805f873dd48139b24a6f6f7c3667bbf/mypy-2.3.0-cp314-cp314t-muslli
nux_1_2_x86_64.whl", hash = "sha256:f4ed18f111bfe2d599bca7468e7f9251042c1c2118f762c8de2766a56d773c60", size = 17043490, upload-time = "2026-07-13T11:3
0:53.927Z" },
    { url = "https://files.pythonhosted.org/packages/7c/ab/0dc91d80f3f016634c68d451f294a97320fe903a9b6f90b9e57b3f7f1717/mypy-2.3.0-cp314-cp314t-win_am
d64.whl", hash = "sha256:0b025a93cffb9781d231f232be07a17912f35f10a313c24f301c81e842870654", size = 12146869, upload-time = "2026-07-13T11:29:38.874Z"
},
    { url = "https://files.pythonhosted.org/packages/85/b5/4c964d02634ba81f4d1c84838e5c5b18ab06d13ed568960f5d6318495ccc/mypy-2.3.0-cp314-cp314t-win_ar
m64.whl", hash = "sha256:adebc76aab4f3495a88b41d48aa4aff0c03f2822501da76625afcca5975f19e5", size = 10965113, upload-time = "2026-07-13T11:28:07.056Z"
},
    { url = "https://files.pythonhosted.org/packages/2c/fa/fdc54fe583ba3cafbcedfb70eeeaf03849f75b1827a07096c7bd996f582d/mypy-2.3.0-py3-none-any.whl",
hash = "sha256:6b1cdb579446b60432432b2b2403a6201b4b475a004d7f488511c9ba177c9e88", size = 2753292, upload-time = "2026-07-13T11:33:18.48Z" },
]

[[package]]
name = "mypy-extensions"
version = "1.1.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a2/6e/371856a3fb9d31ca8dac321cda606860fa4548858c0cc45d9d1d4ca2628b/mypy_extensions-1.1.0.tar.
gz", hash = "sha256:52e68efc3284861e772bbcd66823fde5ae21fd2fdb51c62a211403730b916558", size = 6343, upload-time = "2025-04-22T14:54:24.164Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/79/7b/2c79738432f5c924bef5071f933bcc9efd0473bac3b4aa584a6f7c1c8df8/mypy_extensions-1.1.0-py3-none
-any.whl", hash = "sha256:1be4cccdb0f2482337c4743e60421de3a356cd97508abadd57d47403e94f5505", size = 4963, upload-time = "2025-04-22T14:54:22.983Z" },
]

[[package]]
name = "nest-asyncio"
version = "1.6.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/83/f8/51569ac65d696c8ecbee95938f89d4abf00f47d58d48f6fbabfe8f0baefe/nest_asyncio-1.6.0.tar.gz"
, hash = "sha256:6f172d5449aca15afd6c646851f4e31e02c598d553a667e38cafa997cfec55fe", size = 7418, upload-time = "2024-01-21T14:25:19.227Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a0/c4/c2971a3ba4c6103a3d10c4b0f24f461ddc027f0f09763220cf35ca1401b3/nest_asyncio-1.6.0-py3-none-an
y.whl", hash = "sha256:87af6efd6b5e897c81050477ef65c62e2b2f35d51703cae01aff2905b1852e1c", size = 5195, upload-time = "2024-01-21T14:25:17.223Z" },
]

[[package]]
```

```
name = "networkx"
version = "3.6.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/6a/51/63fe664f3908c97be9d2e4f1158eb633317598cfa6e1fc14af5383f17512/networkx-3.6.1.tar.gz", hash = "sha256:26b7c357accc0c8cde558ad486283728b65b6a95d85ee1cd66bafab4c8168509", size = 2517025, upload-time = "2025-12-08T17:02:39.908Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/9e/c9/b2622292ea83fbb4ec318f5b9ab867d0a28ab43c5717bb85b0a5f6b3b0a4/networkx-3.6.1-py3-none-any.whl", hash = "sha256:d47fbf302e7d9cbbb9e2555a0d267983d2aa476bac30e90dfbe5669bd57f3762", size = 2068504, upload-time = "2025-12-08T17:02:38.159Z" },
]

[[package]]
name = "nltk"
version = "3.10.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "click" },
    { name = "defusedxml" },
    { name = "joblib" },
    { name = "regex" },
    { name = "tqdm" },
]
sdist = { url = "https://files.pythonhosted.org/packages/72/16/24d639531e73cbc6884fb251d116dfe469df9c595e0dcf24668c54d0e8d3/nltk-3.10.2.tar.gz", hash = "sha256:fcfd80fb77931868cea8357573c79838b8abc609942ef9914d1c9f6070d4645c", size = 3101716, upload-time = "2026-08-05T09:56:20.657Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/1f/2b/bf677eb32ca6684b270c0d19ab133c2271e9f3375997e5a8dd2b08e3152d/nltk-3.10.2-py3-none-any.whl", hash = "sha256:2c7ccacb765c5e26b0cb60fb1b57080af522c6924d12a714a243305ba3637412", size = 1725815, upload-time = "2026-08-05T09:56:09.657Z" },
]

[[package]]
name = "numpy"
version = "2.5.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/22/fd/89965aa4ac08c74998539fcbf24fa3540f3e15237fbeb6bcf9c908f4aade/numpy-2.5.1.tar.gz", hash = "sha256:a48a113e6afea91f5608793bafa7ef2ad481fefbda87ec5069f483de61cb9fa3", size = 20755553, upload-time = "2026-07-04T17:08:00.933Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/62/7b/14687aa674250e5e546f616f486b0d56d3631cd5b2415739141ce40bdcea/numpy-2.5.1-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:2c889b56fe48b1018f764b0eec8df59ab654e9148aa91faa12596043500de277", size = 16801574, upload-time = "2026-07-04T17:06:12.423Z" },
    { url = "https://files.pythonhosted.org/packages/e1/19/cc5bb2a3f2913d27d6dbb2c78d25921fabaedc6741d4a5a615a11f3c5bf3/numpy-2.5.1-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:ab451b59c5643c570974c43aef780703ef1d3b4965d2be07afd530615a9358d1", size = 11772250, upload-time = "2026-07-04T17:06:15.726Z" },
    { url = "https://files.pythonhosted.org/packages/42/77/fdf34a71dd30f54979b18603bee915e0aaf825b07afe79acd60b04b691e2/numpy-2.5.1-cp312-cp312-macosx_14_0_arm64.whl", hash = "sha256:78798bd5b9ad744056af8efa90e3b9ddaa53272a0848a483084a1cc0a13b2dc0", size = 5331516, upload-time = "2026-07-04T17:06:17.913Z" },
    { url = "https://files.pythonhosted.org/packages/ce/e2/eb7efa015b4cce41e2517bf182a7fce0d7d5b9d9ed76a29bfa0f4fe4505c/numpy-2.5.1-cp312-cp312-macosx_14_0_x86_64.whl", hash = "sha256:2ae0ca40bcb22d6ba59c1dfd5446f49940b0f2d821fde133f10dda11f816b84e", size = 6664863, upload-time = "2026-07-04T17:06:20.02Z" },
    { url = "https://files.pythonhosted.org/packages/a9/4b/a2b32dd94ee9ffbeecb28152240042a3949db33b1c834d44090b80e1b3b8/numpy-2.5.1-cp312-cp312-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:61ac47e772e6b8ea489e1d2f441a34c5c3ac17327e7ce294cbdf535795ad4e75", size = 15167977, uploaded-time = "2026-07-04T17:06:21.621Z" },
    { url = "https://files.pythonhosted.org/packages/b8/a9/6e73d68500f80773f65f0654ea932019d6694329a0eb0ed0533de38df376/numpy-2.5.1-cp312-cp312-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:59fda5e192b570217ec2580c96f00e9a7e12ef6866a900eb089b62c1a32545ca", size = 16672469, upload-time = "2026-07-04T17:06:24.064Z" },
    { url = "https://files.pythonhosted.org/packages/24/7d/ad3e59015135f5261c95fd4cafeff159c955febd83a99a1d9250c4233815/numpy-2.5.1-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:f7119ebff1a9829e9f431a4f9d28e703023bb6b9fe7c8f724467dbfc27c94ab3", size = 16527531, upload-time = "2026-07-04T17:06:26.69Z" },
    { url = "https://files.pythonhosted.org/packages/83/d0/a39b2fbcde9cb17a1dac678f254b33a6336298af9df338824c685425d5e8/numpy-2.5.1-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:e824c2acf8862052246be5a44c15da1777940c60d010dd2aab897824d9c430f9", size = 18431940, upload-time = "2026-07-04T17:06:29.521Z" },
    { url = "https://files.pythonhosted.org/packages/04/12/cff070947791c1ed425ff76413189adbdc2fbe215eba7ce7fa454a03c7f8/numpy-2.5.1-cp312-cp312-win32.whl", hash = "sha256:08d60c810432eb83360958dea0999ac4cfb94531ea8efcbf0b7f277c2068aeb2", size = 6066764, upload-time = "2026-07-04T17:06:32.571Z" },
    { url = "https://files.pythonhosted.org/packages/65/66/53f31807a48a750f9d748da273bc3fcedd12b27ff1f3e373bfec55ef2dc0/numpy-2.5.1-cp312-cp312-win_amd64.whl", hash = "sha256:f7d60026c0bdb1380e83bfa7a0419c4577ee4b9a08880afcb6dadeb74c649fa2", size = 12430966, upload-time = "2026-07-04T17:06:34.926Z" },
    { url = "https://files.pythonhosted.org/packages/2b/2a/d1a88066b1c14186f5d3c0d18c94f17b064511982bab0578d49ee9d43c29/numpy-2.5.1-cp312-cp312-win_arm64.whl", hash = "sha256:17a25e09640602e10bc8de0e6fa2b3fd68eedd84ba6d7842dc8f32f9ab87bd0b", size = 10350488, upload-time = "2026-07-04T17:06:37.785Z" },
    { url = "https://files.pythonhosted.org/packages/eb/07/ec2a3f0c91761581d4b7104a740791800025983f9a4dc4e73f91a99aeac4/numpy-2.5.1-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:0bfebd8695f9863592fe744be833a258120b14a9f39da255e8aa8fade2c0ddd1", size = 16796419, upload-time = "2026-07-04T17:06:40.37Z" },
    { url = "https://files.pythonhosted.org/packages/ab/ab/ddb499fc4f8780354395face5b65c7fd107bcd6e1d667a5f07d046956f6f/numpy-2.5.1-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:30b44a6b53a7ae63c54c089a8726e5563ed302716c5b7ccc85afade40b0e7ff6", size = 11765832, upload-time = "2026-07-04T17:06:42.768Z" },
    { url = "https://files.pythonhosted.org/packages/88/b3/3c28c558a09fc72100c646dac6d2fce8e834c471b0edca01a29996706117/numpy-2.5.1-cp313-cp313-macosx_14_0_arm64.whl", hash = "sha256:6165343f81b56ef8f514f396989e529b61d9dc709b99421b07e9f3e698e2287d", size = 5325143, upload-time = "2026-07-04T17:06:45.466Z" },
    { url = "https://files.pythonhosted.org/packages/5e/0e/ce19b985bb15c596f4f05954e76cccc77c845083b3b8f938a6c68e523128/numpy-2.5.1-cp313-cp313-macosx_14_0_x86_64.whl", hash = "sha256:4939237038ada79308dda3204ac6462df056b5672b2e25db1149cf873668b3e1", size = 6659749, upload-time = "2026-07-04T17:06:47.288Z" },
    { url = "https://files.pythonhosted.org/packages/2e/20/1ee6614d64332a1bba6411f38e68cb79eec1b2459e20a623777c5c5492a2/numpy-2.5.1-cp313-cp313-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:1c6759f538fb912fc46de0a6b1758ccf7b57bc7c7ebebc23974fdac3de8db0cd", size = 15164716, uploaded-time = "2026-07-04T17:06:49.494Z" },
    { url = "https://files.pythonhosted.org/packages/ed/a7/2bcd3fdbb87804755c35b729bf8709d62025c5f4cfd7d5b2415997097515/numpy-2.5.1-cp313-cp313-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9726558e8db4a5bf7929a70ae50f63abda4daf0efe810e3bfbab95976f75fc1a", size = 16661440, upload-time = "2026-07-04T17:06:52.061Z" },
    { url = "https://files.pythonhosted.org/packages/fc/d7/a41e3310c886fe457d36e670bbf24fae411aca8a7b6ad92a32afd924077c/numpy-2.5.1-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:3935f3b419b244a02732676fa5317a9193cc596a4c0646db07e5b421229ac9f7", size = 16526305, upload-time = "2026-07-04T17:06:54.605Z" },
    { url = "https://files.pythonhosted.org/packages/53/75/4333a9a707c1edd3a4e1a0c58eca52c0f31e55089fa80db02b5565b24df7/numpy-2.5.1-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:dc932a65ded7ce9013d120845a2514dcccb1a67bfc8deb8d37633762951904a6", size = 18423008, upload-time = "2026-07-04T17:06:57.54Z" },
    { url = "https://files.pythonhosted.org/packages/ee/90/e314a32b1c11a2ffe818ddad3a57b50b4b6e1b6c487192eb50cdef0415d0/numpy-2.5.1-cp313-cp313-win32.whl", hash = "sha256:4b4ff1608417eb7a59da7b967bbb798cacfe071d2caf526a24281cd562072ed9", size = 6063885, upload-time = "2026-07-04T17:07:00.14Z" },
    { url = "https://files.pythonhosted.org/packages/10/70/800b3fca480af32df9e8ea9f3d4a0c8feb4b32d7f195d174eabbda4829ad/numpy-2.5.1-cp313-cp313-win_amd64.whl", hash = "sha256:6c3fe51bc6a16453d452997053454f309e8e0ed7b42d6b361ce4ac8c32913d74", size = 12425674, upload-time = "2026-07-04T17:07:02.387Z" },
    { url = "https://files.pythonhosted.org/packages/8b/0b/196350c122f50f6ca56846f2d71efd5e0d24b7b2e07355e019b2e2c7a11e/numpy-2.5.1-cp313-cp313-win_arm64.whl", hash = "sha256:f7feb014281029e628ba2d5a007407443b06e418b6fe451d1e2adcbc8eba0107", size = 10350256, upload-time = "2026-07-04T17:07:04.878Z" },
    { url = "https://files.pythonhosted.org/packages/db/f4/731b6085a83faf6ca843394cbd5e217280c214399f7e8b21b9f552af0ae2/numpy-2.5.1-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:7c786fe9a5bbe360022e584c5a34cf6b54265c71bd7ec8ac3d8fec38968071f8", size = 16795063, upload-time = "2026-07-04T17:07:07.374Z" },
    { url = "https://files.pythonhosted.org/packages/bf/64/0e215f2048dd11a55bb989ed41b3585ef57452404e638d703a211a3e4157/numpy-2.5.1-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:32985c896d897419ef8da6917872d80b78ad0ea26d85b23245c7366ffde76d75", size = 11776652, upload-time = "2026-07-04T17:07:09.907Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/b5/59/2b844c7a6e9deff69b404a66221e1542937734f65d5e6e39411876053862/numpy-2.5.1-cp314-cp314-macosx_14_0_arm64.whl", hash = "sha256:efd736408cc97c79b9e6917338dfc8f06013b2274f992e96b1d9a81a71e2a2c2", size = 5335944, upload-time = "2026-07-04T17:07:12.227Z" },
    { url = "https://files.pythonhosted.org/packages/86/51/9bf7cb2cabcebc9e017e4ec7e6322b378317a542c08b4cb68479c1efc716/numpy-2.5.1-cp314-cp314-macosx_14_0_x86_64.whl", hash = "sha256:ab84dc6b074fa881cae55bea94cc4f68e285181ba7f32497bf7dee6b1496165b", size = 6656266, upload-time = "2026-07-04T17:07:14.368Z" },
    { url = "https://files.pythonhosted.org/packages/83/3e/fb7615b211b82a32f44d5180a6d421b61f84d4fadd578b48ba4ac34e189f/numpy-2.5.1-cp314-cp314-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:caf3e317d33d60c37986b452613f4ab51246d0691350c03d0cb4a898627f4a95", size = 15179720, uploaded-time = "2026-07-04T17:07:16.272Z" },
    { url = "https://files.pythonhosted.org/packages/41/5f/0f992cb24560673496c5d68de61913b57166ce530ffda07c1f280e0cc464/numpy-2.5.1-cp314-cp314-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:54ad769f17bc2d833b620851989f62054fb9ab93c969d9e1dc3c8e3d56beea21", size = 16664835, upload-time = "2026-07-04T17:07:19.021Z" },
    { url = "https://files.pythonhosted.org/packages/a2/2f/97d6475ee91afe2587797d09446f9d3e475ad4cb681662d824809327b75a/numpy-2.5.1-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:c12afb53450fa976d4c681c50a7423729a4c51c0465ed9f32b8a9cabbc472373", size = 16539135, upload-time = "2026-07-04T17:07:22.015Z" },
    { url = "https://files.pythonhosted.org/packages/c4/5b/4db81e4ba0be7e2776b1de68c82aa862c7f8ec27e1b4927d4ae075e20678/numpy-2.5.1-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:e8c11c405efc5ff6816d5983c96cdfa215bab3428961243af3ff59b228490438", size = 18426684, upload-time = "2026-07-04T17:07:24.941Z" },
    { url = "https://files.pythonhosted.org/packages/1f/64/c0ba2d90724d450279a7df8f32057241070250a26a7e2b5337d77347f481/numpy-2.5.1-cp314-cp314-win32.whl", hash = "sha256:f2479a47f8d5932d1718168a681ad6e536a9df484c83cfcf9de365e164537ace", size = 6116103, upload-time = "2026-07-04T17:07:27.622Z" },
    { url = "https://files.pythonhosted.org/packages/c1/1a/837f9ed7405adcd7a40538792eb169eddd8fa5630c16a1ef49dae71a30f4/numpy-2.5.1-cp314-cp314-win_amd64.whl", hash = "sha256:24d0eb82c0541d3415a33425db64ae439dffccd7b4dbcb30e7c35120205c506a", size = 12562177, upload-time = "2026-07-04T17:07:29.887Z" },
    { url = "https://files.pythonhosted.org/packages/22/ed/49707938b6dd0a78a9178dd93227dc89e4c11af47f5c798d70366e8d0483/numpy-2.5.1-cp314-cp314-win_arm64.whl", hash = "sha256:5a4c988b38d261deeeaad9954e3deb091ad905c94e8bb6708654ef1d97f286b0", size = 10627739, upload-time = "2026-07-04T17:07:32.568Z" },
    { url = "https://files.pythonhosted.org/packages/a6/c7/bb4b882cfe7f299cbc8b66e42e7dd78cf9d14e40f9469fc5e3db7e15b3bd/numpy-2.5.1-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:a33276be12fa045805f477f22482088b66bb758ffbe89a9d21457de863a32e22", size = 11894709, upload-time = "2026-07-04T17:07:34.941Z" },
    { url = "https://files.pythonhosted.org/packages/40/3f/5af7f4a7f6224aef48017aa82bb6174c7a659d724be0c75017b7e64a55b4/numpy-2.5.1-cp314-cp314t-macosx_14_0_arm64.whl", hash = "sha256:f089d7b00756190aacf1f5d34bdf38c3c430ac82b4f868f8cede73380460fce7", size = 5453810, upload-time = "2026-07-04T17:07:37.495Z" },
    { url = "https://files.pythonhosted.org/packages/20/c9/3474309bc94d634d3f9c3eddf03250ecb8c22cd948ef16fef69a77cc5d7b/numpy-2.5.1-cp314-cp314t-macosx_14_0_x86_64.whl", hash = "sha256:09e9bfd8d2cf479c7d174804fb3811c53a8e9f20a37444008606b57d6b7a826d", size = 6761189, upload-time = "2026-07-04T17:07:39.563Z" },
    { url = "https://files.pythonhosted.org/packages/90/8a/558ae39fdd55d7e7f7fef9a84a6e964ac6b23edbd2a07e52bb084500507d/numpy-2.5.1-cp314-cp314t-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:e68d8dd1e7eba712948f2053a29ec86917bc70ba1358df869d9f06649ef9cf09", size = 15225039, upload-time = "2026-07-04T17:07:41.682Z" },
    { url = "https://files.pythonhosted.org/packages/63/27/ca7392b2d030277bdf0273e7d23255b3ee57d57a7c170a6f4fb3981e1e5d/numpy-2.5.1-cp314-cp314t-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:99d5095fa265a0c4152e7bb12759e14381ef5496152f1ce58f44bdf55c44beb4", size = 16701306, upload-time = "2026-07-04T17:07:44.611Z" },
    { url = "https://files.pythonhosted.org/packages/02/42/03d53ae7996c44d4374a8262e9dc41671fd56cbb98f7d47ef85cf5da4c6b/numpy-2.5.1-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:ab87a91b3cc3382b8956095bd8f95e00cf679bb81554339be1a2ba404a1473c1", size = 16589955, upload-time = "2026-07-04T17:07:47.694Z" },
    { url = "https://files.pythonhosted.org/packages/7b/15/6c1784ae469640e65db111e9a34b3d0f14d91e8a38b9ce34810ced370dbb/numpy-2.5.1-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:224ca51130ef7da85bea2191625181cb4f337f9cb64b471f10c1a12aa8b60077", size = 18464252, upload-time = "2026-07-04T17:07:50.684Z" },
    { url = "https://files.pythonhosted.org/packages/94/a8/f98e50356cf167df656c526c2dfeec2d7dde182f2a3da4b458a5938e2776/numpy-2.5.1-cp314-cp314t-win32.whl", hash = "sha256:6eab239876581b2b3c5a242281b6007bbdbcd1c7085d7709bb57c5929b11e6bf", size = 6263298, upload-time = "2026-07-04T17:07:53.445Z" },
    { url = "https://files.pythonhosted.org/packages/72/ac/96ae880cdecad0b3275d9359fcec72667b49a4863c9f12942e43679dda02/numpy-2.5.1-cp314-cp314t-win_amd64.whl", hash = "sha256:83ce9c80d5b521b0d77ddcbe5447c218d247929b6cc056ca5351342accfff0af", size = 12748623, upload-time = "2026-07-04T17:07:55.384Z" },
    { url = "https://files.pythonhosted.org/packages/a1/5a/4d2b1601df3602dba7a14f3348ba9bfe94a18adb428e693df6154c293831/numpy-2.5.1-cp314-cp314t-win_arm64.whl", hash = "sha256:5a6db61f9aaa57e369905c67d852045d3c4f7126405b29d09b19dec118e9c9cb", size = 10697674, upload-time = "2026-07-04T17:07:58.506Z" },
]

[[package]]
name = "openai"
version = "2.53.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "distro" },
    { name = "httpx" },
    { name = "jiter" },
    { name = "pydantic" },
    { name = "sniffio" },
    { name = "tqdm" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/ef/cf/36e3e7235fdf6d125c052acc0970924611b17a20a4fe580596faf4566a65/openai-2.53.0.tar.gz", hash = "sha256:baf5802ad08980e1d9d561e1b996e800c8bcd14af5847c6d0e7a5cc59e4d4116", size = 1099435, upload-time = "2026-08-03T21:42:01.664Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/78/0f/cc6afea3542a5142c5d8fc8211c5e059a8375105d004a41dfa2c7948dbb0/openai-2.53.0-py3-none-any.whl", hash = "sha256:c694ffc747a3c4d1663ef2b07b811315a476164ee5efa3a993967349ebca7618", size = 1659829, upload-time = "2026-08-03T21:41:59.581Z" },
]

[[package]]
name = "opentelemetry-api"
version = "1.44.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/ee/8b/aa9e2d8b8dfa7c946f7dec5d1f8f6ba8eca062f43509a06bdb5ce93d26c0/opentelemetry_api-1.44.0.tar.gz", hash = "sha256:67647e5e9566edcf421166fdf022b3537f818635daa852b289e34604dc6fb33a", size = 72406, upload-time = "2026-07-16T15:25:32.678Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/ca/6f/a04e900f465ff3221ccc395522503e2d10e79fa21f2723c8e177aae1e0d1/opentelemetry_api-1.44.0-py3-none-any.whl", hash = "sha256:94b98c893a91b88657eaac1e3ba89618cdb85be6918196705354f34728b2cdef", size = 60018, upload-time = "2026-07-16T15:25:11.657Z" },
]

[[package]]
name = "opentelemetry-exporter-otlp-proto-common"
version = "1.44.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "opentelemetry-proto" },
]
sdist = { url = "https://files.pythonhosted.org/packages/61/09/4d717852c1cf3f854b76c7110a5d00883bc3c99288b9b0dbcbeb9e306eb6/opentelemetry_exporter_otlp_proto_common-1.44.0.tar.gz", hash = "sha256:dc87a5a5bc58f149a56d1547e4691588fa12994cdc3bc039a694ccb3375862ac", size = 20202, upload-time = "2026-07-16T15:25:37.658Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/5e/71/65fd9d54c10b860f87c045ccee1264cab7011268895d3528818a29c1172a/opentelemetry_exporter_otlp_pr
```

```
oto_common-1.44.0-py3-none-any.whl", hash = "sha256:9a9fe61bba73d802904bc989f1d6b4a7b1ee40f06c40e98d6f85af65aaebb694", size = 17045, upload-time = "2026-07-16T15:25:18.201Z" },
]

[[package]]
name = "opentelemetry-exporter-otlp-proto-http"
version = "1.44.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "googleapis-common-protos" },
    { name = "opentelemetry-api" },
    { name = "opentelemetry-exporter-otlp-proto-common" },
    { name = "opentelemetry-proto" },
    { name = "opentelemetry-sdk" },
    { name = "requests" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/1a/87/95e2a5aaa795b4e2260d74e16df2d5541deb2ea9de010bcd615f4dee2654/opentelemetry_exporter_otlp_proto_http-1.44.0.tar.gz", hash = "sha256:c633d7270ad6b57cd4cfbe8b0007a9e2e7c0cb50bd6c50fe2a7b245f721a09d8", size = 25806, upload-time = "2026-07-16T15:25:39.162Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/cd/d0/fdeb1a98d8d3a6205f5f297c51b4a9bfe65126ab60339669bbe3dd54c2e2/opentelemetry_exporter_otlp_proto_http-1.44.0-py3-none-any.whl", hash = "sha256:838592fce774c1c8bb7b9a0a7facbfa82e17be5a8a4e94cef10cb84ae026bae3", size = 21850, upload-time = "2026-07-16T15:25:20.006Z" },
]

[[package]]
name = "opentelemetry-instrumentation"
version = "0.65b0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "opentelemetry-api" },
    { name = "opentelemetry-semantic-conventions" },
    { name = "packaging" },
    { name = "wrapt" },
]
sdist = { url = "https://files.pythonhosted.org/packages/13/91/3c58961cb0360cd60509064734f0be4275383c8681d73c580a40ca83ddce/opentelemetry_instrumentation-0.65b0.tar.gz", hash = "sha256:071d9d9eced9bd6460444ec3b0c77229870ed05a881c22c84fdede58e4eed09b", size = 42689, upload-time = "2026-07-16T15:25:50.275Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/40/7b/85eab1215f72adf0e68d3dc4a679b9bff993fa679ff34cd8dd378e2659fd/opentelemetry_instrumentation-0.65b0-py3-none-any.whl", hash = "sha256:ea967a72b9939b5fcfdad572753b4306c59dcb99e3f382d95dae04286805e137", size = 36717, upload-time = "2026-07-16T15:24:51.424Z" },
]

[[package]]
name = "opentelemetry-instrumentation-asgi"
version = "0.65b0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "asgiref" },
    { name = "opentelemetry-api" },
    { name = "opentelemetry-instrumentation" },
    { name = "opentelemetry-semantic-conventions" },
    { name = "opentelemetry-util-http" },
]
sdist = { url = "https://files.pythonhosted.org/packages/17/83/8e8e83b7ac285281687c7be2fd305213ccccbb8c0a2dd4fb45a8ccaf12c7/opentelemetry_instrumentation_asgi-0.65b0.tar.gz", hash = "sha256:892bca67c56522ffa85a8a83cf934d7b50b3be2132e45cbee705825f0a5ba426", size = 26140, upload-time = "2026-07-16T15:25:54.544Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/0b/9c/376962840b619d2d55fe8ee2285f8c70971c090e5fff614516fc654a6f3a/opentelemetry_instrumentation_asgi-0.65b0-py3-none-any.whl", hash = "sha256:3a845a8ebd1c4ef0d8263401e6545f5b219b2feee612090d50f578a87e71fd65", size = 15903, upload-time = "2026-07-16T15:24:57.198Z" },
]

[[package]]
name = "opentelemetry-instrumentation-fastapi"
version = "0.65b0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "opentelemetry-api" },
    { name = "opentelemetry-instrumentation" },
    { name = "opentelemetry-instrumentation-asgi" },
    { name = "opentelemetry-semantic-conventions" },
    { name = "opentelemetry-util-http" },
]
sdist = { url = "https://files.pythonhosted.org/packages/30/23/b057f8196d06efdc1b50e3ff11fbc499a7d96b35c87f217eb7885542f4ea/opentelemetry_instrumentation_fastapi-0.65b0.tar.gz", hash = "sha256:10a3a9548603623041 3a58fe4fdf4a83fa6bba46918407e527476994bd92bd97", size = 26236, upload-time = "2026-07-16T15:26:05.954Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/fa/b0/c9b0300d33349ecc3dfd2362516eaffc44877e90970e6a52178ff953fec3/opentelemetry_instrumentation_fastapi-0.65b0-py3-none-any.whl", hash = "sha256:cda2610a0ec1b22d19886f33e4d861e9f5dbb886aeaa3a1263b47aff82c36943", size = 13261, upload-time = "2026-07-16T15:25:12.429Z" },
]

[[package]]
name = "opentelemetry-proto"
version = "1.44.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "protobuf" },
]
sdist = { url = "https://files.pythonhosted.org/packages/64/01/40ac4ae9a149263cc52c2cee200ddd80cb6d8db1a4610abf8eabce0fe771/opentelemetry_proto-1.44.0.tar.gz", hash = "sha256:c547a79c2f8c0c515d31509154682e5921c7cfd5ca67b70e1f9266e2c3e103f3", size = 46488, upload-time = "2026-07-16T15:25:45.34Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d1/7c/8be563d68e93bbefa5c8affb82ddcff91b3ad858ce49957ba7b16fd3e0ab/opentelemetry_proto-1.44.0-py3-none-any.whl", hash = "sha256:898b155a0e1557afd867478fb6158e8122a46329ca0bb8dc53cc55e98f017f56", size = 72483, upload-time = "2026-07-16T15:25:28.429Z" },
]

[[package]]
name = "opentelemetry-sdk"
version = "1.44.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
```

```
    { name = "opentelemetry-api" },
    { name = "opentelemetry-semantic-conventions" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/5d/77/a6592cbc7c8d9bcc9d6757a9df45e04a7c585e3e6e7a13456da522b21109/opentelemetry_sdk-1.44.0.tar.gz", hash = "sha256:cebe7f65dc12f26ead75c6064de12fd2a9052e5060c0272d402cfa203aae123b", size = 208624, upload-time = "2026-07-16T15:25:46.078Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e7/23/ff077e61886ee020a17ce9c8b6fa11c601c8d8345b09ea24f605445df62a/opentelemetry_sdk-1.44.0-py3-none-any.whl", hash = "sha256:df081c4c6bcfdb1211e3e8614037679264312 8a25f8d72d1d27675936e7e96ad", size = 137221, upload-time = "2026-07-16T15:25:29.534Z" },
]

[[package]]
name = "opentelemetry-semantic-conventions"
version = "0.65b0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "opentelemetry-api" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/8f/73/0cbdebcb4cf545fdd328da14f5137e37d0770c3f26185e478b0d15d94f50/opentelemetry_semantic_conventions-0.65b0.tar.gz", hash = "sha256:f9b2b81e9d5b64f11bc952075e7e9c7fb0aab075c7fd1c46d597f1b919852d60", size = 148774, upload-time = "2026-07-16T15:25:46.902Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a6/0e/49df70d9b81fb5cbae4bbf2a49d865b09bcbcbc4eb53f5851b1027738d78/opentelemetry_semantic_conventions-0.65b0-py3-none-any.whl", hash = "sha256:1cacde7b0ad306f84c5ef08c3dbe1bbaf20165bba6f8bff43b670e555a086bcb", size = 204645, upload-time = "2026-07-16T15:25:30.688Z" },
]

[[package]]
name = "opentelemetry-util-http"
version = "0.65b0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/32/a9/d7525a59fdd240e69b5af4a6338e78fafa1b4203394122cbd6701fb5f84a/opentelemetry_util_http-0.65b0.tar.gz", hash = "sha256:84f82d826978bba416ab453460ff6a7391cdc3534c93a786595e4068680016b7", size = 11243, upload-time = "2026-07-16T15:26:27.898Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/23/3f/ab8d29df207ce5f470a07fa96ebb48af4e95b7fab7e7635311b9a32f2fab/opentelemetry_util_http-0.65b0-py3-none-any.whl", hash = "sha256:7553b606f963097cb190536dc30556cce85090692e471a422fff30ca29b04348", size = 8245, upload-time = "2026-07-16T15:25:46.482Z" },
]

[[package]]
name = "orjson"
version = "3.11.9"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/7e/0c/964746fcafbd16f8ff53219ad9f6b412b34f345c75f384ad434ceaadb538/orjson-3.11.9.tar.gz", hash = "sha256:4fef17e1f8722c11587a6ef18e35902450221da0028e65dbaaa543619e68e48f", size = 5599163, upload-time = "2026-05-06T15:11:08.309Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/16/6d/11867a3ffa3a3608d84a4de51ef4dd0896d6b5cc9132fbe1daf593e677bc/orjson-3.11.9-cp312-cp312-macosx_10_15_x86_64.macosx_11_0_arm64.macosx_10_15_universal2.whl", hash = "sha256:9ef6fe90aadef185c7b128859f40beb24720b4ecea95379fc9000931179c3a49", size = 228515, upload-time = "2026-05-06T15:09:57.265Z" },
    { url = "https://files.pythonhosted.org/packages/24/75/05912954c8b288f34fcf5cd4b9b071cb4f6e77b9961e175e56ebb258089f/orjson-3.11.9-cp312-cp312-macosx_15_0_arm64.whl", hash = "sha256:e5c9b8f28e726e97d97696c826bc7bea5d71cecd63576dba92924a32c1961291", size = 128409, upload-time = "2026-05-06T15:09:59.063Z" },
    { url = "https://files.pythonhosted.org/packages/ab/86/1c3a47df3bc8191ea9ac51603bbb872a95167a364320c269f2557911f406/orjson-3.11.9-cp312-cp312-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:26a473dbb4162108b27901492546f83c76fdcea3d0eadff00ae7a07e18dcce09", size = 132106, upload-time = "2026-05-06T15:10:00.798Z" },
    { url = "https://files.pythonhosted.org/packages/d7/cf/b33b5f3e695ae7d63feef9d915c37cc3b8f465493dcd4f8e0b4c697a2366/orjson-3.11.9-cp312-cp312-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:011382e2a60fda9d46f1cdee31068cfc52ffe952b587d683ec0463002802a0f4", size = 127864, upload-time = "2026-05-06T15:10:02.15Z" },
    { url = "https://files.pythonhosted.org/packages/31/6a/6cf69385a58208024fcb8c014e2141b8ce838aba6492b589f8acfff97fab/orjson-3.11.9-cp312-cp312-manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:c2d3dc759490128c5c1711a53eeaa8ee1d437fd0038ffd2b6008abf46db3f882", size = 135213, upload-time = "2026-05-06T15:10:03.515Z" },
    { url = "https://files.pythonhosted.org/packages/e8/f8/0b1bd3e8f2efcdd376af5c8cfd79eaf13f018080c0089c80ebd724e3c7fb/orjson-3.11.9-cp312-cp312-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:d8ea516b3726d190e1b4297e6f4e7a8650347ae053868a18163b4dd3641d1fff", size = 145994, upload-time = "2026-05-06T15:10:05.083Z" },
    { url = "https://files.pythonhosted.org/packages/f3/59/dab79f61044c529d2c81aecdc589b1f833a1c8dec11ba3b1c2498a02ca7e/orjson-3.11.9-cp312-cp312-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:380cdce7ba24989af81d0a7013d0aaec5d0e2a21734c0e2681b1bc4f141957fe", size = 132744, upload-time = "2026-05-06T15:10:06.853Z" },
    { url = "https://files.pythonhosted.org/packages/0e/a4/82b7a2fe5d8a67a59ed831b24d59a3d46ea7d207b66e1602d376541d94a6/orjson-3.11.9-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:be4fa4f0af7fa18951f7ab3fc2148e223af211bf03f59e1c6034ec3f97f21d61", size = 134014, upload-time = "2026-05-06T15:10:08.213Z" },
    { url = "https://files.pythonhosted.org/packages/50/c7/375e83a76851b73b2e39f3bcf0e5a19e2b89bad13e5bca97d0b293d27f24/orjson-3.11.9-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:a8f5f8bc7ce7d59f08d9f99fa510c06496164a24cb5f3d34537dbd9ca30132e2", size = 141509, upload-time = "2026-05-06T15:10:09.595Z" },
    { url = "https://files.pythonhosted.org/packages/7f/7c/49d5d82a3d3097f641f094f552131f1e2723b0b8cb0fa2874ab65ecfffa6/orjson-3.11.9-cp312-cp312-musllinux_1_2_armv7l.whl", hash = "sha256:4d7fde5501b944f83b3e665e1b31343ff6e154b15560a16b7130ea1e594a4206", size = 415127, upload-time = "2026-05-06T15:10:11.049Z" },
    { url = "https://files.pythonhosted.org/packages/3a/dc/7446c538590d55f455647e5f3c61fc33f7108714e7afcffa6a2a033f8350/orjson-3.11.9-cp312-cp312-musllinux_1_2_i686.whl", hash = "sha256:cde1a448023ba7d5bb4c01c5afb48894380b5e4956e0627266526587ef4e535f", size = 148025, upload-time = "2026-05-06T15:10:12.842Z" },
    { url = "https://files.pythonhosted.org/packages/df/e5/4d2d8af06f788329b4f78f8cc3679bb395392fcaa1e4d8d3c33e85308fa4/orjson-3.11.9-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:71e63adb0e1f1ed5d9e168f50a91ceb93ae6420731d222dc7da5c69409aa47aa", size = 136943, upload-time = "2026-05-06T15:10:14.405Z" },
    { url = "https://files.pythonhosted.org/packages/06/69/850264ccf6d80f6b174620d30a87f65c9b1490aba33fe6b62798e618cad3/orjson-3.11.9-cp312-cp312-win32.whl", hash = "sha256:2d057a602cdd19a0ad680417527c45b6961a095081c0f46fe0e03e304aac6470", size = 131606, upload-time = "2026-05-06T15:10:15.791Z" },
    { url = "https://files.pythonhosted.org/packages/b9/d5/973a43fc9c55e20f2051e9830997649f669be0cb3ca52192087c0143f118/orjson-3.11.9-cp312-cp312-win_amd64.whl", hash = "sha256:59e403b1cc5a676da8eaf31f6254801b7341b3e29efa85f92b48d272637e77be", size = 127101, upload-time = "2026-05-06T15:10:17.129Z" },
    { url = "https://files.pythonhosted.org/packages/fe/ae/495470f0e4a18f73fa10b7f6b84b464ec4cc5291c4e0c7c2a6c400bef006/orjson-3.11.9-cp312-cp312-win_arm64.whl", hash = "sha256:9af678d6488357948f1f84c6cd1c1d397c014e1ae2f98ae082a44eb48f602624", size = 126736, upload-time = "2026-05-06T15:10:18.645Z" },
    { url = "https://files.pythonhosted.org/packages/32/33/93fcc25907235c344ae73122f8a4e01d2d393ef062b4af7d2e2487a32c37/orjson-3.11.9-cp313-cp313-macosx_10_15_x86_64.macosx_11_0_arm64.macosx_10_15_universal2.whl", hash = "sha256:4bab1b2d6141fe7b32ae71dac905666ece4f94936efbfb13d55bb7739a3a6021", size = 228458, upload-time = "2026-05-06T15:10:20.079Z" },
    { url = "https://files.pythonhosted.org/packages/8f/27/b1e6dadb3c080313c03fdd8067b85e6a0460c7d8d6a1c3984ef77b904e4d/orjson-3.11.9-cp313-cp313-macosx_15_0_arm64.whl", hash = "sha256:844417969855fc7a41be124aafe83dc424592a7f77cd4501900c67307122b92c", size = 128368, upload-time = "2026-05-06T15:10:21.549Z" },
    { url = "https://files.pythonhosted.org/packages/21/0f/c9ede0bf052f6b4051e64a7d4fa91b725cccf8321a6a786e86eb03519f00/orjson-3.11.9-cp313-cp313-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:ffe02797b5e9f3a9d8292ddcd289b474ad13e81ad83cd1891a240811f1d2cb81", size = 132070, upload-time = "2026-05-06T15:10:23.371Z" },
    { url = "https://files.pythonhosted.org/packages/fd/26/d398e28048dc18205bbe812f2c88cb9b40313db2470778e25964796458fe/orjson-3.11.9-cp313-cp313-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:0e4eed3b200023042814d2fc8a5d2e880f13b52e1ed2485e83da4f3962f7dc1a", size = 127892, upload-time = "2026-05-06T15:10:24.714Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/66/60/52b0054c4c700d5aa7fc5b7ca96917400d8f061307778578e67a10e25852/orjson-3.11.9-cp313-cp313-manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:8aff7da9952a5ad1cef8e68017724d96c7b9a66e99e91d6252e1b133d67a7b10", size = 135217, upload-time = "2026-05-06T15:10:26.084Z" },
    { url = "https://files.pythonhosted.org/packages/d5/97/1e3dc2b2a28b7b2528f403d2fc1d79ec5f39af3bc143ab65d3ec26426385/orjson-3.11.9-cp313-cp313-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:4d4e98d6f3b8afed8bc8cd9718ec0cdf46661826beefb53fe8eafb37f2bf0362", size = 145980, upload-time = "2026-05-06T15:10:28.062Z" },
    { url = "https://files.pythonhosted.org/packages/fc/39/31fbfe7850f2de32dee7e7e5c09f26d403ab01e440ac96001c6b01ad3c99/orjson-3.11.9-cp313-cp313-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:3a81d52442a7c99b3662333235b3adf96a1715864658b35bb797212be7bddb97", size = 132738, upload-time = "2026-05-06T15:10:29.727Z" },
    { url = "https://files.pythonhosted.org/packages/a1/08/dca0082dd2a194acb93e5457e73455388e2e2ca464a2672449a9ddbb679d/orjson-3.11.9-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:4e39364e726a8fff737309aff059ff67d8a8c8d5b677be7bb49a8b3e84b7e218", size = 134033, upload-time = "2026-05-06T15:10:31.152Z" },
    { url = "https://files.pythonhosted.org/packages/11/d4/5bdb062680123013998738555 4c5d4c42255218ac906525bf4347f22cd95/orjson-3.11.9-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:4fd66214623f1b17501df9f0543bef0b833979ab5b6ded1e1d123222866aa8c9", size = 141492, upload-time = "2026-05-06T15:10:32.641Z" },
    { url = "https://files.pythonhosted.org/packages/fa/88/a21fb53b3ede703aede6dce4710ed4111e5b201cfa6bbff5e544f9d47d7/orjson-3.11.9-cp313-cp313-musllinux_1_2_armv7l.whl", hash = "sha256:8ecc30f10465fa1e0ce13fd01d9e22c316e5053a719a8d915d4545a09a5ff677", size = 415087, upload-time = "2026-05-06T15:10:34.438Z" },
    { url = "https://files.pythonhosted.org/packages/3d/57/1b30daf70f0d8180e9a73cefbfbdd99e4bf19eb020466502b01fba7e0e50/orjson-3.11.9-cp313-cp313-musllinux_1_2_i686.whl", hash = "sha256:97db4c94a7db398a5bd636273324f0b3fd58b350bbbac8bb380ceb825a9b40f4", size = 148031, upload-time = "2026-05-06T15:10:36.358Z" },
    { url = "https://files.pythonhosted.org/packages/04/83/45fbb6d962e260807f99441db9613cee868ceda4baceda59b3720a563f97/orjson-3.11.9-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:9f78cf8fec5bd627f4082b8dfeac7871b43d7f3274904492a43dab39f18a19a0", size = 136915, upload-time = "2026-05-06T15:10:38.013Z" },
    { url = "https://files.pythonhosted.org/packages/5f/cc/2d10025f9056d376e4127ec05a5808b218d46f035fdc08178a5411b34250/orjson-3.11.9-cp313-cp313-win32.whl", hash = "sha256:d4087e5c0209a0a8efe4de3303c234b9c44d1174161dcd851e8eea07c7560b32", size = 131613, upload-time = "2026-05-06T15:10:39.569Z" },
    { url = "https://files.pythonhosted.org/packages/67/bd/2775ff28bfe883b9aa1ff348300542eb2ef1ee18d8ae0e3a49846817a865/orjson-3.11.9-cp313-cp313-win_amd64.whl", hash = "sha256:051b102c93b4f634e89f3866b07b9a9a98915ada541f4ec30f177067b2694979", size = 127086, upload-time = "2026-05-06T15:10:41.262Z" },
    { url = "https://files.pythonhosted.org/packages/91/2b/d26799e580939e32a7da9a39531bc9e58e15ca32ffaa6a8cb3e9bb0d22cd/orjson-3.11.9-cp313-cp313-win_arm64.whl", hash = "sha256:cce9127885941bd28f080cecf1f1d288336b7e0d812c345b08be88b572796254", size = 126696, upload-time = "2026-05-06T15:10:42.651Z" },
    { url = "https://files.pythonhosted.org/packages/8e/eb/5da01e356015aee6ecfa1187ced87aef51364e306f5e695dd52719bf0e78/orjson-3.11.9-cp314-cp314-macosx_10_15_x86_64.macosx_11_0_arm64.macosx_10_15_universal2.whl", hash = "sha256:b6ef1979adc4bc243523f1a2ba91418030a8e29b0a99cbe7e0e2d6807d4dce6e", size = 228465, upload-time = "2026-05-06T15:10:44.097Z" },
    { url = "https://files.pythonhosted.org/packages/64/62/3e0e0c14c957133bcd855395c62b55ed4e3b0af23ffea11b032cb1dcbdb1/orjson-3.11.9-cp314-cp314-macosx_15_0_arm64.whl", hash = "sha256:f36b7f32c7c0db4a719f1fc5824db4a9c6f8bd1a354debb91faf26ebf3a4c71e", size = 128364, upload-time = "2026-05-06T15:10:45.839Z" },
    { url = "https://files.pythonhosted.org/packages/5a/5a/07d8aa117211a8ed7630bda80c8c0b14d04e0f8dcf99bcf49656e4a710eb/orjson-3.11.9-cp314-cp314-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:08f4d8ebb44925c794e535b2bebc507cebf32209df81de22ae285fb0d8d66de0", size = 132063, upload-time = "2026-05-06T15:10:47.267Z" },
    { url = "https://files.pythonhosted.org/packages/d6/ec/4acaf21483e18aa945be74a474c74b434f284b549f275a0a39b9f98956e9/orjson-3.11.9-cp314-cp314-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:6cc7923789694fd58f001cbcac7e47abc13af4d560ebbfcf3b41a8b1a0748124", size = 122356, upload-time = "2026-05-06T15:10:48.765Z" },
    { url = "https://files.pythonhosted.org/packages/13/d8/5f0555e7638801323b7a75850f92e7dfa891bc84fe27a1ba4449170d1200/orjson-3.11.9-cp314-cp314-manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:ea5c46eb2d3af39e806b986f4b09d5c2706a1f5afde3cbf7544ce6616127173c", size = 129592, upload-time = "2026-05-06T15:10:50.13Z" },
    { url = "https://files.pythonhosted.org/packages/b6/30/ed9860412a3603ceb3c5955bfd72d28b9d0e7ba6ed81add14f83d7114236/orjson-3.11.9-cp314-cp314-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:f5d89a2ed90731df3be64bab0aa44f78bff39fdc9d71c291f4a8023aa46425b7", size = 140491, upload-time = "2026-05-06T15:10:51.582Z" },
    { url = "https://files.pythonhosted.org/packages/d0/17/adc514dea7ac7c505527febf884934b815d34f0c7b8693c1a8b39c5c4a57/orjson-3.11.9-cp314-cp314-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:25e4aed0312d292c09f61af25bba34e0b2c88546041472b09088c39a4d828af1", size = 127309, upload-time = "2026-05-06T15:10:53.329Z" },
    { url = "https://files.pythonhosted.org/packages/76/3e/c0b690253f0b82d86e99949af13533363acfb5432ecb5d53dd5b3bce9c34/orjson-3.11.9-cp314-cp314-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:aaea64f3f467d22e70eeed68bdccb3bc4f83f650446c4a03c59f2cba28a108db", size = 134030, upload-time = "2026-05-06T15:10:54.988Z" },
    { url = "https://files.pythonhosted.org/packages/c1/7a/bc82a0bb25e9faaf92dc4d9ef002732efc09737706af83e346788641d4a7/orjson-3.11.9-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:a028425d1b440c5d92a6be1e1a020739dfe67ea87d96c6dbe828c1b30041728b", size = 141482, upload-time = "2026-05-06T15:10:56.663Z" },
    { url = "https://files.pythonhosted.org/packages/01/55/e69188b939f77d5d32a9833745ace31ea5ccae3ab613a1ec185d3cd2c4fb/orjson-3.11.9-cp314-cp314-musllinux_1_2_armv7l.whl", hash = "sha256:5b192c6cf397e4455b11523c5cf2b18ed084c1bbd61b6c0926344d2129481972", size = 415178, upload-time = "2026-05-06T15:10:58.446Z" },
    { url = "https://files.pythonhosted.org/packages/2e/1a/b8a5a7ac527e80b9cb11d51e3f6689b709279183264b9ec5c7bc680bb8b5/orjson-3.11.9-cp314-cp314-musllinux_1_2_i686.whl", hash = "sha256:ea407d4ccf5891d667d045fecae97a7a1e5e87b3b97f97ae1803c2e741130be0", size = 148089, upload-time = "2026-05-06T15:11:00.441Z" },
    { url = "https://files.pythonhosted.org/packages/97/4e/00503f64204bf859b37213a63927028f30fb6268cd8677fb0a5ad48155e1/orjson-3.11.9-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:5f63aaf97afd9f6dec5b1a68e1b8da12bfccb4cb9a9a65c3e0b6c847849e7586", size = 136921, upload-time = "2026-05-06T15:11:02.176Z" },
    { url = "https://files.pythonhosted.org/packages/0d/ba/a23b82a0a8d0ed7bed4e5f5035aae751cad4ff6a1e8d2ecd14d8860f5929/orjson-3.11.9-cp314-cp314-win32.whl", hash = "sha256:e30ab17845bb9fa54ccf67fa4f9f5282652d54faa6d17452f47d0f369d038673", size = 131638, upload-time = "2026-05-06T15:11:03.696Z" },
    { url = "https://files.pythonhosted.org/packages/f3/c3/0c6798456bade745c75c452342dabacce5798196483e77e643be1f53877d/orjson-3.11.9-cp314-cp314-win_amd64.whl", hash = "sha256:32ef5f4283a3be81913947d19608eacb7c6608026851123790cd9cc8982af34b", size = 127078, upload-time = "2026-05-06T15:11:05.123Z" },
    { url = "https://files.pythonhosted.org/packages/16/21/5a3f1e8913103b703a436a5664238e5b965ec392b555fe68943ea3691e6b/orjson-3.11.9-cp314-cp314-win_arm64.whl", hash = "sha256:eebdbdeef0094e4f5aefa20dcd4eb2368ab5e7a3b4edea27f1e7b2892e009cf9", size = 126687, upload-time = "2026-05-06T15:11:06.602Z" },
]

[[package]]
name = "ormsgpack"
version = "1.12.2"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/12/0c/f1761e21486942ab9bb6feaebc610fa074f7c5e496e6962dea5873348077/ormsgpack-1.12.2.tar.gz", hash = "sha256:944a2233640273bee67521795a73cf1e959538e0dfb7ac635505010455e53b33", size = 39031, upload-time = "2026-01-18T20:55:28.023Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/4c/36/16c4b1921c308a92cef3bf6663226ae283395aa0ff6e154f925c32e91ff5/ormsgpack-1.12.2-cp312-cp312-macosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:7a29d09b64b9694b588ff2f80e9826bdceb3a2b91523c5beae1fab27d5c940e7", size = 378618, upload-time = "2026-01-18T20:55:50.835Z" },
    { url = "https://files.pythonhosted.org/packages/c0/68/468de634079615abf66ed13bb5c34ff71da237213f29294363beeeca5306/ormsgpack-1.12.2-cp312-cp312-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:0b39e629fd2e1c5b2f46f99778450b59454d1f901bc507963168985e79f09c5d", size = 203186, upload-time = "2026-01-18T20:56:11.163Z" },
    { url = "https://files.pythonhosted.org/packages/73/a9/d756e01961442688b7939bacd87ce13bfad7d26ce24f910f6028178b2cc8/ormsgpack-1.12.2-cp312-cp312-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:958dcb270d30a7cb633a45ee62b9444433fa571a752d2ca484efdac07480876e", size = 210738, upload-time = "2026-01-18T20:56:09.181Z" },
    { url = "https://files.pythonhosted.org/packages/7b/ba/795b1036888542c9113269a3f5690ab53dd2258c6fb17676ac4bd44fcf94/ormsgpack-1.12.2-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:58d379d72b6c5e964851c77cfedfb386e474adee4fd39791c2c5d9efb53505cc", size = 212569, upload-time = "2026-01-18T20:56:06.135Z" },
    { url = "https://files.pythonhosted.org/packages/6c/aa/bff73c57497b9e0cba8837c7e4bcab584b1a6dbc91a5dd5526784a5030c8/ormsgpack-1.12.2-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:8463a3fc5f09832e67bdb0e2fda6d518dc4281b133166146a67f54c08496442e", size = 387166, upload-time = "2026-01-18T20:55:36.738Z" },
    { url = "https://files.pythonhosted.org/packages/d3/cf/f8283cba44bcb7b14f97b6274d449db276b3a86589bdb363169b51bc12de/ormsgpack-1.12.2-cp312-cp312-musllinux_1_2_armv7l.whl", hash = "sha256:eddffb77eff0bad4e67547d67a130604e7e2dfbb7b0cde0796045be4090f35c6", size = 482498, upload-time = "2026-01-18T20:55:29.626Z" },
    { url = "https://files.pythonhosted.org/packages/05/be/71e37b852d723dfcbe952ad04178c030df60d6b78eba26bfd14c9a40575e/ormsgpack-1.12.2-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:fcd55e5f6ba0dbce624942adf9f152062135f991a0126064889f68eb850de0dd", size = 425518, upload-time = "2026-01-18T2
```

```
0:55:49.556Z" },
    { url = "https://files.pythonhosted.org/packages/7a/0c/9803aa883d18c7ef197213cd2cbf73ba76472a11fe100fb7dab2884edf48/ormsgpack-1.12.2-cp312-cp312-w
in_amd64.whl", hash = "sha256:d024b40828f1dde5654faebd0d824f9cc29ad46891f626272dd5bfd7af2333a4", size = 117462, upload-time = "2026-01-18T20:55:47.726
Z" },
    { url = "https://files.pythonhosted.org/packages/c8/9e/029e898298b2cc662f10d7a15652a53e3b525b1e7f07e21fef8536a09bb8/ormsgpack-1.12.2-cp312-cp312-w
in_arm64.whl", hash = "sha256:da538c542bac7d1c8f3f2a937863dba36f013108ce63e55745941dda4b75dbb6", size = 111559, upload-time = "2026-01-18T20:55:54.273
Z" },
    { url = "https://files.pythonhosted.org/packages/eb/29/bb0eba3288c0449efbb013e9c6f58aea79cf5cb9ee1921f8865f04c1a9d7/ormsgpack-1.12.2-cp313-cp313-m
acosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:5ea60cb5f210b1cfbad8c002948d73447508e629ec375acb82910e3efa8ff355", s
ize = 378661, upload-time = "2026-01-18T20:55:57.765Z" },
    { url = "https://files.pythonhosted.org/packages/6e/31/5efa31346affdac489acade2926989e019e8ca98129658a183e3add7af5e/ormsgpack-1.12.2-cp313-cp313-m
anylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:f3601f19afdbea273ed70b06495e5794606a8b690a568d6c996a90d7255e51c1", size = 203194, upl
oad-time = "2026-01-18T20:56:08.252Z" },
    { url = "https://files.pythonhosted.org/packages/eb/56/d0087278beef833187e0167f8527235ebe6f6ffc2a143e9de12a98b1ce87/ormsgpack-1.12.2-cp313-cp313-m
anylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:29a9f17a3dac6054c0dce7925e0f4995c727f7c41859adf9b5572180f640d172", size = 210778, uploa
d-time = "2026-01-18T20:55:17.694Z" },
    { url = "https://files.pythonhosted.org/packages/1c/a2/072343e1413d9443e5a252a8eb591c2d5b1bffbe5e7bfc78c069361b92eb/ormsgpack-1.12.2-cp313-cp313-m
anylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:39c1bd2092880e413902910388be8715f70b9f15f20779d44e673033a6146f2d", size = 212592, uploa
d-time = "2026-01-18T20:55:32.747Z" },
    { url = "https://files.pythonhosted.org/packages/a2/8b/a0da3b98a91d41187a63b02dda14267eefc2a74fcb43cc2701066cf1510e/ormsgpack-1.12.2-cp313-cp313-m
usllinux_1_2_aarch64.whl", hash = "sha256:50b7249244382209877deedeee838aef1542f3d0fc28b8fe71ca9d7e1896a0d7", size = 387164, upload-time = "2026-01-18T
20:55:40.853Z" },
    { url = "https://files.pythonhosted.org/packages/19/bb/6d226bc4cf9fc20d8eb1d976d027a3f7c3491e8f08289a2e76abe96a65f3/ormsgpack-1.12.2-cp313-cp313-m
usllinux_1_2_armv7l.whl", hash = "sha256:5af04800d844451cf102a59c74a841324868d3f1625c296a06cc655c542a6685", size = 482516, upload-time = "2026-01-18T2
0:55:42.033Z" },
    { url = "https://files.pythonhosted.org/packages/fb/f1/bb2c7223398543dedb3dbf8bb93aaa737b387de61c5feaad6f908841b782/ormsgpack-1.12.2-cp313-cp313-m
usllinux_1_2_x86_64.whl", hash = "sha256:cec70477d4371cd524534cd16472d8b9cc187e0e3043a8790545a9a9b296c258", size = 425539, upload-time = "2026-01-18T2
0:55:24.727Z" },
    { url = "https://files.pythonhosted.org/packages/7b/e8/0fb45f57a2ada1fed374f7494c8cd55e2f88ccd0ab0a669aa3468716bf5f/ormsgpack-1.12.2-cp313-cp313-w
in_amd64.whl", hash = "sha256:21f4276caca5c03a818041d637e4019bc84f9d6ca8baa5ea03e5cc8bf56140e9", size = 117459, upload-time = "2026-01-18T20:55:56.876
Z" },
    { url = "https://files.pythonhosted.org/packages/7a/d4/0cfeea1e960d550a131001a7f38a5132c7ae3ebde4c82af1f364ccc5d904/ormsgpack-1.12.2-cp313-cp313-w
in_arm64.whl", hash = "sha256:baca4b6773d20a82e36d6fd25f341064244f9f86a13dead95dd7d7f996f51709", size = 111577, upload-time = "2026-01-18T20:55:43.605
Z" },
    { url = "https://files.pythonhosted.org/packages/94/16/24d18851334be09c25e87f74307c84950f18c324a4d3c0b41dabdbf19c29/ormsgpack-1.12.2-cp314-cp314-m
acosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:bc68dd5915f4acf66ff2010ee47c8906dc1cf07399b16f4089f8c71733f6e36c", s
ize = 378717, upload-time = "2026-01-18T20:55:26.164Z" },
    { url = "https://files.pythonhosted.org/packages/b5/a2/88b9b56f83adae8032ac6a6fa7f080c65b3baf9b6b64fd3d37bd202991d4/ormsgpack-1.12.2-cp314-cp314-m
anylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:46d084427b4132553940070ad95107266656cb646ea9da4975f85cb1a6676553", size = 203183, upl
oad-time = "2026-01-18T20:55:18.815Z" },
    { url = "https://files.pythonhosted.org/packages/a9/80/43e4555963bf602e5bdc79cbc8debd8b6d5456c00d2504df9775e74b450b/ormsgpack-1.12.2-cp314-cp314-m
anylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:c010da16235806cf1d7bc4c96bf286bfa91c686853395a299b3ddb49499a3e13", size = 210814, uploa
d-time = "2026-01-18T20:55:33.973Z" },
    { url = "https://files.pythonhosted.org/packages/78/e1/7cfbf28de8bca6efe7e525b329c31277d1b64ce08dcba723971c241a9d60/ormsgpack-1.12.2-cp314-cp314-m
anylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:18867233df592c997154ff942a6503df274b5ac1765215bceba7a231bea2745d", size = 212634, uploa
d-time = "2026-01-18T20:55:28.634Z" },
    { url = "https://files.pythonhosted.org/packages/95/f8/30ae5716e88d792a4e879debee195653c26ddd3964c968594ddef0a3cc7e/ormsgpack-1.12.2-cp314-cp314-m
usllinux_1_2_aarch64.whl", hash = "sha256:b009049086ddc6b8f80c76b3955df1aa22a5fbd7673c525cd63bf91f23122ede", size = 387139, upload-time = "2026-01-18T
20:56:02.013Z" },
    { url = "https://files.pythonhosted.org/packages/dc/81/aee5b18a3e3a0e52f718b37ab4b8af6fae0d9d6a65103036a90c2a8ffb5d/ormsgpack-1.12.2-cp314-cp314-m
usllinux_1_2_armv7l.whl", hash = "sha256:1dcc17d92b6390d4f18f937cf0b99054824a7815818012ddca925d6e01c2e49e", size = 482578, upload-time = "2026-01-18T2
0:55:35.117Z" },
    { url = "https://files.pythonhosted.org/packages/bd/17/71c9ba472d5d45f7546317f467a5fc941929cd68fb32796ca3d13dcbaec2/ormsgpack-1.12.2-cp314-cp314-m
usllinux_1_2_x86_64.whl", hash = "sha256:f04b5e896d510b07c0ad733d7fce2d44b260c5e6c402d272128f8941984e4285", size = 425539, upload-time = "2026-01-18T2
0:56:04.009Z" },
    { url = "https://files.pythonhosted.org/packages/2e/a6/ac99cd7fe77e822fed5250ff4b86fa66dd4238937dd178d2299f10b69816/ormsgpack-1.12.2-cp314-cp314-w
in_amd64.whl", hash = "sha256:ae3aba7eed4ca7cb79fd3436eddd29140f17ea254b91604aa1eb19bfcedb990f", size = 117493, upload-time = "2026-01-18T20:56:07.343
Z" },
    { url = "https://files.pythonhosted.org/packages/3a/67/339872846a1ae4592535385a1c1f93614138566d7af094200c9c3b45d1e5/ormsgpack-1.12.2-cp314-cp314-w
in_arm64.whl", hash = "sha256:118576ea6006893aea811b17429bfc561b4778fad393f5f538c84af70b01260c", size = 111579, upload-time = "2026-01-18T20:55:21.161
Z" },
    { url = "https://files.pythonhosted.org/packages/49/c2/6feb972dc87285ad381749d3882d8aecbde9f6ecf908dd717d33d66df095/ormsgpack-1.12.2-cp314-cp314t-
macosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:7121b3d355d3858781dc40dafe25a32ff8a8242b9d80c692fd548a4b1f7fd3c8",
size = 378721, upload-time = "2026-01-18T20:55:52.12Z" },
    { url = "https://files.pythonhosted.org/packages/a3/9a/900a6b9b413e0f8a471cf07830f9cf65939af039a362204b36bd5b581d8b/ormsgpack-1.12.2-cp314-cp314t-
manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:4ee766d2e78251b7a63daf1cddfac36a73562d3ddef68cacfb41b2af64698033", size = 203170, up
load-time = "2026-01-18T20:55:44.469Z" },
    { url = "https://files.pythonhosted.org/packages/87/4c/27a95466354606b256f24fad464d7c97ab62bce6cc529dd4673e1179b8fb/ormsgpack-1.12.2-cp314-cp314t-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:292410a7d23de9b40444636b9b8f1e4e4b814af7f1ef476e44887e52a123f09d", size = 212816, uplo
ad-time = "2026-01-18T20:55:23.501Z" },
    { url = "https://files.pythonhosted.org/packages/73/cd/29cee6007bddf7a834e6cd6f536754c0535fcb939d384f0f37a38b1cddb8/ormsgpack-1.12.2-cp314-cp314t-
win_amd64.whl", hash = "sha256:837dd316584485b72ef451d08dd3e96c4a11d12e4963aedb40e08f89685d8ec2", size = 117232, upload-time = "2026-01-18T20:55:45.44
8Z" },
]

[[package]]
name = "packaging"
version = "26.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/7d/fa/3944b40b07da9ce895c0e6303a5ab7d53da063554f534556b134a54d6093/packaging-26.3.tar.gz", ha
sh = "sha256:94edc256424af38762eb31306eed28beb9f0efc50a8837492c9d6fd6004aed79", size = 313412, upload-time = "2026-08-04T18:15:28.737Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/63/34/ba1c580383c9eada3711951fef0795c80b829a078d72188184bcab9dd527/packaging-26.3-py3-none-any.wh
l", hash = "sha256:d7193f7c8e4e93f444fde0262bf90af30e16fa0ad0ad44cb553c87339b23cd1c", size = 129956, upload-time = "2026-08-04T18:15:27.159Z" },
]

[[package]]
name = "pandas"
version = "3.0.5"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "numpy" },
    { name = "python-dateutil" },
    { name = "tzdata", marker = "sys_platform == 'emscripten' or sys_platform == 'win32'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/be/4f/5f3422a2afec5ffc46308b79e53291365a93748b498ac2e58bead0197916/pandas-3.0.5.tar.gz", hash
 = "sha256:dca3734d6ab7c906e6730f0788b0a1dbb9f2467731f9711f77995c8e9d62d712", size = 4658219, upload-time = "2026-07-22T22:19:28.819Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/1c/54/1dc810ea558d1320b597aa140a514f2fdf1d2ea09c38cf556f13ea712ec9/pandas-3.0.5-cp312-cp312-macos
x_10_13_x86_64.whl", hash = "sha256:fa290c16964d4963fbfbc358928239cf3bd755b20e988ce944877def2f44471d", size = 10411717, upload-time = "2026-07-22T22:1
8:08.307Z" },
    { url = "https://files.pythonhosted.org/packages/68/56/fbe81c09195924d8b7b8d4461a20458fe80a6a5ed6b24f0314da684277e1/pandas-3.0.5-cp312-cp312-macos
x_11_0_arm64.whl", hash = "sha256:c2e26bb46934b8a2ca0c3de1d3d606fc5f6746584791b2db264d58cf370e08dc", size = 9957095, upload-time = "2026-07-22T22:18:1
0.6Z" },
    { url = "https://files.pythonhosted.org/packages/e0/51/fac252f4a913ed5eabf3c11b880a9e8d5a6c10f0b2129d0462212d238b4d/pandas-3.0.5-cp312-cp312-manyl
```

```
inux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:73fa87b08a7ef706f8aafda39ddaccf2a99047bea62d8c88a0361bcafb2237bc", size = 10485458, uplo
ad-time = "2026-07-22T22:18:12.834Z" },
    { url = "https://files.pythonhosted.org/packages/12/98/e976540c1addf70442be7842a18cf70884a964abbf69442504f4d2939989/pandas-3.0.5-cp312-cp312-manyl
inux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:d373ce03ffd84010ed9839fa73672a9c8256990532e158440c0085db7d914b34", size = 10998091, upload
-time = "2026-07-22T22:18:15.209Z" },
    { url = "https://files.pythonhosted.org/packages/a4/8c/1f29b5be8d3fc47dd7567eb167fabba2085879b31e0287ce7cba6d3d2ff4/pandas-3.0.5-cp312-cp312-musll
inux_1_2_aarch64.whl", hash = "sha256:2a29c53d85ea98c5e792c59ef82ee9fbe6ca902c0d0adb6b23f45ef894cd7bf6", size = 11499501, upload-time = "2026-07-22T22
:18:17.689Z" },
    { url = "https://files.pythonhosted.org/packages/9d/e2/bd9c98ad2df7b38bde002adde4cdf353519da51881634323b126c55997f9/pandas-3.0.5-cp312-cp312-musll
inux_1_2_x86_64.whl", hash = "sha256:a5ad3b02ed6bc7d7ae9b70804b2c6aa31827489d150f8e623ce82491b82085d7", size = 12060559, upload-time = "2026-07-22T22:
18:20.147Z" },
    { url = "https://files.pythonhosted.org/packages/f3/9a/ffbd852d58bd74a617fe2f8ee6a58a96982271ce41cf981eab22190b4a4b/pandas-3.0.5-cp312-cp312-pyems
cripten_2024_0_wasm32.whl", hash = "sha256:b2acb4650527eec6822c3dadb2b771277b65e7dae7a267d4bccf65fd1bb3fbce", size = 7197652, upload-time = "2026-07-2
2T22:18:22.502Z" },
    { url = "https://files.pythonhosted.org/packages/70/b5/d2d3e9ae73362ba4229651b0ee1455cf78073a1ce585f6ff693782ce263e/pandas-3.0.5-cp312-cp312-win_a
md64.whl", hash = "sha256:80a611068e8a3ac23f7398c6c14eb46dc974e5cc9997f653e2dcfd1da74edd41", size = 9831691, upload-time = "2026-07-22T22:18:24.534Z"
},
    { url = "https://files.pythonhosted.org/packages/52/51/dea1e89d6a6796b9c43f85a09b484ee03edb8a4c4842e73e200a8c11301c/pandas-3.0.5-cp312-cp312-win_a
rm64.whl", hash = "sha256:25ff585b972a18ef1fe9ffa3ac6544d9950508aa76832e5147640b6022821e49", size = 9105796, upload-time = "2026-07-22T22:18:27.064Z"
},
    { url = "https://files.pythonhosted.org/packages/bf/09/7b95c4a0025227d6f118c4039b423412ac6a982db02864166185d812fbc7/pandas-3.0.5-cp313-cp313-macos
x_10_13_x86_64.whl", hash = "sha256:c1c05a767fe8e5b4fe9e1c29806829c582052eaedb9120a3da83ba3f69e24a5b", size = 10385742, upload-time = "2026-07-22T22:1
8:29.346Z" },
    { url = "https://files.pythonhosted.org/packages/8d/0c/dc78fd8c4da477b4b5e8ad37295af352190d21ef63a9ee1bc071753074cc/pandas-3.0.5-cp313-cp313-macos
x_11_0_arm64.whl", hash = "sha256:b86765f268b56f7e665b93bce9d5df69dee7f99e595cf8fb839483ab315942a3", size = 9932067, upload-time = "2026-07-22T22:18:3
1.833Z" },
    { url = "https://files.pythonhosted.org/packages/3e/71/3592c055cf44df9808550f9368ceda80ff2b224d355ef73fe251dcda1802/pandas-3.0.5-cp313-cp313-manyl
inux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:c597ecf5616b5c420372c1d4d4c00dbbfba7398bea857dcc984347e1ea48417b", size = 10466756, uplo
ad-time = "2026-07-22T22:18:34.195Z" },
    { url = "https://files.pythonhosted.org/packages/e3/70/4363150359f95b4cb4bcbb34ca23572bb5495749a621a8f3d5a1ddfd293c/pandas-3.0.5-cp313-cp313-manyl
inux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:4b11c36e218331d0387cbe3a0a5f75162357a1d92d57b2b08a336ff94b19b2be", size = 10938525, upload
-time = "2026-07-22T22:18:36.81Z" },
    { url = "https://files.pythonhosted.org/packages/f7/d0/317e7a0c67c0e69fa905a0161409397a7dc2d46ff611f6ca4803352c042b/pandas-3.0.5-cp313-cp313-musll
inux_1_2_aarch64.whl", hash = "sha256:cf52e1f61d229496da17dc7ab54acdee627357e7008fd4fecba3d0ba2937fa58", size = 11489303, upload-time = "2026-07-22T22
:18:39.287Z" },
    { url = "https://files.pythonhosted.org/packages/f1/8d/36dade89b49e4f9d5cbdbe863772581f98c0c6d78fc39ad4c557f6f2e17e/pandas-3.0.5-cp313-cp313-musll
inux_1_2_x86_64.whl", hash = "sha256:db172144bb56422bd157812f3b021eacc255451470b31e2c633c349490a1cfee", size = 11989004, upload-time = "2026-07-22T22:
18:42.208Z" },
    { url = "https://files.pythonhosted.org/packages/9c/ba/18c4ec8a746e177da05a9e7a7963781d8ea195780724f854601b6ebd6b78/pandas-3.0.5-cp313-cp313-win_a
md64.whl", hash = "sha256:0d298e951f23016ce4699951d044ae6418dbc91bf68cefca0f77666fcbb4e5c6", size = 9826896, upload-time = "2026-07-22T22:18:44.539Z"
},
    { url = "https://files.pythonhosted.org/packages/de/ec/28a57266b753799a87b8bc79e7887ac6fd981b8c6d2978a0b7e7b6bd708c/pandas-3.0.5-cp313-cp313-win_a
rm64.whl", hash = "sha256:66266d3442a5e8b3c90274c2b8b230bee42dd1c286bc822cc2f9f2c7e12b883e", size = 9094790, upload-time = "2026-07-22T22:18:47.468Z"
},
    { url = "https://files.pythonhosted.org/packages/51/2f/cf6aae281264f4463f0875bcbb15fd2bb6d291cc535187dad1732475e4a9/pandas-3.0.5-cp314-cp314-macos
x_10_15_x86_64.whl", hash = "sha256:2f264fc46911cc8131a7322a16199bbf8e353d27c10bb211f5bd0c814324dc36", size = 10390034, upload-time = "2026-07-22T22:1
8:49.818Z" },
    { url = "https://files.pythonhosted.org/packages/06/ec/5189518c7a7659c4bdcc6b1eb32c46c6f3c86b0661ffd84143d1112c7732/pandas-3.0.5-cp314-cp314-macos
x_11_0_arm64.whl", hash = "sha256:53730687fcd161883b24e10411c06d6a4c0f2275d2faf3bb2bc25deb4ba8007c", size = 9980065, upload-time = "2026-07-22T22:18:5
2.249Z" },
    { url = "https://files.pythonhosted.org/packages/ea/f1/598503ce8d7e3c35601e0747ba288c7864baae66380725bc12f13f884dfe/pandas-3.0.5-cp314-cp314-manyl
inux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:960d3ebcf249f75206899fcd2c6de53f736b7265759ced0d3e559df0b8b709b0", size = 10545532, uplo
ad-time = "2026-07-22T22:18:54.813Z" },
    { url = "https://files.pythonhosted.org/packages/fa/de/ceae2adf7034e07e9910299fe412e1819c4f0dd520700a888bcb03625448/pandas-3.0.5-cp314-cp314-manyl
inux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9e94c2c5ca43bd3ca32bf64d32308887b65e5f9bfd8023ea52755107a999f93b", size = 10963120, upload
-time = "2026-07-22T22:18:57.42Z" },
    { url = "https://files.pythonhosted.org/packages/66/25/86e0f4451874eb79e688deeebe3c451fec4557f8952005818d800ee8ac7e/pandas-3.0.5-cp314-cp314-musll
inux_1_2_aarch64.whl", hash = "sha256:e819dd5f62966b481a8cb649d3299ebd886a1ea91ed5a99bf7ce77c98d18ab94", size = 11563178, upload-time = "2026-07-22T22
:18:59.729Z" },
    { url = "https://files.pythonhosted.org/packages/f3/45/8643daa3b4147e433adfcccefdd0380d3aad79d86b15d8999730fe1944d5/pandas-3.0.5-cp314-cp314-musll
inux_1_2_x86_64.whl", hash = "sha256:3c5ed2e7c06e91d340dfd091d7934f9bc82e4a36b95f647f090b9d1c9ac649da", size = 12028708, upload-time = "2026-07-22T22:
19:02.164Z" },
    { url = "https://files.pythonhosted.org/packages/96/58/ad979ae617615576e8aafd569c9d4b62f1191d896e38f51d66ba06f3b89a/pandas-3.0.5-cp314-cp314-win_a
md64.whl", hash = "sha256:cd8f7c6dc98527058ee6264219343f5392240a6f1bfa654fc5d79023020d0c92", size = 9951806, upload-time = "2026-07-22T22:19:04.596Z"
},
    { url = "https://files.pythonhosted.org/packages/69/32/7ac03886b304049a9d2625ee88f59af760d8a93bd30ed9239bce7b9869a8/pandas-3.0.5-cp314-cp314-win_a
rm64.whl", hash = "sha256:5183427f5a8156d480f30333777bc978be93650a49a7c01db26adffe95b31e85", size = 9238297, upload-time = "2026-07-22T22:19:06.836Z"
},
    { url = "https://files.pythonhosted.org/packages/be/ed/1d1f2ee5547d5167face2376d11c8b2a4c7bfff5a416ee7a9046891fab1e/pandas-3.0.5-cp314-cp314t-maco
sx_10_15_x86_64.whl", hash = "sha256:303da736987d481074ca720ada325f8bd80c64ebc2d45ed79b29df3aaa4a26ca", size = 10849690, upload-time = "2026-07-22T22:
19:09.391Z" },
    { url = "https://files.pythonhosted.org/packages/57/55/17e17152e98fbb0c4b1e562bc65387a2f20a80db0f4a86bf8d3a0e4248d4/pandas-3.0.5-cp314-cp314t-maco
sx_11_0_arm64.whl", hash = "sha256:3b2801bbb049d0136f6c213eae02b5fca969384fc2064dd728d8620552aa49da", size = 10509945, upload-time = "2026-07-22T22:19
:11.773Z" },
    { url = "https://files.pythonhosted.org/packages/88/90/817d44dbf83facf9556f33576d9af0a241981e7bb5c00606c0bcb5df8dda/pandas-3.0.5-cp314-cp314t-many
linux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:cce3a9d11d2b1f82c69a27ec1f4948a170e2c403c4bbfa8cca62e3fdebe2ef3a", size = 10392197, upl
oad-time = "2026-07-22T22:19:14.024Z" },
    { url = "https://files.pythonhosted.org/packages/f1/da/889f00c0a6f5aa1545add70abbf01502dff87ab577adb855bd631c54d2f2/pandas-3.0.5-cp314-cp314t-many
linux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:ef01af4d8dc6cd2c8d6c7736f149574ef93fe043811eeb5e445f2647154b5040", size = 10862726, uploa
d-time = "2026-07-22T22:19:16.351Z" },
    { url = "https://files.pythonhosted.org/packages/bc/98/f1e934fb3c98fce859c6147c6785816c7b5b9ab7821115c5d8c4de9842b9/pandas-3.0.5-cp314-cp314t-musl
linux_1_2_aarch64.whl", hash = "sha256:e2759e890db96dfcffdbd9b86c3c2cb6afaf58def482820317e06163ec1066cd", size = 11414864, upload-time = "2026-07-22T2
2:19:18.981Z" },
    { url = "https://files.pythonhosted.org/packages/fe/be/d448af7d657d82e1888dd8551f79c6d6fb161080b5b9752d84d910ec2319/pandas-3.0.5-cp314-cp314t-musl
linux_1_2_x86_64.whl", hash = "sha256:b58b1b39d46a5862e3fb18f50d1a201398619d16a0f9f73f57eea5583cf0e63c", size = 11925105, upload-time = "2026-07-22T22
:19:21.515Z" },
    { url = "https://files.pythonhosted.org/packages/29/c1/ccb4238212c8c4f496c584f3044d94e0c030ed8e1d68999db46c91c2242f/pandas-3.0.5-cp314-cp314t-win_
amd64.whl", hash = "sha256:1c10461f6eeb35d8f05b6184c65c8b9991663b66c46b1d559b682cb34ae7c6ea", size = 10387612, upload-time = "2026-07-22T22:19:24.257Z
" },
    { url = "https://files.pythonhosted.org/packages/d2/cf/6a51b2c38980e04c279fd2fa908a1b0982064e860444acfca4ec2e2c8359/pandas-3.0.5-cp314-cp314t-win_
arm64.whl", hash = "sha256:3c5015fd1730fbf883647e88068176c839c102cea883ba1769a6f4593bfc1f8c", size = 9509776, upload-time = "2026-07-22T22:19:26.694Z"
 },
]

[[package]]
name = "pathspec"
version = "1.1.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/5a/82/42f767fc1c1143d6fd36efb827202a2d997a375e160a71eb2888a925aac1/pathspec-1.1.1.tar.gz", ha
sh = "sha256:17db5ecd524104a120e173814c90367a96a98d07c45b2e10c2f3919fff91bf5a", size = 135180, upload-time = "2026-04-27T01:46:08.907Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/f1/d9/7fb5aa316bc299258e68c73ba3bddbc499654a07f151cba08f6153988714/pathspec-1.1.1-py3-none-any.wh
l", hash = "sha256:a00ce642f577bf7f473932318056212bc4f8bfdf53128c78bbd5af0b9b20b189", size = 57328, upload-time = "2026-04-27T01:46:07.06Z" },
]
```

```
[[package]]
name = "pgvector"
version = "0.5.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a7/ec/6eb80aebc728200f95229219882994c1b0585b956ca47da5edb9d062627a/pgvector-0.5.0.tar.gz", ha
sh = "sha256:07a9dcf735696879406983afc6eba9a787cef7c0cf6c367ca1a5779f036dee74", size = 35170, upload-time = "2026-07-06T18:27:27.767Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/5c/e4/a5573f2c579ca9ad133293bfb624148ba0893674ca4a6eeec85ced9a6a09/pgvector-0.5.0-py3-none-any.wh
l", hash = "sha256:fedc9800894e6da2be51358d7b7c574bf34f247ca741a5a09513622135f5964f", size = 30958, upload-time = "2026-07-06T18:27:26.797Z" },
]

[[package]]
name = "pillow"
version = "12.3.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/1c/3d/bb7fca845737cf9d7dbde16ed1843984665ff2e0a518f5db43e77ec540b9/pillow-12.3.0.tar.gz", has
h = "sha256:3b8182a766685eaa002637e28b4ec8d6b18819a0c71f579bf0dbaa5830297cce", size = 47025035, upload-time = "2026-07-01T11:56:38.965Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/37/bf/fb3ebff8ddcb76aac5a01389251bbbb9519922a9b520d8247c1ca864a25d/pillow-12.3.0-cp312-cp312-maco
sx_10_13_x86_64.whl", hash = "sha256:ba09209fbe443b4acccebe845d8a138b89a8f4fbaeedd44953490b5315d5e965", size = 5345969, upload-time = "2026-07-01T11:5
4:06.397Z" },
    { url = "https://files.pythonhosted.org/packages/d8/66/9a386a92561f402389a4fc70c18838bf6d35eb5eb5c6850b4b2dc64f5048/pillow-12.3.0-cp312-cp312-maco
sx_11_0_arm64.whl", hash = "sha256:ffd0c5368496f41b0944be820fcb7a838aa6e623d250b01acf2643939c3f99d7", size = 4780323, upload-time = "2026-07-01T11:54:
09.351Z" },
    { url = "https://files.pythonhosted.org/packages/25/27/ac8f99618ffd3dde21db0f4d4b1d2ab00c0880595bfd17df103f7f39fd0c/pillow-12.3.0-cp312-cp312-many
linux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:d9c7f76c0673154f044e9d78c8655fb4213f6ca31a836df48b40fe5d187717b9", size = 6266838, uplo
ad-time = "2026-07-01T11:54:11.71Z" },
    { url = "https://files.pythonhosted.org/packages/84/21/a35af28dcc61f37ed850a2d64c65c701321dfbf25085e469d5559360cbbf/pillow-12.3.0-cp312-cp312-many
linux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:78cb2c6865a35ab8ff8b75fd122f6033b92a62c82801110e48ddd6c936a45d91", size = 6940830, upload
-time = "2026-07-01T11:54:13.732Z" },
    { url = "https://files.pythonhosted.org/packages/eb/51/8b08617af3ad95e33ce6d7dd2c99ed6c8298f7fb131636303956be022e25/pillow-12.3.0-cp312-cp312-musl
linux_1_2_aarch64.whl", hash = "sha256:e491916b378fba47242221bb9ead245211b70d504f495d105d17b14a24b4907c", size = 6344383, upload-time = "2026-07-01T11
:54:15.756Z" },
    { url = "https://files.pythonhosted.org/packages/1d/72/cf78ac9780bb93c28328f408973845a309d4d145041665f734572ced1b52/pillow-12.3.0-cp312-cp312-musl
linux_1_2_x86_64.whl", hash = "sha256:0dd2064cbc55aaec028ef5fbb60fa47bb6c3e7918e07ff17935284b227a9d2df", size = 7052934, upload-time = "2026-07-01T11:
54:17.721Z" },
    { url = "https://files.pythonhosted.org/packages/20/20/25e0f4dc178a6bc0696793720055519a0de89e7661dae886992decbd2f81/pillow-12.3.0-cp312-cp312-win3
2.whl", hash = "sha256:dbce0b29841537a2fa4a214c2bbf14de3587c9680caa9b4e217568472490b28f", size = 6472684, upload-time = "2026-07-01T11:54:19.839Z" },
    { url = "https://files.pythonhosted.org/packages/45/89/da2f7971a317f83d807fdd4065c0af40208e59e692cc43d315a71a0e96d1/pillow-12.3.0-cp312-cp312-win_
amd64.whl", hash = "sha256:a2b55dd6b2a4c4b7d87ffa56bdb33fdc5fdb9a462173861a7bc097f17d91cb09", size = 7227137, upload-time = "2026-07-01T11:54:22.025Z"
 },
    { url = "https://files.pythonhosted.org/packages/de/47/4845a0a6c0dbf1db8456bd9fc791f13c5ced7ced20606d08a0aacfd25b49/pillow-12.3.0-cp312-cp312-win_
arm64.whl", hash = "sha256:331b624368d4f1d069149002f25f44bc61c8919ce8ddb3c45bdad8f6e2d89510", size = 2568267, upload-time = "2026-07-01T11:54:24.051Z"
 },
    { url = "https://files.pythonhosted.org/packages/9d/ac/31fb64e1e7efb5a4b50cd3d92049ba89ac6e4d8d3bb6a74e15048ca3353e/pillow-12.3.0-cp313-cp313-ios_
13_0_arm64_iphoneos.whl", hash = "sha256:21900ce7ba264168cd50defae43cd75d25c833ad4ad6e73ffc5596d12e25ac89", size = 4161684, upload-time = "2026-07-01T
11:54:25.934Z" },
    { url = "https://files.pythonhosted.org/packages/87/b4/9805e23d2b4d77842b468513841fda254ee42f0289d25088340e4ff46e2d/pillow-12.3.0-cp313-cp313-ios_
13_0_arm64_iphonesimulator.whl", hash = "sha256:4e8c2a84d977f50b9daed6eeaf3baef67d00d5d74d932288f02cb94518ee3ace", size = 4255487, upload-time = "2026
-07-01T11:54:27.935Z" },
    { url = "https://files.pythonhosted.org/packages/df/39/ecf519435a200c693fe053a6ee4d835b41cf963a4dfc2551c4e637cb2a71/pillow-12.3.0-cp313-cp313-ios_
13_0_x86_64_iphonesimulator.whl", hash = "sha256:ae26d61dfa7a47befdc7572b521024e8745f3d809bd95ca9505a7bba9ef849ec", size = 3696433, upload-time = "202
6-07-01T11:54:29.813Z" },
    { url = "https://files.pythonhosted.org/packages/42/92/2fc3ffad878ae8dd5469ec1bc8eb83b71f48e13efdf68f02709003982a32/pillow-12.3.0-cp313-cp313-maco
sx_10_13_x86_64.whl", hash = "sha256:7a743ff716f746fc19a9557f60dab1600d4613255f8a7aeb3cdde4db7eb15a66", size = 5345889, upload-time = "2026-07-01T11:5
4:31.97Z" },
    { url = "https://files.pythonhosted.org/packages/10/76/8803c13605b763d33d156c4678fc77f8443389c0c51c8aef707bb02015f4/pillow-12.3.0-cp313-cp313-maco
sx_11_0_arm64.whl", hash = "sha256:d69141514cc30b774ceea5e3ed3a6635c8d8a96edf664689b890f4089111fb35", size = 4780109, upload-time = "2026-07-01T11:54:
34.026Z" },
    { url = "https://files.pythonhosted.org/packages/1f/01/e18aff37cb0b4aac47ac90f016d347a49aca667ef97f190b06ac2aabc928/pillow-12.3.0-cp313-cp313-many
linux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:f7401aebd7f581d7f83a439d87d474999317ee099218e5ad25d125290990ba65", size = 6263736, uplo
ad-time = "2026-07-01T11:54:36.131Z" },
    { url = "https://files.pythonhosted.org/packages/f7/62/de5bdd77d935331f4f802edc11e4d82950f642caad6cb2f949837b8560e2/pillow-12.3.0-cp313-cp313-many
linux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:0847a763afefb695bc912d7c131e7e0632d4edc1d8698f58ddabec8e46b8b6d3", size = 6937129, upload
-time = "2026-07-01T11:54:38.216Z" },
    { url = "https://files.pythonhosted.org/packages/70/4d/105627a13300c5e0df1d174230b32fd1273062c96f7745fd552b945d1e1d/pillow-12.3.0-cp313-cp313-musl
linux_1_2_aarch64.whl", hash = "sha256:571b9fcb07b97ef3a492028fb3d2dc0993ca23a06138b0315286566d29ef718a", size = 6339562, upload-time = "2026-07-01T11
:54:40.354Z" },
    { url = "https://files.pythonhosted.org/packages/6b/1d/f13de01a553988ab895ba1c722e06cf3144d4f57656fd5b81b6d881f1179/pillow-12.3.0-cp313-cp313-musl
linux_1_2_x86_64.whl", hash = "sha256:756c768d0c9c2955feb7a56c37ea24aea2e369f8d36a88da270b6a9f19e62b5e", size = 7049439, upload-time = "2026-07-01T11:
54:42.489Z" },
    { url = "https://files.pythonhosted.org/packages/c9/f9/066794cca041b969964f779ee5fa66a9498bbf34248ac39c5d7954e4198f/pillow-12.3.0-cp313-cp313-win3
2.whl", hash = "sha256:a876864214e136f0eb367788dbd7df045f4806801518e2cfe9e13229cfe06d8f", size = 6473287, upload-time = "2026-07-01T11:54:44.9Z" },
    { url = "https://files.pythonhosted.org/packages/a6/9b/7a58e61d62be561da3a356fe2384d4059a6345fc130e23ef1c36a5b81d24/pillow-12.3.0-cp313-cp313-win_
amd64.whl", hash = "sha256:1cca606cd25738df4ed873d5ad46bbdb3d83b5cbca291f6b4ff13a4df6b0bbe8", size = 7239691, upload-time = "2026-07-01T11:54:47.141Z"
 },
    { url = "https://files.pythonhosted.org/packages/aa/b0/c4ed4f0ef8f8fa5ee8351537db6650bb8189f7e118842978dd6589065692/pillow-12.3.0-cp313-cp313-win_
arm64.whl", hash = "sha256:b629de27fda84b42cde7edef0d85f13b958b47f6e9bbcbba9b673c562a89bd8b", size = 2568185, upload-time = "2026-07-01T11:54:49.137Z"
 },
    { url = "https://files.pythonhosted.org/packages/dc/01/001f65b68192f0228cc1dbbc8d2530ab5d58b61037ba0587f946fea607cd/pillow-12.3.0-cp314-cp314-ios_
13_0_arm64_iphoneos.whl", hash = "sha256:9cf95fe4d0f84c82d282745d9bb08ad9f926efa00be4697e767b814ce40d4330", size = 4161736, upload-time = "2026-07-01T
11:54:51.156Z" },
    { url = "https://files.pythonhosted.org/packages/1a/d2/0219746d0fd16fc8a84498e79452375be3797d3ce4044596ce565164b84f/pillow-12.3.0-cp314-cp314-ios_
13_0_arm64_iphonesimulator.whl", hash = "sha256:8728f216dcdb6e6d555cf971cb34076139ad74b31fc2c14da4fafc741c5f6217", size = 4255435, upload-time = "2026
-07-01T11:54:53.414Z" },
    { url = "https://files.pythonhosted.org/packages/c8/02/8d0bc62ef0302318c46ff2a512822d2610e81c7aa46c9b3abe6cbaca5ad0/pillow-12.3.0-cp314-cp314-ios_
13_0_x86_64_iphonesimulator.whl", hash = "sha256:a45650e8ce7faffd731db8550230db6b0d306d181a90b67d3e6bca2f1990930", size = 3696262, upload-time = "202
6-07-01T11:54:55.739Z" },
    { url = "https://files.pythonhosted.org/packages/85/e2/73c77d218410b14f5f2d565e8a998d5317b7b9c75368d29985139f7a46f0/pillow-12.3.0-cp314-cp314-maco
sx_10_15_x86_64.whl", hash = "sha256:ba54cfebe86920a559a7c4d6b9050791c20513650a1952ebe3368c7dc70306f8", size = 5350344, upload-time = "2026-07-01T11:5
4:57.657Z" },
    { url = "https://files.pythonhosted.org/packages/c7/da/32c752228ae345f489e3a42499d817b6c3996da7e8a3bc7a04fc806b243b/pillow-12.3.0-cp314-cp314-maco
sx_11_0_arm64.whl", hash = "sha256:e158cb00350dc278f3b91551101aa7d12415a66ebf2c91d8d5ac14e56ddd3ad0", size = 4780131, upload-time = "2026-07-01T11:54:
59.713Z" },
    { url = "https://files.pythonhosted.org/packages/b1/9d/8b2c807dbef61a5197c047afe99823787eb66f63daf9fb2432f91d6f0462/pillow-12.3.0-cp314-cp314-many
linux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:e9aeb04d6aef139de265b29683e119b638208f88cf73cdd1658aa07221165321", size = 6263757, uplo
ad-time = "2026-07-01T11:55:01.778Z" },
    { url = "https://files.pythonhosted.org/packages/5c/44/c85361f65dbe00eea8576ee467c768d25129989efb76e94f205e9ca9bb46/pillow-12.3.0-cp314-cp314-many
linux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:251bf95b67017e27b13d82f5b326234ca62d70f9cf4c2b9032de2358a3b12c7b", size = 6936962, upload
-time = "2026-07-01T11:55:03.93Z" },
    { url = "https://files.pythonhosted.org/packages/18/7e/e483414b35800b86b6f08dbbc7803fb5cd52c4d6f897f47d53ea2c7e6f65/pillow-12.3.0-cp314-cp314-musl
linux_1_2_aarch64.whl", hash = "sha256:fe3cca2e4e8a592be0f269a1ca4835c25199d9f3ce815c8491048f785b0a0198", size = 6339171, upload-time = "2026-07-01T11
:55:05.989Z" },
    { url = "https://files.pythonhosted.org/packages/f0/f4/68c491844841ede6bed70189546b3ee9731cf9f2cbad396faff5e1ccba45/pillow-12.3.0-cp314-cp314-musl
linux_1_2_x86_64.whl", hash = "sha256:23aceaa007d6172b02c277f0cd359c79492bbb14f7072b4ede9fbcaf20648130", size = 7048116, upload-time = "2026-07-01T11:
```

55:08.131Z" },
    { url = "https://files.pythonhosted.org/packages/a3/34/77f3f793fed8efc7d243f21b33c5a3f0d1c97ee70346d3db855587e155ff/pillow-12.3.0-cp314-cp314-win32.whl", hash = "sha256:af8d94b0db561cf68b88a267c5c44b49e134f525d0dc2cb7ed413a66bc23559a", size = 6467209, upload-time = "2026-07-01T11:55:10.408Z" },
    { url = "https://files.pythonhosted.org/packages/f1/e0/492879f69d94f91f60fc8cd05ba03650e9520afebb2fb7aa12777d7c7f38/pillow-12.3.0-cp314-cp314-win_amd64.whl", hash = "sha256:fdafc9cce40277e0f7a0feabce0ee50dd2fa1800f3b38015e51296b5e814048d", size = 7237707, upload-time = "2026-07-01T11:55:12.745Z" },
    { url = "https://files.pythonhosted.org/packages/c9/ac/6b11f2875f1c2ac040d84e1bbf9cf22a88038f901ca1037898b280b38365/pillow-12.3.0-cp314-cp314-win_arm64.whl", hash = "sha256:e91206ee562682b51b98ef4b26a6ef48fd84e15fd4c4bc5ec768eb641d206838", size = 2565995, upload-time = "2026-07-01T11:55:14.736Z" },
    { url = "https://files.pythonhosted.org/packages/52/69/c2208e56af9bfc1913afb24020297a691eb1d4ef688474c8a04913f65e04/pillow-12.3.0-cp314-cp314t-macosx_10_15_x86_64.whl", hash = "sha256:164b31cd1a0490ab6efae01aa5df49da7061be0af1b30e035b6e9a1bfe34ee6e", size = 5352503, upload-time = "2026-07-01T11:55:17.076Z" },
    { url = "https://files.pythonhosted.org/packages/07/70/e5686d753e898a45d778ff1718dba8516ead6ab6b95d85fc8c4b70650cf2/pillow-12.3.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:5afb51d599ea772b8365ae807ae557f18bccfe46ab261fd1c2a9ed700fc6eb17", size = 4782956, upload-time = "2026-07-01T11:55:19.448Z" },
    { url = "https://files.pythonhosted.org/packages/d5/37/25c6692f06927ee973ff18c8d9ee98ad0b4d84ee67a09610c2dd1447958e/pillow-12.3.0-cp314-cp314t-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:3edce1d53195db527e0191f84b71d02022de0540bf43a16ed734ed7537b07385", size = 6322855, upload-time = "2026-07-01T11:55:21.613Z" },
    { url = "https://files.pythonhosted.org/packages/cc/91/420637fcb8f1bc11029e403b4538e6694744428d8246118e45719f944556/pillow-12.3.0-cp314-cp314t-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:bf16ba1b4d0b6b7c8e534936632270cf70eb00dbe09005bc345b2677b726855c", size = 6989642, uploaded-time = "2026-07-01T11:55:24.006Z" },
    { url = "https://files.pythonhosted.org/packages/10/08/b94d7811281ccf0d143a1cf768d1c49e1e54af63e7b708ab2ee3eb87face/pillow-12.3.0-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:24870b09b224f7ae3c39ed07d10e819d06f8720bc551847b1d623832b5b0e28d", size = 6391281, upload-time = "2026-07-01T11:55:26.252Z" },
    { url = "https://files.pythonhosted.org/packages/d2/87/24233f785f55474dc02ce3e739c5528a77e3a862e9333d1dd7a25cc31f70/pillow-12.3.0-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:30f2aa603c41533cc25c05acd0da21636e84a315768feb631c937177db558931", size = 7096716, upload-time = "2026-07-01T11:55:28.318Z" },
    { url = "https://files.pythonhosted.org/packages/23/26/fcb2f6e37175b04f53570b59937867e2b80ee1685e744023153028fc14f9/pillow-12.3.0-cp314-cp314t-win32.whl", hash = "sha256:4b0a7fe987b14c31ebda6083f74f22b561fd3739bc0ac51e019622e3d72668c7", size = 6474125, upload-time = "2026-07-01T11:55:30.956Z" },
    { url = "https://files.pythonhosted.org/packages/90/de/3634abee5f1c9e13c56787b7d5517b0ba8d6de51700b95578cf338349c9f/pillow-12.3.0-cp314-cp314t-win_amd64.whl", hash = "sha256:962864dc93511324d51ddbb5b9f8731bf71675b93ca612a07441896f4688fb8c", size = 7242939, upload-time = "2026-07-01T11:55:34.044Z" },
    { url = "https://files.pythonhosted.org/packages/ce/2a/fd13f8eb24de5714a6eb444a3d67e2842c6c576e159a43793adf23051351/pillow-12.3.0-cp314-cp314t-win_arm64.whl", hash = "sha256:0740a512dc522224c77d9aa5a8d70d8b7d73fb91f2c21125d8d025d3b8990e45", size = 2567506, upload-time = "2026-07-01T11:55:35.988Z" },
    { url = "https://files.pythonhosted.org/packages/5d/dc/8fdce34ec725a33c81c6ba122b904d6b9024e50ea9ac7bede62fab54506c/pillow-12.3.0-cp315-cp315-ios_13_0_arm64_iphoneos.whl", hash = "sha256:0feb2e9d6ad6c9e3c06effe9d00f3f1e618a6643273576b016f591e9315a7139", size = 4162063, upload-time = "2026-07-01T11:55:37.941Z" },
    { url = "https://files.pythonhosted.org/packages/76/66/2044b9a63d3b84ff048228dfcb7cd9bf0df983e8470971bf7d4c57b693de/pillow-12.3.0-cp315-cp315-ios_13_0_arm64_iphonesimulator.whl", hash = "sha256:9e881fca225083806662a5c43d627d215f258ff43c890f831966c7d7ba9c7402", size = 4255549, upload-time = "2026-07-01T11:55:40.022Z" },
    { url = "https://files.pythonhosted.org/packages/52/7e/1f67e6f4ece6b582ee4b539decbcc9f848dc245a93ed8cd7338bafef72f1/pillow-12.3.0-cp315-cp315-ios_13_0_x86_64_iphonesimulator.whl", hash = "sha256:4998562bf62a445225f22e07c896bb04b35b1b1f2eb6d760584c9c51d7a5f78c", size = 3696331, upload-time = "2026-07-01T11:55:41.98Z" },
    { url = "https://files.pythonhosted.org/packages/12/40/d306fc2c8e4d45d7f175c77edca7063be7b86fe7fe6e68f4353bf71d808c/pillow-12.3.0-cp315-cp315-macosx_10_15_x86_64.whl", hash = "sha256:dc624f6bc473dacdf7ef7eb8678d0d08edf15cd94fad6ae5c7d6cc67a4e4902f", size = 5350370, upload-time = "2026-07-01T11:55:44.028Z" },
    { url = "https://files.pythonhosted.org/packages/dd/44/668fb1437e8ce420f62d6106eb66e44a5971602a4d794615bdf79315d82d/pillow-12.3.0-cp315-cp315-macosx_11_0_arm64.whl", hash = "sha256:71d6097b330eea8fd15097780c8e89cb1a8ce7838669f48c5bacd6f663dd4701", size = 4780147, upload-time = "2026-07-01T11:55:46.073Z" },
    { url = "https://files.pythonhosted.org/packages/0c/08/93fa2e70e30a2d81547e481b6ee2bb9522117221fb1e0ce4b5df70967677/pillow-12.3.0-cp315-cp315-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:28ce87c5ab450a9dd970b52e5aca5fe63ed432d18a2eaddd1979a00a1ba24ace", size = 6273659, upload-time = "2026-07-01T11:55:48.264Z" },
    { url = "https://files.pythonhosted.org/packages/f8/6d/043e96ff814fc31a33077e4cba86082167db520c93632afdf2042febbb0c/pillow-12.3.0-cp315-cp315-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:6b02afb9b97f65fbca5f31db6a2a3ba21aa93030225f150fa3f249717e938fb4", size = 6947439, upload-time = "2026-07-01T11:55:50.503Z" },
    { url = "https://files.pythonhosted.org/packages/af/92/ba71d2ee2ac0edf3fa33bd9d5ee9ee080da70b1766f3ca3934f9938ddac9/pillow-12.3.0-cp315-cp315-musllinux_1_2_aarch64.whl", hash = "sha256:1182d52bc2d5e5d7d0949503aa7e36d12f42205dc287e4883f407b1988820d39", size = 6353577, upload-time = "2026-07-01T11:55:52.697Z" },
    { url = "https://files.pythonhosted.org/packages/0f/ce/e63064e2122923ff687c8ad792d0d736a7b3920a56a46982e81a7fdd25d6/pillow-12.3.0-cp315-cp315-musllinux_1_2_x86_64.whl", hash = "sha256:e795b7eb908249c4e43c7c99fac7c2c75dab0c43566e37db472a355f63693d71", size = 7060394, upload-time = "2026-07-01T11:55:55.149Z" },
    { url = "https://files.pythonhosted.org/packages/54/76/a09cc3ccc8d773a7283d34c38bec1708f9e3cc932093cbc4c5e71ac4060b/pillow-12.3.0-cp315-cp315-win32.whl", hash = "sha256:57b3d78c95ba9059768b10e28b813002261d3f3dfc55cc48b0c988f625175827", size = 6467375, upload-time = "2026-07-01T11:55:57.769Z" },
    { url = "https://files.pythonhosted.org/packages/3e/03/1846c49ba3b1d5550392a4bbd06d6fb4578e1cd91a803198b5c90f5f7d53/pillow-12.3.0-cp315-cp315-win_amd64.whl", hash = "sha256:fa4ecea169a355be7a3ade2c783e2ed12f0e40d2c5621cda8b3297faf7fbb9f5", size = 7237048, upload-time = "2026-07-01T11:55:59.975Z" },
    { url = "https://files.pythonhosted.org/packages/fb/bb/89f35dcc79610423f9f195504d7def7f0d1416a711541b42867e25fe3412/pillow-12.3.0-cp315-cp315-win_arm64.whl", hash = "sha256:877c3f311ff35410f690861c4409e7ccbf0cd2f878e50628a28e5a0bb689e658", size = 2566006, upload-time = "2026-07-01T11:56:02.143Z" },
    { url = "https://files.pythonhosted.org/packages/30/88/707027ba09942dfa2c28759b5c222d769290a41c6d20ea60ec250801941f/pillow-12.3.0-cp315-cp315t-macosx_10_15_x86_64.whl", hash = "sha256:e9871b1ffbfa9656b60aeee92ed5136a5742696006fa322b29ea3d8da0ecc9cf", size = 5352509, upload-time = "2026-07-01T11:56:04.2Z" },
    { url = "https://files.pythonhosted.org/packages/b0/6d/00352fa25332c2569cd387851f568cc5a4b75a9adbfb37ac4fbce4c02eec/pillow-12.3.0-cp315-cp315t-macosx_11_0_arm64.whl", hash = "sha256:53aa02d20d10c3d814d536aa4e5ac9b84ca0ff5a88377963b085ad6822f93e64", size = 4783167, upload-time = "2026-07-01T11:56:06.631Z" },
    { url = "https://files.pythonhosted.org/packages/13/4f/9e049dfa21af7c22427275720e2490267ba8138120add5c4c574deb69782/pillow-12.3.0-cp315-cp315t-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:446c34dcc4324b084a53b705127dc15717b22c5e140ae0a3c38349d4efec071e", size = 6329237, upload-time = "2026-07-01T11:56:08.868Z" },
    { url = "https://files.pythonhosted.org/packages/36/16/cf6eeaae8d0fce8dd390a33437cf68c5d5bd73834a2bc6e2f14efda0ab45/pillow-12.3.0-cp315-cp315t-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:cf1845d02ad822a369a49f2bb9345b1614744267682e7a03527dc3bf6eea1777", size = 6997047, upload-time = "2026-07-01T11:56:11.379Z" },
    { url = "https://files.pythonhosted.org/packages/1e/69/dbf769bdd55f48bf5733cac28edc6364ffaa072ec9ba336266e4fe66be55/pillow-12.3.0-cp315-cp315t-musllinux_1_2_aarch64.whl", hash = "sha256:186941b6aef820ad110fb01fb06eb925374dc3a21b17e37ec9a53b250c6fe2d1", size = 6400440, upload-time = "2026-07-01T11:56:13.908Z" },
    { url = "https://files.pythonhosted.org/packages/a0/e1/ffc9cfc2eea0d178da8018e18e959301ad9d6bc9f3edb7181e748a474b97/pillow-12.3.0-cp315-cp315t-musllinux_1_2_x86_64.whl", hash = "sha256:f13c32a3abd6079a66d9526e18dad9b6d280384d49d7c54040cd57b6424041d9", size = 7105895, upload-time = "2026-07-01T11:56:16.575Z" },
    { url = "https://files.pythonhosted.org/packages/18/f0/a5595c1e8c3ae44b9828cb2f0fa8155e5095ef04d6327b8f61cf44a3df85/pillow-12.3.0-cp315-cp315t-win32.whl", hash = "sha256:1657923d2d45afb66526e5b933e5b3052e6bdea196c90d3abb2424e18c77dae8", size = 6474384, upload-time = "2026-07-01T11:56:18.855Z" },
    { url = "https://files.pythonhosted.org/packages/e4/04/62bcd9f844984c5938d3b05264a61d797a29d3e0812341a8204af70bbdee/pillow-12.3.0-cp315-cp315t-win_amd64.whl", hash = "sha256:8cd2f7bdda092d99c9fc2fb7391354f306d01443d22785d0cbfafa2e2c8bb418", size = 7243537, upload-time = "2026-07-01T11:56:21.214Z" },
    { url = "https://files.pythonhosted.org/packages/3d/68/1f3066acedf37673694a7141381d8f811ae97f30d34413d236abe7d489f1/pillow-12.3.0-cp315-cp315t-win_arm64.whl", hash = "sha256:06ff022112bc9cbf83b60f8e028d94ad87b60621706487e65f673de61610ab59", size = 2567491, upload-time = "2026-07-01T11:56:23.506Z" },
]

[[package]]
name = "platformdirs"
version = "4.11.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/78/9b/560e4be8e26f6fd133a03630a8df0c663b9e8d61b4ade152b72005aec83b/platformdirs-4.11.0.tar.gz", hash = "sha256:0555d18370482847566ffabcaa53ad7c6c1c29f195989ae1ed634a05f76ea1e0", size = 31953, upload-time = "2026-07-21T13:09:36.565Z" }
wheels = [

```
    { url = "https://files.pythonhosted.org/packages/7d/68/d8d58938dfb1370b266a1a729e6d77a985be23689a0496498ee17b2cbf90/platformdirs-4.11.0-py3-none-any.whl", hash = "sha256:360ccded2b7fce0af0ff80cc8f5942a1c5d99b0e856033acb030bfc634709e74", size = 23247, upload-time = "2026-07-21T13:09:35.422Z" },
]

[[package]]
name = "pluggy"
version = "1.6.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/f9/e2/3e91f31a7d2b083fe6ef3fa267035b518369d9511ffab804f839851d2779/pluggy-1.6.0.tar.gz", hash = "sha256:7dcc130b76258d33b90f61b658791dede3486c3e6bfb003ee5c9bfb396dd22f3", size = 69412, upload-time = "2025-05-15T12:30:07.975Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/54/20/4d324d65cc6d9205fabedc306948156824eb9f0ee1633355a8f7ec5c66bf/pluggy-1.6.0-py3-none-any.whl", hash = "sha256:e920276dd6813095e9377c0bc5566d94c932c33b27a3e3945d8389c374dd4746", size = 20538, upload-time = "2025-05-15T12:30:06.134Z" },
]

[[package]]
name = "prometheus-client"
version = "0.26.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/52/73/f1334c29c2af4cd9dba6c7817e61b611bd0215e2eb5565c6064a4de18802/prometheus_client-0.26.0.tar.gz", hash = "sha256:04a91bcf94e2cf74a44a1a874d651a2e853ed354b6e822f3b7487751465d5c2b", size = 92910, upload-time = "2026-07-24T19:36:41.893Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/eb/a3/b69efbf4143b5b9859b977770bbbabcc2796b702fa69dc40271e45cd5a56/prometheus_client-0.26.0-py3-none-any.whl", hash = "sha256:fa93d06737aa02bacd05794768508bb97d2fbee28cb3bca04eaae92f0ca953d6", size = 64494, upload-time = "2026-07-24T19:36:40.854Z" },
]

[[package]]
name = "propcache"
version = "0.5.2"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/ec/44/c87281c333769159c50594f22610f77398a47ccbfbbf23074e744e86f87c/propcache-0.5.2.tar.gz", hash = "sha256:01c4fc7480cd0598bb4b57022df55b9ca296da7fc5a8760bd8451a7e63a7d427", size = 50208, upload-time = "2026-05-08T21:02:12.199Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/4a/cb/e27bc2b2737a0bb49962b275efa051e8f1c35a936df7d5139b6b658b7dc9/propcache-0.5.2-cp312-cp312-macosx_10_13_universal2.whl", hash = "sha256:806719138ecd720339a12410fb9614ac9b2b2d3a5fdf8235d56981c36f4039ba", size = 95887, upload-time = "2026-05-08T21:00:11.277Z" },
    { url = "https://files.pythonhosted.org/packages/e6/13/b8ae04c59392f8d11c6cd9fb4011d1dc7c86b81225c770280300e259ffe1/propcache-0.5.2-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:db2b80ea58eab4f86b2beec3cc8b39e8ff9276ac20e96b7cce43c8ae84cd6b5a", size = 54654, upload-time = "2026-05-08T21:00:12.604Z" },
    { url = "https://files.pythonhosted.org/packages/2c/7d/49777a3e20b55863d4794384a38acd460c04157b0a00f8602b0d508b8431/propcache-0.5.2-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:e5cbfac9f61484f7e9f3597775500cd3ebe8274e9b050c38f9525c77c97520bf", size = 55190, upload-time = "2026-05-08T21:00:13.935Z" },
    { url = "https://files.pythonhosted.org/packages/44/c7/085d0cd63062e84044e3f05797749c3f8e3938ff3aeb0eb2f69d43fafc91/propcache-0.5.2-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:5dbc581d2814337da56222fab8dc5f161cd798a434e49bac27930aaef798e144", size = 59995, upload-time = "2026-05-08T21:00:15.526Z" },
    { url = "https://files.pythonhosted.org/packages/9c/42/32cf8e3009e92b2645cf1e944f701e8ea4e924dffde1ee26db860bcbf7e4/propcache-0.5.2-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:857187f381f88c8e2fa2fe56ab94879d011b883d5a2ee5a1b60a8cd2a06846d9", size = 63422, upload-time = "2026-05-08T21:00:16.824Z" },
    { url = "https://files.pythonhosted.org/packages/9e/1b/f112433f99fc979431b87a39ef169e3f8df070d99a72792c56d6937ac48b/propcache-0.5.2-cp312-cp312-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:178b4a2cdaac1818e2bf1c5a99b94383fa73ea5382e032a48dec07dc5668dc42", size = 64342, upload-time = "2026-05-08T21:00:18.362Z" },
    { url = "https://files.pythonhosted.org/packages/14/15/5574111ae50dd6e879456888c0eadd4c5a869959775854e18e18a6b345f3/propcache-0.5.2-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:6f328175a2cde1f0ff2c4ed8ce968b9dcfb55f3a7153f39e2957ed994da13476", size = 61639, upload-time = "2026-05-08T21:00:19.692Z" },
    { url = "https://files.pythonhosted.org/packages/cc/da/4d775080b1490c0ae604acda868bd71aabe3a89ed16f2aa4339eb8a283e7/propcache-0.5.2-cp312-cp312-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:5671d09a36b06d0fd4a3da0fccbcae360e9b1570924171a15e9e0997f0249fba", size = 61588, upload-time = "2026-05-08T21:00:21.155Z" },
    { url = "https://files.pythonhosted.org/packages/04/ac/f076982cbe2195ee9cf32de5a1e46951d9fb399fc207f390562dd0fd8fb2/propcache-0.5.2-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:80168e2ebe4d3ec6599d10ad8f520304ae1cad9b6c5a95372aef1b66b7bfb53a", size = 60029, upload-time = "2026-05-08T21:00:22.713Z" },
    { url = "https://files.pythonhosted.org/packages/70/60/189be62e0dd898dce3b331e1b8c7a543cd3a405ac0c81fe8ee8a9d5d77e1/propcache-0.5.2-cp312-cp312-musllinux_1_2_armv7l.whl", hash = "sha256:45f11346f884bc47444f6e66471310558441 34c3175b629f84952e2b5cd62b64", size = 56774, upload-time = "2026-05-08T21:00:24.001Z" },
    { url = "https://files.pythonhosted.org/packages/ea/9e/93377b9c7939c1ffae98f878dee955efadfd638078bc86dbc21f9d52f651/propcache-0.5.2-cp312-cp312-musllinux_1_2_ppc64le.whl", hash = "sha256:8e778ebd44ef4f66ed60a0416b06b489687db264a9c0b3620362f26489492913", size = 63532, upload-time = "2026-05-08T21:00:25.545Z" },
    { url = "https://files.pythonhosted.org/packages/14/f9/590ef6cfb9b8028d516d287812ece32bb0bc5f11fbb9c8bf6b2e6313fec8/propcache-0.5.2-cp312-cp312-musllinux_1_2_riscv64.whl", hash = "sha256:c0cb9ed24c8964e172768d455a38254c2dd8a55290 5729ce006cad3d3dda59b1", size = 61592, upload-time = "2026-05-08T21:00:27.186Z" },
    { url = "https://files.pythonhosted.org/packages/b4/5e/70958b3034c297a630bba2f17ca7abc2d5f39a803ad7e370ab79d1ecd022/propcache-0.5.2-cp312-cp312-musllinux_1_2_s390x.whl", hash = "sha256:1d1ad32d9d4355e2be65574fd0bfd3677e7066b009cd5b9b2dee8aa6a6393b33", size = 64788, upload-time = "2026-05-08T21:00:28.8Z" },
    { url = "https://files.pythonhosted.org/packages/12/fd/77fe5936d8c3086ca9048f7f415f122ed82e53884a9ec193646b42deef06/propcache-0.5.2-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:c80f4ba3e8f00189165999a742ee526ebeccedf6c3f7beb0c7df821e9772435a", size = 62514, upload-time = "2026-05-08T21:00:30.098Z" },
    { url = "https://files.pythonhosted.org/packages/cf/74/66bd798b5b3be70aa1b391f5cc9d6a0a5532d7fd3b19ec0b213e72e6ad9d/propcache-0.5.2-cp312-cp312-win32.whl", hash = "sha256:8c7972d8f193740d9175f0998ab38717e6cd322d5935c5b0fef8c0d323fd9031", size = 39018, upload-time = "2026-05-08T21:00:31.622Z" },
    { url = "https://files.pythonhosted.org/packages/61/7c/5c0d34aa3024694d6dcb9271cdbdd08c4e47c1c0ad95ec7e7bc74cdea145/propcache-0.5.2-cp312-cp312-win_amd64.whl", hash = "sha256:d9ee8826a7d47863a08ac44e1a5f611a462eefc3a194b492da242128bec75b42", size = 42322, upload-time = "2026-05-08T21:00:32.918Z" },
    { url = "https://files.pythonhosted.org/packages/4d/91/875812f1a3feb20ceba818ef39fbe4d92f1081e04ac815c822496d0d038b/propcache-0.5.2-cp312-cp312-win_arm64.whl", hash = "sha256:2800a4a8ead6b28cccd1ec54b59346f0def7922ee1c7598e8499c733cfbb7c84", size = 38172, upload-time = "2026-05-08T21:00:35.124Z" },
    { url = "https://files.pythonhosted.org/packages/c5/09/f049e45385503fe67db75a6b6186a7b9f0c3930366dc960522c312a825b1/propcache-0.5.2-cp313-cp313-macosx_10_13_universal2.whl", hash = "sha256:099aaf4b4d1a02265b92a977edf00b5c4f63b3b17ac6de39b0d637c9cac0188a", size = 94457, upload-time = "2026-05-08T21:00:36.355Z" },
    { url = "https://files.pythonhosted.org/packages/6b/65/83d1d05655baf63113731bd5a1008435e14f8d1e5a06cbe4ec5b23ad7a31/propcache-0.5.2-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:68ce1c44c7a813a7f71ea04315a8c7b330b63db99d059a797a4651bb6f69f117", size = 53835, upload-time = "2026-05-08T21:00:38.072Z" },
    { url = "https://files.pythonhosted.org/packages/a9/12/a6ba6482bb5ea3260c000c9b20881c95fa11c6b30173715668259f844ed7/propcache-0.5.2-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:fc299c129490f55f254cd90be0deca4764e36e9a7c08b4aa588479a3bbed3098", size = 54545, upload-time = "2026-05-08T21:00:39.319Z" },
    { url = "https://files.pythonhosted.org/packages/a9/19/7fa086f5764c59ec8a8e157cd93aa8497acc00aba9dcdec56bfffb32602d/propcache-0.5.2-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:a6ae2198be502c10f09b2516e7b5d019816924bc3183a43ce792a7bd6625e6f4", size = 59886, upload-time = "2026-05-08T21:00:40.621Z" },
    { url = "https://files.pythonhosted.org/packages/a1/e4/5d7663dc8235956c8f5281698a3af1d351d8820341ddd890f59d9a9127f2/propcache-0.5.2-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:6041d31504dc1779d700e1edcfb08eea334b357620b06681a4eabb57a74e574e", size = 63261, upload-time = "2026-05-08T21:00:41.775Z" },
    { url = "https://files.pythonhosted.org/packages/4a/4a/15a03adee24d6350da4292caeac44c34c033d2afe5e87eb370f38854560f/propcache-0.5.2-cp313-cp313-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:f7eabc04151c78a9f4d5bbb5f1faf571e4defeb4b585e0fe95b60ff2dbe4d3d7", size = 64184, upload-time = "2026-05-08T21:00:43.018Z" },
    { url = "https://files.pythonhosted.org/packages/8b/c6/979176efdaa3d239e36d503d5af63a0a773b36662ed8f52e5b6a6d9fd40e/propcache-0.5.2-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:4db0ba63d693afd40d249bd93f842b5f144f8fcbb83de05660373bcf30517b1d", size = 61534, upload-time = "2026-05-08T21:00:44.507Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/c8/22/63e8cd1bae4c2d2be6493b6b7d10566ddafad88137cfbc99964a1119853c/propcache-0.5.2-cp313-cp313-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:1dbcf7675229b35d31abb6547d8ebc8c27a830ac3f9a794edff6254873ec7c0a", size = 61500, upload-time = "2026-05-08T21:00:45.796Z" },
    { url = "https://files.pythonhosted.org/packages/60/5a/28e5d9acbac1cc9ccb67045e8c1b943aa8d79fdf39c93bd73cacd68008ea/propcache-0.5.2-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:d310c013aad2c72f1c3f2f8dd3279d460a858c551f97aeb8c63e4693cca7b4d2", size = 59994, upload-time = "2026-05-08T21:00:47.093Z" },
    { url = "https://files.pythonhosted.org/packages/f3/40/db650677f554a95b9c01a7c9d93d629e93a15562f5deb4573c9ee136fed2/propcache-0.5.2-cp313-cp313-musllinux_1_2_armv7l.whl", hash = "sha256:06187263ddad280d05b4d8a8b3bb7d164cbebd469236544a42e6d9b28ac6a4fa", size = 56884, upload-time = "2026-05-08T21:00:48.376Z" },
    { url = "https://files.pythonhosted.org/packages/80/45/70b39b89516ff8b96bf732fa6fded8cef20f293cb1508690101c3c07ec51/propcache-0.5.2-cp313-cp313-musllinux_1_2_ppc64le.whl", hash = "sha256:3115559b8effafd63b142ea5ed53d63a16ea6469cbc63dce4ee194b42db5d853", size = 63464, upload-time = "2026-05-08T21:00:49.954Z" },
    { url = "https://files.pythonhosted.org/packages/f9/e2/fa59d3a89eac5534293124af4f1d0d0ada091ce4a0ab4610ce03fd2bdd8d/propcache-0.5.2-cp313-cp313-musllinux_1_2_riscv64.whl", hash = "sha256:c60462af8e6dc30c35407c7237ea908d777b22862bbee27bc4699c0d8bcdc45a", size = 61588, upload-time = "2026-05-08T21:00:51.281Z" },
    { url = "https://files.pythonhosted.org/packages/0b/97/efb547a55c4bc7381cfb202d6a2239ac621045277bc1ea5dfd3a7f0516c0/propcache-0.5.2-cp313-cp313-musllinux_1_2_s390x.whl", hash = "sha256:40314bca9ac559716fe374094fc81c11dcc34b64fd6c585360f5775690505704", size = 64667, upload-time = "2026-05-08T21:00:52.602Z" },
    { url = "https://files.pythonhosted.org/packages/92/56/f5c7d9b4b7595d5127da38974d791b2153f3d1eae6c674af3583ace92ad3/propcache-0.5.2-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:cfa21e036ce1e1db2be04ba3b85d2df1bb1702fa01932d984c5464c665228ff4", size = 62463, upload-time = "2026-05-08T21:00:54.303Z" },
    { url = "https://files.pythonhosted.org/packages/bd/3b/484a3a65fc9f9f60c41dcd17b428bace5389544e2c680994534a20755066/propcache-0.5.2-cp313-cp313-win32.whl", hash = "sha256:f156a3529f38063b6dbaf356e15602a7f95f8055b1295a438433a6386f10463d", size = 38621, upload-time = "2026-05-08T21:00:55.808Z" },
    { url = "https://files.pythonhosted.org/packages/1c/fd/3f0f10dba4dabad3bf53102be007abf55481067952bde0fdddff439e7c61/propcache-0.5.2-cp313-cp313-win_amd64.whl", hash = "sha256:dfed59d0a5aeb01e242e66ff0300bc4a265a7c05f612d30016f0b60b1017d757", size = 41649, upload-time = "2026-05-08T21:00:57.061Z" },
    { url = "https://files.pythonhosted.org/packages/90/ec/6ce619cc32bb500a482f811f9cd509368b4e58e638d13f2c68f370d6b475/propcache-0.5.2-cp313-cp313-win_arm64.whl", hash = "sha256:ba338430e87ceb9c8f0cf754de38a9860560261e56c00376debd628698a7364f", size = 37636, upload-time = "2026-05-08T21:00:58.646Z" },
    { url = "https://files.pythonhosted.org/packages/1b/82/c1d268bbbf2ef981c5bf0fbbe746db617c66e3bcefe431a1aa8943fbe23a/propcache-0.5.2-cp313-cp313t-macosx_10_13_universal2.whl", hash = "sha256:a592f5f3da71c8691c788c13cb6734b6d17663d2e1cb8caddf0673d01ef8847d", size = 98872, upload-time = "2026-05-08T21:00:59.889Z" },
    { url = "https://files.pythonhosted.org/packages/f4/d4/52c871e73e864e6b34c0e2d58ac1ec5ccd149497ddc7ad2137ae98323a35/propcache-0.5.2-cp313-cp313t-macosx_10_13_x86_64.whl", hash = "sha256:6a997d0489e9668a384fcfd5061b857aa5361de73191cac204d04b889cfbbafa", size = 56257, upload-time = "2026-05-08T21:01:01.195Z" },
    { url = "https://files.pythonhosted.org/packages/67/f0/9b90ca2a210b3d09bcfcd96ecd0f55545c091535abce2a45de2775cfd357/propcache-0.5.2-cp313-cp313t-macosx_11_0_arm64.whl", hash = "sha256:10734b5484ea113152ee25a91dccedf81631791805d2c9ccb054958e51842c94", size = 56696, upload-time = "2026-05-08T21:01:02.941Z" },
    { url = "https://files.pythonhosted.org/packages/9d/0e/6e9d4ba07c8e56e21ddec1e75f12148142b21ca83a51871babce095334f4/propcache-0.5.2-cp313-cp313t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:cafca7e56c12bb02ae16d283742bef25a61122e9dab2b5b3f2ccbe589ce32164", size = 62378, upload-time = "2026-05-08T21:01:04.475Z" },
    { url = "https://files.pythonhosted.org/packages/65/19/c10badaa463dde8a27ce884f8ee2ec37e6035b7c9f5ff0c8f74f06f08dac/propcache-0.5.2-cp313-cp313t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:f064f8d2b59177878b7615df1735cd8fe3462ed6be8c7b217d17a276489c2b7f", size = 65283, upload-time = "2026-05-08T21:01:05.959Z" },
    { url = "https://files.pythonhosted.org/packages/b0/b6/93bea99ca80e19cef6512a8580e5b7857bbe09422d9daa7fd4ef5723306c/propcache-0.5.2-cp313-cp313t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:f78abfa8dfc32376fd1aacf597b2f2fbbe0ea751419aee718af5d4f82537ef8c", size = 66616, upload-time = "2026-05-08T21:01:07.228Z" },
    { url = "https://files.pythonhosted.org/packages/83/e4/5c7462e50625f051f37fb38b8224f7639f667184bbd34424ec83819bb1b7/propcache-0.5.2-cp313-cp313t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:f7467da8a9822bf1a55336f877340c5bcbd3c482afc43a99771169f74a26dedc", size = 63773, upload-time = "2026-05-08T21:01:08.514Z" },
    { url = "https://files.pythonhosted.org/packages/ca/b6/99238894047b13c823be25027e736626cd414a52a5e30d2c3347c2733529/propcache-0.5.2-cp313-cp313t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:a6ddc6ac9e25de626c1f129c1b467d7ecd33ce2237d3fd0c4e429feef0a7ee1f", size = 63664, upload-time = "2026-05-08T21:01:09.874Z" },
    { url = "https://files.pythonhosted.org/packages/85/1e/a3a1a63116a2b8edb415a8bb9a6f0c34bd03830b1e18e8ce2904e1dc1cf4/propcache-0.5.2-cp313-cp313t-musllinux_1_2_aarch64.whl", hash = "sha256:2f22cbbac9e26a8e864c0985ff1268d5d939d53d9d9411a9824279097e03a2cb", size = 62643, upload-time = "2026-05-08T21:01:11.132Z" },
    { url = "https://files.pythonhosted.org/packages/e4/03/893cf147de2fc6543c5eaa07ad833170e7e2a2385725bbebe8c0503723bb/propcache-0.5.2-cp313-cp313t-musllinux_1_2_armv7l.whl", hash = "sha256:fc76378c62a0f04d0cd82fbb1a2cd2d7e28fcb40d5873f28a6c44e388aaa2751", size = 59595, upload-time = "2026-05-08T21:01:12.387Z" },
    { url = "https://files.pythonhosted.org/packages/86/3b/04c1a2e12c57766568ba75ba72b3bf2042818d4c1425fab6fc07155c7cff/propcache-0.5.2-cp313-cp313t-musllinux_1_2_ppc64le.whl", hash = "sha256:acd2c8edba48e31e58a363b8cf4e5c7db3b04b3f9e371f601df30d9b0d244836", size = 65711, upload-time = "2026-05-08T21:01:13.676Z" },
    { url = "https://files.pythonhosted.org/packages/1c/34/80f8d0099f8d6bacc4de1624c85672681c8cd1149ca2da0e38fd120b817f/propcache-0.5.2-cp313-cp313t-musllinux_1_2_riscv64.whl", hash = "sha256:452b5065457eb9991ec5eb38ff41d6cd4c991c9ac7c531c4d5849ae473a9a13f", size = 64247, upload-time = "2026-05-08T21:01:14.936Z" },
    { url = "https://files.pythonhosted.org/packages/f3/1a/8b08f3a5f1037e9e370c55883ceeeee0f6dd0416fb2d2d67b8bfc91f2a79/propcache-0.5.2-cp313-cp313t-musllinux_1_2_s390x.whl", hash = "sha256:3430bb2bfe1331885c427745a751e774ee679fd4344f80b97bf879815fe8fa55", size = 67102, upload-time = "2026-05-08T21:01:16.281Z" },
    { url = "https://files.pythonhosted.org/packages/34/68/8bdb7bb7756d76e005490649d10e4a8369e610c74d619f71e1aedf889e9c/propcache-0.5.2-cp313-cp313t-musllinux_1_2_x86_64.whl", hash = "sha256:cef6cea3922890dd6c9654971001fa797b526c16ab5e1e46c05fd6f877be7568", size = 64964, upload-time = "2026-05-08T21:01:17.57Z" },
    { url = "https://files.pythonhosted.org/packages/0a/aa/50fb0b5d3968b61a510926ff8b8465f1d6e976b3ab74496d7a4b9fc42515/propcache-0.5.2-cp313-cp313t-win32.whl", hash = "sha256:72d61e16dd78228b58c5d47be830ff3da7e5f139abdf0aef9d86cde1c5cf2191", size = 42546, upload-time = "2026-05-08T21:01:18.946Z" },
    { url = "https://files.pythonhosted.org/packages/ae/4c/0ddbae64321bd4a95bcbfc19307238016b5b1fee645c84626c8d539e5b74/propcache-0.5.2-cp313-cp313t-win_amd64.whl", hash = "sha256:0958834041a0166d343b8d2cedcd8bcbaeb4fdbe0cf08320c5379f143c3be6e7", size = 46330, upload-time = "2026-05-08T21:01:20.162Z" },
    { url = "https://files.pythonhosted.org/packages/00/d9/9cddc8efb78d8af264c5ec9f6d10b62f57c515feda8d321595f56010fb23/propcache-0.5.2-cp313-cp313t-win_arm64.whl", hash = "sha256:6de8bd93ddde9b992cf2b2e0d796d501a19026b5b9fd87356d7d0779531a8d96", size = 40521, upload-time = "2026-05-08T21:01:21.399Z" },
    { url = "https://files.pythonhosted.org/packages/e2/ea/23ee535d90ce8bcc465a3028eb3cc0ce3bd1005f4bb27710b30587de798d/propcache-0.5.2-cp314-cp314-macosx_10_15_universal2.whl", hash = "sha256:46088abff4cba581dea21ae0467a480526cb25aa5f3c269e909f800328bc3999", size = 94662, upload-time = "2026-05-08T21:01:22.683Z" },
    { url = "https://files.pythonhosted.org/packages/b5/06/c5a52f419b5d8972f8d46a7577476090d8e3263ff589ce40b5ca4968d5be/propcache-0.5.2-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:fc88b26f08d634f7bc819a7852e5214f5802641ab8d9fd5326892292eee1993e", size = 53928, upload-time = "2026-05-08T21:01:23.986Z" },
    { url = "https://files.pythonhosted.org/packages/63/b1/4260d67d6bd85e58a66b72d54ce15d5de789b6f3870cc6bedf8ff9667401/propcache-0.5.2-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:97797ebb098e670a2f92dd66f32897e30d7615b14e7f59711de23e30a9072539", size = 54650, upload-time = "2026-05-08T21:01:25.305Z" },
    { url = "https://files.pythonhosted.org/packages/70/06/2f46c318e3307cd7a6a7481def374ce838c0fe20084b39dd54b0879d0e99/propcache-0.5.2-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:ba57fffe4ac99c5d30076161b5866336d97600769bad35cc68f7774b15298a4e", size = 59912, upload-time = "2026-05-08T21:01:26.545Z" },
    { url = "https://files.pythonhosted.org/packages/4c/29/fe1aebec2ce57ab985a9c382bded1124431f85078113aa222c5d278430d4/propcache-0.5.2-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:583c19759d9eec1e5b69e2fbef36a7d9c326041be9746cb822d335c8cedc2979", size = 63300, upload-time = "2026-05-08T21:01:27.937Z" },
    { url = "https://files.pythonhosted.org/packages/b4/18/2334b26768b6c82be8c69e83671b767d5ef426aa09b0cba6c2ea47816774/propcache-0.5.2-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:d0326e2e5e1f3163fa306c834e48e8d490e5fae607a097a40c0648109b47ba80", size = 64208, upload-time = "2026-05-08T21:01:29.484Z" },
    { url = "https://files.pythonhosted.org/packages/2b/76/7f1bfd6afff4c5e38e36a3c6d68eb5f4b7311ea80baf693db78d95b603c4/propcache-0.5.2-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:e00820e192c8dbebcafb383ebbf99030895f09905e7a0eb2e0340a0bcc2bc825", size = 61633, upload-time = "2026-05-08T21:01:31.068Z" },
    { url = "https://files.pythonhosted.org/packages/c4/46/b3ff8aba2b4953a3e50de2cf72f1b5748b8eca93b15f3dc2c84339084c09/propcache-0.5.2-cp314-cp314-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:c66afea89b1e43725731d2004732a046fe6fe955d51f952c3e95a7314a284a39", size = 61724, upload-time = "2026-05-08T21:01:32.374Z" },
    { url = "https://files.pythonhosted.org/packages/c5/01/814cfcafbcff954f94c01cf30e097ddc88a076b5440fbcf4570753437d40/propcache-0.5.2-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:d4dc37dec6c6cdad0b57881a5658fd14fbf53e333b1a86cf86559f190e1d9ec4", size = 60069, upload-time = "2026-05-08T21
```

```
:01:33.67Z" },
    { url = "https://files.pythonhosted.org/packages/da/68/5c6f7622d510cc666a300687e06fd060c1a43361c0c9b20d284f06d8096a/propcache-0.5.2-cp314-cp314-mu
sllinux_1_2_armv7l.whl", hash = "sha256:5570dbcc97571c15f68068e529c92715a12f8d54030e272d264b377e22bd17a5", size = 57099, upload-time = "2026-05-08T21:
01:34.915Z" },
    { url = "https://files.pythonhosted.org/packages/55/27/9cb0b4c679124085327957d42521c99dba04c88c90c3e55a6f0b633ebccc/propcache-0.5.2-cp314-cp314-mu
sllinux_1_2_ppc64le.whl", hash = "sha256:f814362777a9f841adddb200ecdf8f5cb1e5a3c4b7a86378edbd6ccb26edd702", size = 63391, upload-time = "2026-05-08T21
:01:36.231Z" },
    { url = "https://files.pythonhosted.org/packages/f0/9d/7258aaa5bdf60fc6f27591eef6fe52768cb0beda7140be477c8b12c9794a/propcache-0.5.2-cp314-cp314-mu
sllinux_1_2_riscv64.whl", hash = "sha256:196913dea116aeb5a2ba95af4ddcb7ea85559ae07d8eee8751688310d09168c3", size = 61626, upload-time = "2026-05-08T21
:01:37.545Z" },
    { url = "https://files.pythonhosted.org/packages/8e/0d/41c602003e8a9b16fe1e7eadf62c7bfba9d5474370b24200bf48b315f45f/propcache-0.5.2-cp314-cp314-mu
sllinux_1_2_s390x.whl", hash = "sha256:6e7b8719005dd1175be4ab1cd25e9b98659a5e0347331506ec6760d2773a7fb5", size = 64781, upload-time = "2026-05-08T21:0
1:38.83Z" },
    { url = "https://files.pythonhosted.org/packages/8b/f3/38e66b1856e9bd079deea015bc4a55f7767c0e4db2f7dcf69e7e680ba4ce/propcache-0.5.2-cp314-cp314-mu
sllinux_1_2_x86_64.whl", hash = "sha256:51f96d685ab16e88cab128cd37a52c5da540809c8b879fa047731bfcb4ad35a4", size = 62570, upload-time = "2026-05-08T21:
01:40.415Z" },
    { url = "https://files.pythonhosted.org/packages/95/ca/bbfe9b910ce57dde8bb4876b4520fc02a4e89497c10de26be936758a3aaa/propcache-0.5.2-cp314-cp314-wi
n32.whl", hash = "sha256:cc6fc3cc62e8501d3ed62894425040d2728ecddb1ed072737a5c70bd537aa9f0", size = 39436, upload-time = "2026-05-08T21:01:41.654Z" },
    { url = "https://files.pythonhosted.org/packages/61/d2/45c9defbaa1ea297035d9d4cce9e8f80daafbf19319c6007f157c6256ea9/propcache-0.5.2-cp314-cp314-wi
n_amd64.whl", hash = "sha256:81e3a30b0bb60caa22033dd0f8a3618d1d67356212514f62c57db75cb0ef410c", size = 42373, upload-time = "2026-05-08T21:01:43.041Z"
 },
    { url = "https://files.pythonhosted.org/packages/44/68/9ea5103f41d5217d7d6ec24db90018e23aebec070c3f9a6e54d12b841fd8/propcache-0.5.2-cp314-cp314-wi
n_arm64.whl", hash = "sha256:0d2c9bf8528f135dbb805ce027567e09164f7efa51a2be07458a2c0420f292d0", size = 38554, upload-time = "2026-05-08T21:01:44.336Z"
 },
    { url = "https://files.pythonhosted.org/packages/8a/81/fadf555f42d3b762eea8a53950b0489fdc0aa9da5f8ed9e10ce0a4e01b48/propcache-0.5.2-cp314-cp314t-m
acosx_10_15_universal2.whl", hash = "sha256:4bc8ff1feffc6a61c7002ffe84634c41b822e104990ae009f44a0834430070bb", size = 99395, upload-time = "2026-05-08
T21:01:45.883Z" },
    { url = "https://files.pythonhosted.org/packages/f5/c9/c61e134a686949cf7971af3a390148b1156f7be81c73bc0cd12c873e2d48/propcache-0.5.2-cp314-cp314t-m
acosx_10_15_x86_64.whl", hash = "sha256:79aa3ff0a9b566633b642fa9caf7e21ed1c13d6feca718187873f199e1514078", size = 56653, upload-time = "2026-05-08T21:
01:47.307Z" },
    { url = "https://files.pythonhosted.org/packages/cb/73/daf935ea7048ddd7ec8eec5345b4a40b619d2d178b3c0a0900796bc3c794/propcache-0.5.2-cp314-cp314t-m
acosx_11_0_arm64.whl", hash = "sha256:1b31822f4474c4036bae62de9402710051d431a606d6a0f907fec79935a071aa", size = 56914, upload-time = "2026-05-08T21:01
:48.573Z" },
    { url = "https://files.pythonhosted.org/packages/79/9f/aba959b435ea18617edd7cf0a7ad0b9c574b8fc7e3d2cd55fb59cb255d33/propcache-0.5.2-cp314-cp314t-m
anylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:13fef48778b5a2a756523fdb781326b028ca75e32858b04f2cdd19f3945649
17", size = 62567, upload-time = "2026-05-08T21:01:49.903Z" },
    { url = "https://files.pythonhosted.org/packages/6c/a1/859942de9a791ff42f6141736f5b37749b8f53e65edfa49638c67dd67e6a/propcache-0.5.2-cp314-cp314t-m
anylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:8b73ab70f1a3351fbc71f663b3e645af6dd0329100c353081cf69c37433fc6
fe", size = 65542, upload-time = "2026-05-08T21:01:51.204Z" },
    { url = "https://files.pythonhosted.org/packages/b5/61/315bc0fd6c0fc7f80a528b8afd209e5fc4a875ea79571b91b8f50f442907/propcache-0.5.2-cp314-cp314t-m
anylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:5538d2c13d93e4698af7e092b57bc7298fd35d1d58e656ae18f23ee0d0378e03", s
ize = 66845, upload-time = "2026-05-08T21:01:52.539Z" },
    { url = "https://files.pythonhosted.org/packages/47/f7/9f8122e3132e8e354ac41975ef8f1099be7d5a16bc7ae562734e993665c0/propcache-0.5.2-cp314-cp314t-m
anylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:cd645f03898405cabe694fb8bc35241e3a9c332ec85627584fe3de201452b335"
, size = 63985, upload-time = "2026-05-08T21:01:53.847Z" },
    { url = "https://files.pythonhosted.org/packages/c8/54/c317819ec157cbf6f35df9df9657a6f82daf34d5faf15948b2f639c2192e/propcache-0.5.2-cp314-cp314t-m
anylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:a473b3440261e0c60706e732b2ed2f517857344fc21bf48fdfe211e2d98eb285", size = 63999, upl
oad-time = "2026-05-08T21:01:55.179Z" },
    { url = "https://files.pythonhosted.org/packages/5a/56/387e3f7dfce0a9233df41fb888aa1c30222cb4bbbf09537c02dd9bd85fe2/propcache-0.5.2-cp314-cp314t-m
usllinux_1_2_aarch64.whl", hash = "sha256:7afa37062e6650640e932e4cc9297d81f9f42d9944029cc386b8247dea4da837", size = 62779, upload-time = "2026-05-08T2
1:01:57.489Z" },
    { url = "https://files.pythonhosted.org/packages/a1/9c/596784cb5824ed61ee960d3f8655a3f0993e107c6e98ab6c818b7fb92ccb/propcache-0.5.2-cp314-cp314t-m
usllinux_1_2_armv7l.whl", hash = "sha256:8a90efd5777e996e42d568db9ac740b944d691e565cbfd31b2f7832f9184b2b8", size = 59796, upload-time = "2026-05-08T21
:01:58.736Z" },
    { url = "https://files.pythonhosted.org/packages/c2/3d/1a6cfa1726a48542c1e8784a0761421476a5b68e09b7f36bf95eb954aaba/propcache-0.5.2-cp314-cp314t-m
usllinux_1_2_ppc64le.whl", hash = "sha256:f19bb891234d72535764d703bfed1153cc34f4214d5bd7150aee1eec9e8f4366", size = 66023, upload-time = "2026-05-08T2
1:02:00.228Z" },
    { url = "https://files.pythonhosted.org/packages/e4/0e/05fd6990369477076e4e280bcb970de760fddf0161a46e988bc95f7940ec/propcache-0.5.2-cp314-cp314t-m
usllinux_1_2_riscv64.whl", hash = "sha256:32775082acd2d807ee3db715c7770d38767b817870acfa08c29e057f3c4d5b56", size = 64448, upload-time = "2026-05-08T2
1:02:01.888Z" },
    { url = "https://files.pythonhosted.org/packages/cd/86/5f8da315a4309c62c10c0b2516b17492d5d3bbe1bb862b96604db67e2a37/propcache-0.5.2-cp314-cp314t-m
usllinux_1_2_s390x.whl", hash = "sha256:9282fb1a3bccd038da9f768b927b24a0c753e466c086b7c4f3c6982851eefb2d", size = 67329, upload-time = "2026-05-08T21:
02:03.484Z" },
    { url = "https://files.pythonhosted.org/packages/da/d3/3368efe79ab21f0cdf86ef49895811c9cc933131d4cde1f28a624e22e712/propcache-0.5.2-cp314-cp314t-m
usllinux_1_2_x86_64.whl", hash = "sha256:cc49723e2f60d6b32a0f0b08a3fd6d13203c07f1cd9566cfce0f12a917c967a2", size = 65172, upload-time = "2026-05-08T21
:02:04.745Z" },
    { url = "https://files.pythonhosted.org/packages/d5/07/127e8b0bacfb325396196f9d976a22453049b89b9b2b08477cc3145faa44/propcache-0.5.2-cp314-cp314t-w
in32.whl", hash = "sha256:2d7aa89ebca5acc98cba9d1472d976e394782f587bad6661003602a619fd1821", size = 43813, upload-time = "2026-05-08T21:02:06.025Z" },
    { url = "https://files.pythonhosted.org/packages/88/fb/46dad6c0ae49ed230ab1b16c890c2b6314e2403e6c412976f4a72d64a527/propcache-0.5.2-cp314-cp314t-w
in_amd64.whl", hash = "sha256:d447bb0b3054be5818458fbb171208b1d9ff11eba14e18ca18b90cbb45767370", size = 47764, upload-time = "2026-05-08T21:02:07.353Z
" },
    { url = "https://files.pythonhosted.org/packages/e7/c4/a47d0a63aa309d10d59ede6e9d4cff03a344a79d1f0f4cd0cd74997b53e0/propcache-0.5.2-cp314-cp314t-w
in_arm64.whl", hash = "sha256:fe67a3d11cd9b4efabfa45c3d00ffba2b26811442a73a581a94b67c2b5faccf6", size = 41140, upload-time = "2026-05-08T21:02:09.065Z
" },
    { url = "https://files.pythonhosted.org/packages/3a/ed/1cdcab6ba3d6ab7feca11fc14f0eeea80755bb53ef4e892079f31b10a25f/propcache-0.5.2-py3-none-any.w
hl", hash = "sha256:be1ddfcbb376e3de5d2e2db1d58d6d67463e6b4f9f040c000de8e300295465fe", size = 14036, upload-time = "2026-05-08T21:02:10.673Z" },
]

[[package]]
name = "protobuf"
version = "7.35.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/da/01/9ef0afd7999eb9badb3a768b4aedd78c86d4c65cfaf1958ab276199e76b4/protobuf-7.35.1.tar.gz", h
ash = "sha256:ce115a26fe0c39a2c29973d914d327e516a6455464489fe3cd1e51a1b354f81a", size = 458717, upload-time = "2026-06-11T21:55:40.257Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/10/03/8aeeb7458d22546bf64b5250ca1daeb5ff757d900e8e4a7476c6f0db843e/protobuf-7.35.1-cp310-abi3-mac
osx_10_9_universal2.whl", hash = "sha256:24f857477359a85c0c235261b8ba905fd51b2562f4a64ca1df5473f29850cbf6", size = 433226, upload-time = "2026-06-11T2
1:55:31.719Z" },
    { url = "https://files.pythonhosted.org/packages/37/4b/dfb89eb0e652a1ff073c39a59fb5e3a83cfe9b57a2c83fa6d78270101767/protobuf-7.35.1-cp310-abi3-man
ylinux2014_aarch64.whl", hash = "sha256:11d6b0ec246892d85215b0a13ca6e0233cf5284b68f0ac02646427f4ff88a799", size = 328847, upload-time = "2026-06-11T21
:55:34.035Z" },
    { url = "https://files.pythonhosted.org/packages/0f/58/dc12f2cd484951524af6e3382c785869b9b3fb5e52ee95ae23add53ee8f9/protobuf-7.35.1-cp310-abi3-man
ylinux2014_s390x.whl", hash = "sha256:b73f9489a4b8b1c9cb1f8ed951c736392592edb24b9d6819f36d2e10b171d5b4", size = 344030, upload-time = "2026-06-11T21:5
5:34.941Z" },
    { url = "https://files.pythonhosted.org/packages/e4/be/5b3cfe508bfab6761414ff944e3366eb13be4fd71efcd69450f89ba39f43/protobuf-7.35.1-cp310-abi3-man
ylinux2014_x86_64.whl", hash = "sha256:74758715c53d7158fb76caf4f0cfdacc5329a4b1bb994f865d6cf302d413a1c4", size = 327130, upload-time = "2026-06-11T21:
55:35.921Z" },
    { url = "https://files.pythonhosted.org/packages/d8/bc/6d6c7ba8709c85f8f2c390b2b118d6fb08a783676a572271851bf45a7d22/protobuf-7.35.1-cp310-abi3-win
32.whl", hash = "sha256:353652e4efd0bca5b5fc2656abf8307ef351f0cf938c9eba09f0e09c20a25c30", size = 428945, upload-time = "2026-06-11T21:55:37.034Z" },
    { url = "https://files.pythonhosted.org/packages/0a/19/8d0cb6f20a1ef7b18f1c8986ad5783f22f84cce39c6ce9a6e645ea55192e/protobuf-7.35.1-cp310-abi3-win
_amd64.whl", hash = "sha256:230a75ddfc2de4806e56696ce9640c1cdfdb6543b7cfce98d42a4c0a0e7bdb87", size = 439996, upload-time = "2026-06-11T21:55:38.123Z"
 },
    { url = "https://files.pythonhosted.org/packages/19/c7/5f7c636ec43e0c545e28d1f1db71990108306f7bdcb89f069ba97e428e7f/protobuf-7.35.1-py3-none-any.w
hl", hash = "sha256:4bc97768d8fe4ad6743c8a19403e314511ed9f6d13205b687e52421c023ac1b9", size = 171659, upload-time = "2026-06-11T21:55:39.155Z" },
]
```

```

[[package]]
name = "pyasn1"
version = "0.6.4"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a4/9a/23310166d960def5897e91fe20e5b724601b02a22e84ba1f94232c0b7f67/pyasn1-0.6.4.tar.gz", hash
 = "sha256:9c447d8431c947fe4c8febc4ed9e760bc29011a5b01e5c74b67025bd9fb8ce81", size = 151262, upload-time = "2026-07-09T01:12:33.988Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/9a/3b/6163796d69c3977d1e4287bea4a6979161cbbdd170ebb430511e8e1999ce/pyasn1-0.6.4-py3-none-any.whl"
, hash = "sha256:deda9277cfd454080ec40b207fb6df82206a3a2688735233cdcd8d3d565f088b", size = 84410, upload-time = "2026-07-09T01:12:32.92Z" },
]

[[package]]
name = "pycparser"
version = "3.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/1b/7d/92392ff7815c21062bea51aa7b87d45576f649f16458d78b7cf94b9ab2e6/pycparser-3.0.tar.gz", has
h = "sha256:600f49d217304a5902ac3c37e1281c9fe94e4d0489de643a9504c5cdfdfc6b29", size = 103492, upload-time = "2026-01-21T14:26:51.89Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/0c/c3/44f3fbbfa403ea2a7c779186dc2072604442dde72947e7d01069cbe98e3/pycparser-3.0-py3-none-any.whl
", hash = "sha256:b727414169a36b7d524c1c3e31839a521725078d7b2ff038656844266160a992", size = 48172, upload-time = "2026-01-21T14:26:50.693Z" },
]

[[package]]
name = "pydantic"
version = "2.13.4"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "annotated-types" },
    { name = "pydantic-core" },
    { name = "typing-extensions" },
    { name = "typing-inspection" },
]
sdist = { url = "https://files.pythonhosted.org/packages/18/a5/b60d21ac674192f8ab0ba4e9fd860690f9b4a6e51ca5df118733b487d8d6/pydantic-2.13.4.tar.gz", h
ash = "sha256:c40756b57adaa8b1efeeced5c196f3f3b7c435f90e84ea7f443901bec8099ef6", size = 844775, upload-time = "2026-05-06T13:43:05.343Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/fd/7b/122376b1fd3c62c1ed9dc80c931ace4844b3c55407b6fb2d199377c9736f/pydantic-2.13.4-py3-none-any.w
hl", hash = "sha256:45a282cde31d808236fd7ea9d919b128653c8b38b393d1c4ab335c62924d9aba", size = 472262, upload-time = "2026-05-06T13:43:02.641Z" },
]

[[package]]
name = "pydantic-core"
version = "2.46.4"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/9d/56/921726b776ace8d8f5db44c4ef961006580d91dc52b803c489fafd1aa249/pydantic_core-2.46.4.tar.g
z", hash = "sha256:62f875393d7f270851f20523dd2e29f082bcc82292d66db2b64ea71f64b6e1c1", size = 471464, upload-time = "2026-05-06T13:37:06.98Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/ce/8c/af022f0af448d7747c5154288d46b5f2bc5f17366eaa0e23e9aa04d59f3b/pydantic_core-2.46.4-cp312-cp3
12-macosx_10_12_x86_64.whl", hash = "sha256:3245406455a5d98187ec35530fd772b1d799b26667980872c8d4614991e2c4a2", size = 2106158, upload-time = "2026-05-
06T13:38:57.215Z" },
    { url = "https://files.pythonhosted.org/packages/19/95/6195171e385007300f0f5574592e467c568becce2d937a0b6804f218bc49/pydantic_core-2.46.4-cp312-cp3
12-macosx_11_0_arm64.whl", hash = "sha256:962ccbab7b642487b1d8b7df90ef677e03134cf1fd8880bf698649b22a69371f", size = 1951724, upload-time = "2026-05-06
T13:37:02.697Z" },
    { url = "https://files.pythonhosted.org/packages/8e/bc/f47d1ff9cbb1620e1b5b697eef06010035735f07820180e74178226b27b3/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:8233f2947cf85404441fd7e0085f53b10c93e0ee78611099b5c7237e36aacbf7", size = 1975742
, upload-time = "2026-05-06T13:37:09.448Z" },
    { url = "https://files.pythonhosted.org/packages/5b/11/9b9a5b0306345664a2da6410877af6e8082481b5884b3ddd78d47c6013ce/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:3a233125ac121aa3ffba9a2b59edfc4a985a76092dc8279586ab4b71390875e7", size = 2052418,
upload-time = "2026-05-06T13:37:38.234Z" },
    { url = "https://files.pythonhosted.org/packages/f1/b7/a65fec226f5d78fc39f4a13c4cc0c768c22b113438f60c14adc9d2865038/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:5b712b53160b79a5850310b912a5ef8e57e56947c8ad690c227f5c9d7e561712", size = 2232274
, upload-time = "2026-05-06T13:38:27.753Z" },
    { url = "https://files.pythonhosted.org/packages/68/f0/92039db98b907ef49269a8271f67db9cb78ae2fc68062ef7e4e77adb5f61/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:9401557acd873c3a7f3eb9383edef8ac4968f9510e340f4808d427e75667e7b4", size = 2309940, up
load-time = "2026-05-06T13:38:05.353Z" },
    { url = "https://files.pythonhosted.org/packages/5f/97/2aab507d3d00ca626e8e57c1eac6a79e4e5fbcc63eb99733ff55d1717f65/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:926c9541b14b12b1681dca8a0b75feb510b06c6341b70a8e500c2fdcff837cce", size = 2094516,
upload-time = "2026-05-06T13:39:10.577Z" },
    { url = "https://files.pythonhosted.org/packages/22/37/a8aca44d40d737dde2bc05b3c6c07dff0de07ce6f82e9f3167aeaf4d5dea/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_31_riscv64.whl", hash = "sha256:56cb4851bcaf3d117eddcef4fe66afd750a50274b0da8e22be256d10e5611987", size = 2136854, upload-time = "2026-
05-06T13:40:22.59Z" },
    { url = "https://files.pythonhosted.org/packages/24/99/fcef1b79238c06a8cbec70819ac722ba76e02bc8ada9b0fd66eba40da01b/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:c68fcd102d71ea85c5b2dfac3f4f8476eff42a9e078fd5faefff6d145063536b", size = 2180306, upload-t
ime = "2026-05-06T13:40:10.666Z" },
    { url = "https://files.pythonhosted.org/packages/ae/6c/fc44000918855b42779d007ae63b0532794739027b2f417321cddbc44f6a/pydantic_core-2.46.4-cp312-cp3
12-musllinux_1_1_aarch64.whl", hash = "sha256:b2f69dec1725e79a012d920df1707de5caf7ed5e08f3be4435e25803efc47458", size = 2190044, upload-time = "2026-0
5-06T13:40:43.231Z" },
    { url = "https://files.pythonhosted.org/packages/6b/65/d9cadc9f1920d7a127ad2edba16c1db7916e59719285cd6c94600b0080ba/pydantic_core-2.46.4-cp312-cp3
12-musllinux_1_1_armv7l.whl", hash = "sha256:8d0820e8192167f80d88d64038e609c31452eeca865b4e1d9950a27a4609b00b", size = 2329133, upload-time = "2026-05
-06T13:39:57.365Z" },
    { url = "https://files.pythonhosted.org/packages/d0/cf/c873d91679f3a30bcf5e7ac280ce5573483e72295307685120d0d5ad3416/pydantic_core-2.46.4-cp312-cp3
12-musllinux_1_1_x86_64.whl", hash = "sha256:fbdb89b3e1c94a30cc5edfce477c6e6a5dc4d8f84665b455c27582f211a1c72c", size = 2374464, upload-time = "2026-05
-06T13:38:06.976Z" },
    { url = "https://files.pythonhosted.org/packages/47/bd/6f2fc8188f31bf10590f1e98e7b306336161fac930a8c514cd7bd828c7dc/pydantic_core-2.46.4-cp312-cp3
12-win32.whl", hash = "sha256:9aa768456404a8bf48a4406685ac2bec8e72b62c69313734fa3b73cf33b3a894", size = 1974823, upload-time = "2026-05-06T13:40:47.98
5Z" },
    { url = "https://files.pythonhosted.org/packages/40/8c/985c1d41ea1107c2534abd9870e4ed5c8e7669b5c308297835c001e7a1c4/pydantic_core-2.46.4-cp312-cp3
12-win_amd64.whl", hash = "sha256:e9c26f834c65f5752f3f06cb08cb86a913ceb7274d0db6e267808a708b46bc89", size = 2072919, upload-time = "2026-05-06T13:39:2
1.153Z" },
    { url = "https://files.pythonhosted.org/packages/c4/ba/f463d006e0c47373ca7ec5e1a261c59dc01ef4d62b2657af925fb0deee3a/pydantic_core-2.46.4-cp312-cp3
12-win_arm64.whl", hash = "sha256:4fc73cb559bdb54b1134a706a2802a4cddd27a0633f5abb7e5305268751ac6a", size = 2027604, upload-time = "2026-05-06T13:39:0
3.753Z" },
    { url = "https://files.pythonhosted.org/packages/51/a2/5d30b469c5267a17b39dec53208222f76a8d351dfac4af661888c5aee77d/pydantic_core-2.46.4-cp313-cp3
13-macosx_10_12_x86_64.whl", hash = "sha256:5d5902252db0d3cedf8d4a1bc68f70eeb430f7e4c7104c8c476753519b423008", size = 2106306, upload-time = "2026-05-
06T13:37:48.029Z" },
    { url = "https://files.pythonhosted.org/packages/c1/81/4fa520eaffa8bd7d1525e644cd6d39e7d60b1592bc5b516693c7340b50f1/pydantic_core-2.46.4-cp313-cp3
13-macosx_11_0_arm64.whl", hash = "sha256:c94f0688e7b8d0a67abf40e57a7eaaecd17cc9586706a31b76c031f63df052b4", size = 1951906, upload-time = "2026-05-06
T13:37:17.012Z" },
    { url = "https://files.pythonhosted.org/packages/03/d5/fd02da45b659668b05923b17ba3a0100a0a3d5541e3bd8fcc4ecb711309e/pydantic_core-2.46.4-cp313-cp3
13-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:f027324c56cd5406ca49c124b0db10e56c69064fec039acc571c29020cc87c76", size = 1976802
, upload-time = "2026-05-06T13:37:35.113Z" },
    { url = "https://files.pythonhosted.org/packages/21/f2/95727e1368be3d3ed485eaab7adbd7dda408f33f7a36e8b48e0144002b91/pydantic_core-2.46.4-cp313-cp3
```

```
13-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:e739fee756ba1010f8bcccb534252e85a35fe45ae92c295a06059ce58b74ccd3", size = 2052446,
upload-time = "2026-05-06T13:37:12.313Z" },
    { url = "https://files.pythonhosted.org/packages/9c/86/5d99feea3f77c7234b8718075b23db11532773c1a0dbd9b9490215dc2eeb/pydantic_core-2.46.4-cp313-cp3
13-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:9d56801be94b86a9da183e5f3766e6310752b99ff647e38b09a9500d88e46e76", size = 2232757
, upload-time = "2026-05-06T13:39:01.149Z" },
    { url = "https://files.pythonhosted.org/packages/d2/3a/508ac615935ef7588cf6d9e9b91309fdc2da751af865e02a9098de88258c/pydantic_core-2.46.4-cp313-cp3
13-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:2412e734dcb48da14d4e4006b82b46b74f2518b8a26ee7e58c6844a6cd6d03c4", size = 2309275, up
load-time = "2026-05-06T13:37:41.406Z" },
    { url = "https://files.pythonhosted.org/packages/07/f8/41db9de19d7987d6b04715a02b3b40aea467000275d9d758ffaa31af7d50/pydantic_core-2.46.4-cp313-cp3
13-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:9551187363ffc0de2a00b2e47c25aeaeb1020b69b668762966df15fc5659dd5a", size = 2094467,
upload-time = "2026-05-06T13:39:18.847Z" },
    { url = "https://files.pythonhosted.org/packages/2c/e2/f35033184cb11d0052daf4416e8e10a502ea2ac006fc4f459aee872727d1/pydantic_core-2.46.4-cp313-cp3
13-manylinux_2_31_riscv64.whl", hash = "sha256:0186750b482eefa11d7f435892b09c5c606193ef3375bcf94aa00ae6bfb66262", size = 2134417, upload-time = "2026-
05-06T13:40:17.944Z" },
    { url = "https://files.pythonhosted.org/packages/7e/7b/6ceeb1cc90e193862f444ebe373d8fdf613f0a82572dde03fb10734c6c71/pydantic_core-2.46.4-cp313-cp3
13-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:5855698a4856556d86e8e6cd8434bc3ac0314ee8e12089ae0e143f64c6256e4e", size = 2179782, upload-t
ime = "2026-05-06T13:40:32.618Z" },
    { url = "https://files.pythonhosted.org/packages/5a/f2/c8d7773ede6af08036423a00ae0ceffce266c3c52a096c435d68c896083f/pydantic_core-2.46.4-cp313-cp3
13-musllinux_1_1_aarch64.whl", hash = "sha256:cbaf13819775b7f769bf4a1f066cb6df7a28d4480081a589828ef190226881cd", size = 2188782, upload-time = "2026-0
5-06T13:36:51.018Z" },
    { url = "https://files.pythonhosted.org/packages/59/31/0c864784e31f09f05cdd87606f08923b9c9e7f6e51dd27f20f62f975ce9f/pydantic_core-2.46.4-cp313-cp3
13-musllinux_1_1_armv7l.whl", hash = "sha256:633147d34cf4550417f12e2b1a0383973bdf5cdfde212cb09e9a581cf10820be", size = 2328334, upload-time = "2026-05
-06T13:40:37.764Z" },
    { url = "https://files.pythonhosted.org/packages/c2/eb/4f6c8a41efa30baa755590f4141abf3a8c370fab610915733e74134a7270/pydantic_core-2.46.4-cp313-cp3
13-musllinux_1_1_x86_64.whl", hash = "sha256:82cf5301172168103724d49a1444d3378cb20cdee30b116a1bd6031236298a5d", size = 2372986, upload-time = "2026-05
-06T13:39:34.152Z" },
    { url = "https://files.pythonhosted.org/packages/5b/24/b375a480d53113860c299764bfe9f349a3dc9108b3adc0d7f0d786492ebf/pydantic_core-2.46.4-cp313-cp3
13-win32.whl", hash = "sha256:9fa8ae11da9e2b3126c6426f147e0fba88d96d65921799bb30c6abd1cb2c97fb", size = 1973693, upload-time = "2026-05-06T13:37:55.07
2Z" },
    { url = "https://files.pythonhosted.org/packages/7e/e8/cff247591966f2d22ec8c003cd7587e27b7ba7b81ab2fb888e3ab75dc285/pydantic_core-2.46.4-cp313-cp3
13-win_amd64.whl", hash = "sha256:6b3ace8194b0e5204818c92802dcdca7fc6d88aabbb799d7c795540d9cd6d292", size = 2071819, upload-time = "2026-05-06T13:38:4
9.139Z" },
    { url = "https://files.pythonhosted.org/packages/c6/1a/f4aee670d5670e9e148e0c82c7db98d780be566c6e6a97ee8035528ca0b3/pydantic_core-2.46.4-cp313-cp3
13-win_arm64.whl", hash = "sha256:184c081504d17f1c1066e430e117142b2c77d9448a97f7b65c6ac9fd9aee238d", size = 2027411, upload-time = "2026-05-06T13:40:4
5.796Z" },
    { url = "https://files.pythonhosted.org/packages/8d/74/228a26ddad29c6672b805d9fd78e8d251cd04004fa7eed0e622096cd0250/pydantic_core-2.46.4-cp314-cp3
14-macosx_10_12_x86_64.whl", hash = "sha256:428e04521a40150c85216fc8b85e8d39fece235a9cf5e383761238c7fa9b96fb", size = 2102079, upload-time = "2026-05-
06T13:38:41.019Z" },
    { url = "https://files.pythonhosted.org/packages/ad/1f/8970b150a4b4365623ae00fc88603491f763c627311ae8031e3111356d6e/pydantic_core-2.46.4-cp314-cp3
14-macosx_11_0_arm64.whl", hash = "sha256:23ace664830ee0bfe014a0c7bc248b1f7f25ed7ad103852c317624a1083af462", size = 1952179, upload-time = "2026-05-06
T13:36:59.812Z" },
    { url = "https://files.pythonhosted.org/packages/95/30/5211a831ae054928054b2f79731661087a2bc5c01e825c672b3a4a8f1b3e/pydantic_core-2.46.4-cp314-cp3
14-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:ce5c1d2a8b27468f433ca974829c44060b8097eedc39933e3c206a90ee49c4a9", size = 1978926
, upload-time = "2026-05-06T13:37:39.933Z" },
    { url = "https://files.pythonhosted.org/packages/57/e9/689668733b1eb67adeef047db3c2e8788fcf65a7fd9c9e2b46b7744fe245/pydantic_core-2.46.4-cp314-cp3
14-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:7283d57845ecf5a163403eb0702dfc220cc4fbdd18919cb5ccea4f95ee1cdab4", size = 2046785,
upload-time = "2026-05-06T13:38:01.995Z" },
    { url = "https://files.pythonhosted.org/packages/60/d9/6715260422ff50a2109878fd24d948a6c3446bb2664f34ee78cd972b3acd/pydantic_core-2.46.4-cp314-cp3
14-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:8daafc69c93ee8a0204506a3b6b30f586ef54028f52aeeeb5c4cfc5184fd5914", size = 2228733
, upload-time = "2026-05-06T13:40:50.371Z" },
    { url = "https://files.pythonhosted.org/packages/18/ae/fdb2f64316afca925640f8e70bb1a564b0ec2721c1389e25b8eb4bf9a299/pydantic_core-2.46.4-cp314-cp3
14-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:cd2213145bcc2ba85884d0ac63d222fece920678f77b9b4d76f054c561adb28", size = 2307534, up
load-time = "2026-05-06T13:37:21.531Z" },
    { url = "https://files.pythonhosted.org/packages/89/1d/8eff589b45bb8190a9d12c49cfad0f176a5cbd1534908a6b5125e2886239/pydantic_core-2.46.4-cp314-cp3
14-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:7a5f930472650a82629163023e630d160863fce524c616f4e5186e5de9d9a49b", size = 2099732,
upload-time = "2026-05-06T13:39:31.942Z" },
    { url = "https://files.pythonhosted.org/packages/06/d5/ee5a3366637fee41dee51a1fc91562dcf12ddbc68fda34e6b253da2324bb/pydantic_core-2.46.4-cp314-cp3
14-manylinux_2_31_riscv64.whl", hash = "sha256:c1b3f518abeca3aa13c712fd202306e145abf59a18b094a6bafb2d2bbf59192c", size = 2129627, upload-time = "2026-
05-06T13:37:25.033Z" },
    { url = "https://files.pythonhosted.org/packages/94/33/2414be571d2c6a6c4d08be21f9292b6d3fdb08949a97b6dfe985017821db/pydantic_core-2.46.4-cp314-cp3
14-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:1a7dd0b3ee80d90150e3495a3a13ac34dbcbfd4f012996a6a1d8900e91b5c0fb", size = 2179141, upload-t
ime = "2026-05-06T13:37:14.046Z" },
    { url = "https://files.pythonhosted.org/packages/7b/79/7daa95be995be0eecc4cf75064cb33f9bbbfe3fe0158caf2f0d4a996a5c7/pydantic_core-2.46.4-cp314-cp3
14-musllinux_1_1_aarch64.whl", hash = "sha256:3fb702cd90b0446a3a1c5e470bfa0dd23c0233b676a9099ddcc964fa6ca13898", size = 2184325, upload-time = "2026-0
5-06T13:36:53.615Z" },
    { url = "https://files.pythonhosted.org/packages/9f/cb/d0a382f5c0de8a222dc61c65348e0ce831b1f68e0a018450d31c2cace3a5/pydantic_core-2.46.4-cp314-cp3
14-musllinux_1_1_armv7l.whl", hash = "sha256:b8458003118a712e66286df6a707db01c52c0f52f7db8e4a38f0da1d3b94fc4e", size = 2323990, upload-time = "2026-05
-06T13:40:29.971Z" },
    { url = "https://files.pythonhosted.org/packages/05/db/d9ba624cc4a5aced1598e88c04fdbd8310c8a69b9d38b9a3d39ce3a61ed7/pydantic_core-2.46.4-cp314-cp3
14-musllinux_1_1_x86_64.whl", hash = "sha256:372429a130e469c9cd698925ce5fc50940b7a1336b0d82038e63d5bbc4edc519", size = 2369978, upload-time = "2026-05
-06T13:37:23.027Z" },
    { url = "https://files.pythonhosted.org/packages/f2/20/d15df15ba918c423461905802bfd2981c3af0bfa0e40d05e13edbfa48bc3/pydantic_core-2.46.4-cp314-cp3
14-win32.whl", hash = "sha256:85bb3611ff1802f3ee7fdd7dbff26b56f343fb432d57a4728fdd49b6ef35e2f4", size = 1966354, upload-time = "2026-05-06T13:38:03.49
9Z" },
    { url = "https://files.pythonhosted.org/packages/fc/b6/6b8de4c0a7d7ab3004c439c80c5c1e0a3e8d78bbae19379b01960383d9e5/pydantic_core-2.46.4-cp314-cp3
14-win_amd64.whl", hash = "sha256:811ff8e9c313ab425368bcbb36e5c4ebd7108c2bbf4e4089cfbb0b01eff63fac", size = 2072238, upload-time = "2026-05-06T13:39:4
0.807Z" },
    { url = "https://files.pythonhosted.org/packages/32/36/51eb763beec1f4cf59b1db243a7dcc39cbb41230f050a09b9d69faaf0a48/pydantic_core-2.46.4-cp314-cp3
14-win_arm64.whl", hash = "sha256:bfec22eab3c8cc2ceec0248aec886624116dc079afa027ecc8ad4a7e62010f8a", size = 2018251, upload-time = "2026-05-06T13:37:2
6.72Z" },
    { url = "https://files.pythonhosted.org/packages/e8/91/855af51d625b23aa987116a19e231d2aaef9c4a415273ddc189b79a45fee/pydantic_core-2.46.4-cp314-cp3
14t-macosx_10_12_x86_64.whl", hash = "sha256:af8244b2bef6aaad6d92cda81372de7f8c8d36c9f0c3ea36e827c60e7d9467a0", size = 2099593, upload-time = "2026-05
-06T13:39:47.682Z" },
    { url = "https://files.pythonhosted.org/packages/fb/1b/8784a54c65edb5f49f0a14d6977cf1b209bba85a4c77445b255c2de58ab3/pydantic_core-2.46.4-cp314-cp3
14t-macosx_11_0_arm64.whl", hash = "sha256:5a4330cdbc57162e4b3aa303f588ba752257694c9c9be3e7ebb11b4aca659b5d", size = 1935226, upload-time = "2026-05-0
6T13:40:40.428Z" },
    { url = "https://files.pythonhosted.org/packages/e8/e7/1955d28d1afc56dd4b3ad7cc0cf39df1b9852964cf16e5d13912756d6d6b/pydantic_core-2.46.4-cp314-cp3
14t-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:29c61fc04a3d840155ff08e475a04809278972fe6aef51e2720554e96367e34b", size = 197460
5, upload-time = "2026-05-06T13:37:32.029Z" },
    { url = "https://files.pythonhosted.org/packages/93/e2/3fedbf0ba7a22850e6e9fd78117f1c0f10f950182344d8a6c535d468fdd8/pydantic_core-2.46.4-cp314-cp3
14t-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:c50f2528cf200c5eed56faf3f4e22fcd5f38c157a8b78576e6ba3168ec35f000", size = 2030777,
 upload-time = "2026-05-06T13:38:55.239Z" },
    { url = "https://files.pythonhosted.org/packages/f8/61/46be275fcaaba0b4f5b9669dd852267ce1ff616592dccf7a7845588df091/pydantic_core-2.46.4-cp314-cp3
14t-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:0cbe8b01f948de4286c74cdd6c667aceb38f5c1e26f0693b3983d9d74887c65e", size = 223664
1, upload-time = "2026-05-06T13:37:08.096Z" },
    { url = "https://files.pythonhosted.org/packages/60/db/12e93e46a8bac9988be3c016860f83293daea8c716c029c9ace279036f2f/pydantic_core-2.46.4-cp314-cp3
14t-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:617d7e2ca7dcb8c5cf6bcb8c59b8832c94b36196bbf1cbd1bfb56ed341905edd", size = 2286404, u
pload-time = "2026-05-06T13:40:20.221Z" },
    { url = "https://files.pythonhosted.org/packages/e2/4a/4d8b19008f38d31c53b8219cfedc2e3d5de5fe99d90076b7e767de29274f/pydantic_core-2.46.4-cp314-cp3
14t-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:7027560ee92211647d0d34e3f7cd6f50da56399d26a9c8ad0da286d3869a53f3", size = 2109219,
 upload-time = "2026-05-06T13:38:12.153Z" },
    { url = "https://files.pythonhosted.org/packages/88/70/3cbc40978fefb7bb09c6708d40d4ad1a5d70fd7213c3d17f971de868ec1f/pydantic_core-2.46.4-cp314-cp3
14t-manylinux_2_31_riscv64.whl", hash = "sha256:f99626688942fb746e545232e7726926f3be91b5975f8b55327665fafda991c7", size = 2110594, upload-time = "2026
-05-06T13:40:02.971Z" },
    { url = "https://files.pythonhosted.org/packages/9d/20/b8d36736216e29491125531685b2f9e61aa5b4b2599893f8268551da3338/pydantic_core-2.46.4-cp314-cp3
```

```
14t-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:fc3e9034a63de20e15e8ade85358bc6efc614008cab72898b4b4952bea0509ff", size = 2159542, upload-
time = "2026-05-06T13:39:27.506Z" },
    { url = "https://files.pythonhosted.org/packages/1d/a2/367df868eb584dacf6bf82a389272406d7178e301c4ac82545ab98bc2dd9/pydantic_core-2.46.4-cp314-cp3
14t-musllinux_1_1_aarch64.whl", hash = "sha256:97e7cf2be5c77b7d1a9713a05605d49460d02c6078d38d8bef3cbe323c548424", size = 2168146, upload-time = "2026-
05-06T13:38:31.93Z" },
    { url = "https://files.pythonhosted.org/packages/c1/b8/4460f77f7e201893f649a29ab355dddd3beee8a97bcb1a320db414f9a06e/pydantic_core-2.46.4-cp314-cp3
14t-musllinux_1_1_armv7l.whl", hash = "sha256:3bf92c5d0e00fefaab325a4d27828fe6b6e2a21848686b5b60d2d9eeb09d76c6", size = 2306309, upload-time = "2026-0
5-06T13:37:44.717Z" },
    { url = "https://files.pythonhosted.org/packages/64/c4/be2639293acd87dc8ddbcec41a73cee9b2ebf996fe6d892a1a74e88ad3f7/pydantic_core-2.46.4-cp314-cp3
14t-musllinux_1_1_x86_64.whl", hash = "sha256:3ecbc122d18468d06ca279dc26a8c2e2d5acb10943bb35e36ae92096dc3b5565", size = 2369736, upload-time = "2026-0
5-06T13:37:05.645Z" },
    { url = "https://files.pythonhosted.org/packages/30/a6/9f9f380dbb301f67023bf8f707aaa75daadf84f7152d95c410fd7e81d994/pydantic_core-2.46.4-cp314-cp3
14t-win32.whl", hash = "sha256:e846ae7835bf0703ae43f534ab79a867146dadd59dc9ca5c8b53d5c8f7c9ef02", size = 1955575, upload-time = "2026-05-06T13:38:51.1
16Z" },
    { url = "https://files.pythonhosted.org/packages/40/1f/f1eb9eb350e795d1af8586289746f5c5677d16043040d63710e22abc43c9/pydantic_core-2.46.4-cp314-cp3
14t-win_amd64.whl", hash = "sha256:2108ba5c1c1eca18030634489dc544844144ee36357f2f9f780b93e7ddbb44b5", size = 2051624, upload-time = "2026-05-06T13:38:
21.672Z" },
    { url = "https://files.pythonhosted.org/packages/f6/d2/42dd53d0a85c27606f316d3aa5d2869c4e8470a5ed6dec30e4a1abe19192/pydantic_core-2.46.4-cp314-cp3
14t-win_arm64.whl", hash = "sha256:4fcbe087dbc2068af7eda3aa87634eba216dbda64d1ae73c8684b621d33f6596", size = 2017325, upload-time = "2026-05-06T13:40:
52.723Z" },
    { url = "https://files.pythonhosted.org/packages/9d/1d/8987ad40f65ae1432753072f214fb5c74fe47ffbd0698bb9cbbb585664f8/pydantic_core-2.46.4-graalpy31
2-graalpy250_312_native-macosx_10_12_x86_64.whl", hash = "sha256:1d8ba486450b14f3b1d63bc521d410ec7565e52f887b9fb671791886436a42f7", size = 2095527, up
load-time = "2026-05-06T13:39:52.283Z" },
    { url = "https://files.pythonhosted.org/packages/64/d3/84c282a7eee1d3ac4c0377546ef5a1ea436ce26840d9ac3b7ed54a377507/pydantic_core-2.46.4-graalpy31
2-graalpy250_312_native-macosx_11_0_arm64.whl", hash = "sha256:3009f12e4e90b7f88b4f9adb1b0c4a3d58fe7820f3238c190047209d148026df", size = 1936024, uplo
ad-time = "2026-05-06T13:40:15.671Z" },
    { url = "https://files.pythonhosted.org/packages/d7/ca/eac61596cdeb4d7e174d3dc0bd8a6238f14f75f97a24e7b7db4c7e7340a0/pydantic_core-2.46.4-graalpy31
2-graalpy250_312_native-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:ad785e92e6dc634c21555edc8bd6b64957ab844541bcb96a1366c202951a
e526", size = 1990696, upload-time = "2026-05-06T13:38:34.717Z" },
    { url = "https://files.pythonhosted.org/packages/fa/c3/7c8b240552251faf6b3a957db200fcfbbcec36763c050428b601e0c9b83b/pydantic_core-2.46.4-graalpy31
2-graalpy250_312_native-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:00c603d540afdd6b80eb39f078f33ebd46211f02f33e34a32d9f053bba711d
e0", size = 2147590, upload-time = "2026-05-06T13:39:29.883Z" },
]

[[package]]
name = "pydantic-settings"
version = "2.14.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "pydantic" },
    { name = "python-dotenv" },
    { name = "typing-inspection" },
]
sdist = { url = "https://files.pythonhosted.org/packages/5c/b5/8f48e906c3e0205276e8bd8cb7512217a87b2685304d64be27cad5b3019f/pydantic_settings-2.14.2.t
ar.gz", hash = "sha256:c19dd64b19097f1de80184f0cc7b0272a13ae6e170cbf240a3e27e381ed14a5f", size = 237700, upload-time = "2026-06-19T13:44:56.324Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/77/c1/6e422f34e569cf8e18df68d1939c81c099d2b61e4f7d9621c8a77560799c/pydantic_settings-2.14.2-py3-n
one-any.whl", hash = "sha256:a20c97b37910b6550d5ea50fbcc2d4187defe58cd57070b73863d069419c9440", size = 61715, upload-time = "2026-06-19T13:44:55.02Z"
},
]

[[package]]
name = "pygments"
version = "2.20.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/c3/b2/bc9c9196916376152d655522fdcebac55e66de6603a76a02bca1b6414f6c/pygments-2.20.0.tar.gz", h
ash = "sha256:6757cd03768053ff99f3039c1a36d6c0aa0b263438fcab17520b30a303a82b5f", size = 4955991, upload-time = "2026-03-29T13:29:33.898Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/f4/7e/a72dd26f3b0f4f2bf1dd8923c85f7ceb43172af56d63c7383eb62b332364/pygments-2.20.0-py3-none-any.w
hl", hash = "sha256:81a9e26dd42fd28a23a2d169d86d7ac03b46e2f8b59ed4698fb4785f946d0176", size = 1231151, upload-time = "2026-03-29T13:29:30.038Z" },
]

[[package]]
name = "pyjwt"
version = "2.13.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/3b/81/58d0ac84e1ef3a3843791d6954d94c0b33d526c75eeb1efbce9d0a4c4077/pyjwt-2.13.0.tar.gz", hash
 = "sha256:41571c89ca91598c79e8ef18a2d07367d4810fbbd6f637794879baf1b7703423", size = 107515, upload-time = "2026-05-21T19:54:36.618Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a3/5e/ecf12fdb62546d64385c158514e9b2b671f7832108ef2ecd2020ce0af2d1/pyjwt-2.13.0-py3-none-any.whl"
, hash = "sha256:66adcc2aff09b3f1bbd95fc1e1577df8ac8723c978552fd43304c8a290ac5728", size = 31274, upload-time = "2026-05-21T19:54:35.362Z" },
]

[package.optional-dependencies]
crypto = [
    { name = "cryptography" },
]

[[package]]
name = "pytest"
version = "9.1.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "colorama", marker = "sys_platform == 'win32'" },
    { name = "iniconfig" },
    { name = "packaging" },
    { name = "pluggy" },
    { name = "pygments" },
]
sdist = { url = "https://files.pythonhosted.org/packages/e4/47/b9efed96c114afcfa3c9d3fe98a76a1d14c74a9e266d397cf6eb64be5e01/pytest-9.1.1.tar.gz", hash
 = "sha256:1088fbde8f2b49d95a549a195707afa7a76a3ce9bcadc26b6d71f0ffda5fe313", size = 1636369, upload-time = "2026-06-19T10:58:32.857Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/24/25/1de2678b631f5a49215c6c96fff41ba892b0a34df68d6d80292b1b48aa7f/pytest-9.1.1-py3-none-any.whl"
, hash = "sha256:37a86b45efb9a47a61a36449063e8e18d0cab3161329fc099eb21783169c4f0c", size = 386536, upload-time = "2026-06-19T10:58:31.347Z" },
]

[[package]]
name = "pytest-asyncio"
version = "1.4.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "pytest" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/43/7c/d36d04db312ecf4298932ef77e6e4a9e8ad017906e24e34f0b0c361a2473/pytest_asyncio-1.4.0.tar.g
```

```
z", hash = "sha256:c6c0d2259945122819f171a32ecea2c349ead889ee28176caaf492143424be42", size = 58514, upload-time = "2026-05-26T09:56:04.083Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/03/e2/08a497ef684b88559c9cc5f4ad53a37e7b99e727094a86d6ea32536d5d3c/pytest_asyncio-1.4.0-py3-none-any.whl", hash = "sha256:933ca923a23075a87fb7070c0ec272a6848489824d887c85c812670932835aa1", size = 16930, upload-time = "2026-05-26T09:56:02.576Z" },
]

[[package]]
name = "pytest-cov"
version = "7.1.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "coverage" },
    { name = "pluggy" },
    { name = "pytest" },
]
sdist = { url = "https://files.pythonhosted.org/packages/b1/51/a849f96e117386044471c8ec2bd6cfebacda285da9525c9106aeb28da671/pytest_cov-7.1.0.tar.gz", hash = "sha256:30674f2b5f6351aa09702a9c8c364f6a01c27aae0c1366ae8016160d1efc56b2", size = 55592, upload-time = "2026-03-21T20:11:16.284Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/9d/7a/d968e294073affff457b041c2be9868a40c1c71f4a35fcc1e45e5493067b/pytest_cov-7.1.0-py3-none-any.whl", hash = "sha256:a0461110b7865f9a271aa1b51e516c9a95de9d696734a2f71e3e78f46e1d4678", size = 22876, upload-time = "2026-03-21T20:11:14.438Z" },
]

[[package]]
name = "python-dateutil"
version = "2.9.0.post0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "six" },
]
sdist = { url = "https://files.pythonhosted.org/packages/66/c0/0c8b6ad9f17a802ee498c46e004a0eb49bc148f2fd230864601a86dcf6db/python-dateutil-2.9.0.post0.tar.gz", hash = "sha256:37dd54208da7e1cd875388217d5e00ebd4179249f90fb72437e91a35459a0ad3", size = 342432, upload-time = "2024-03-01T18:36:20.211Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/ec/57/56b9bcc3c9c6a792fcbaf139543cee77261f3651ca9da0c93f5c1221264b/python_dateutil-2.9.0.post0-py2.py3-none-any.whl", hash = "sha256:a8b2bc7bffae282281c8140a97d3aa9c14da0b136dfe83f850eea9a5f7470427", size = 229892, upload-time = "2024-03-01T18:36:18.57Z" },
]

[[package]]
name = "python-dotenv"
version = "1.2.2"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/82/ed/0301aeeac3e5353ef3d94b6ec08bbcabd04a72018415dcb29e588514bba8/python_dotenv-1.2.2.tar.gz", hash = "sha256:2c371a91fbd7ba082c2c1dc1f8bf89ca22564a087c2c287cd9b662adde799cf3", size = 50135, upload-time = "2026-03-01T16:00:26.196Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/0b/d7/1959b9648791274998a9c3526f6d0ec8fd2233e4d4acce81bbae76b44b2a/python_dotenv-1.2.2-py3-none-any.whl", hash = "sha256:1d8214789a24de455a8b8bd8ae6fe3c6b69a5e3d64aa8a8e5d68e694bbcb285a", size = 22101, upload-time = "2026-03-01T16:00:25.09Z" },
]

[[package]]
name = "python-jose"
version = "3.5.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "ecdsa" },
    { name = "pyasn1" },
    { name = "rsa" },
]
sdist = { url = "https://files.pythonhosted.org/packages/c6/77/3a1c9039db7124eb039772b935f2244fbb73fc8ee65b9acf2375da1c07bf/python_jose-3.5.0.tar.gz",
 hash = "sha256:fb4eaa44dbeb1c26dcc69e4bd7ec54a1cb8dd64d3b4d81ef08d90ff453f2b01b", size = 92726, upload-time = "2025-05-28T17:31:54.288Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d9/c3/0bd11992072e6a1c513b16500a5d07f91a24017c5909b02c72c62d7ad024/python_jose-3.5.0-py2.py3-none-any.whl", hash = "sha256:abd1202f23d34dfad2c3d28cb8617b90acf34132c7afd60abd0b0b7d3cb55771", size = 34624, upload-time = "2025-05-28T17:31:52.802Z" },
]

[package.optional-dependencies]
cryptography = [
    { name = "cryptography" },
]

[[package]]
name = "python-multipart"
version = "0.0.32"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/5b/42/55c32bb9b12693c092ad250a0e82edb5b31ddeda6eb772de5f308b3804ad/python_multipart-0.0.32.tar.gz", hash = "sha256:be54b7f3fa167bb83e4fcd936b887b708f4e57fe75911c02aebf53efaf8d938e", size = 46881, upload-time = "2026-06-04T16:18:58.647Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e1/04/e8135ebd1ad02c56ec633277529b2602ff99ff634be76cdba5744cf554fd/python_multipart-0.0.32-py3-none-any.whl", hash = "sha256:ff6d3f776f16878c894e52e107296ffc890e913c611b1a4ec6c44e2821fe2e23", size = 30042, upload-time = "2026-06-04T16:18:57.319Z" },
]

[[package]]
name = "pywin32"
version = "312"
source = { registry = "https://pypi.org/simple" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/83/ff/32aa7d2ed0ab12b323aaa64f9b75e6ad4f8fd09f9ccfc28c79414d46838d/pywin32-312-cp312-cp312-win32.whl", hash = "sha256:dab4f65ac9c4e48400a2a0530c46c3c579cd5905ecd11b806923739152692 08b", size = 6371877, upload-time = "2026-06-04T07:49:28.836Z" },
    { url = "https://files.pythonhosted.org/packages/03/d9/77040d3b43df3f3be32ea289433d660d2727f5ba327bc73be835127d9d60/pywin32-312-cp312-cp312-win_amd64.whl", hash = "sha256:b457f6d628a47e8a7346ce22acb7e1a46a4a78b52e1d17e1af56871bd19a93bc", size = 6914841, upload-time = "2026-06-04T07:49:31.85Z" },
    { url = "https://files.pythonhosted.org/packages/e3/cc/7b1ec671775756020a0ee7f4feeaf3c568f0ab86bd3900088cf986937a92/pywin32-312-cp312-cp312-win_arm64.whl", hash = "sha256:6017c58e12f6809fbb0555b75df144c2922a9ffd18e4b9b5afa863b6c1a9d950", size = 6727901, upload-time = "2026-06-04T07:49:34.244Z" },
    { url = "https://files.pythonhosted.org/packages/2d/41/12fbfd7f36ed2146d8bc9de96c2741296bf0d490b98508496cff322e274c/pywin32-312-cp313-cp313-win32.whl", hash = "sha256:7a27df850933d16a8eabfbaeb73d52b273e2da667f80d70b01a89d1f6828d02c", size = 6370184, upload-time = "2026-06-04T07:49:36.253Z" },
    { url = "https://files.pythonhosted.org/packages/ba/db/36a78e3403099d31d9746d13fdcde5accc43c1155f375a34d15983a479a7/pywin32-312-cp313-cp313-win_amd64.whl", hash = "sha256:c53e878d15a1c44788082bfe712a905433473aa38f86375b7cf8b45e3acbaaf9", size = 6914298, upload-time = "2026-06-04T07:49:38.876Z" },
    { url = "https://files.pythonhosted.org/packages/84/37/c1697194092b76de9ed47ca124323f02c57ffc8a45c06f88a3d5acaf01eb/pywin32-312-cp313-cp313-win_arm64.whl", hash = "sha256:59aba5d5940842075343a5ddc6b11f1cdf0d1567fe745290359dfbcc7c2eb831", size = 6727640, upload-time = "2026-06-04T07:49:41.083Z" },
    { url = "https://files.pythonhosted.org/packages/fc/2b/1f3cded5822fd49c02f40544cbb5f58c7cfd6b1694869fd476cb6170ee97/pywin32-312-cp314-cp314-win32.whl", hash = "sha256:a77a90fbb6881238d2ca9c6fd797b25817f3768fe78d214a90137ff055a75f5b", size = 6468928, upload-time = "2026-06-04T07:49:43.188Z" },
    { url = "https://files.pythonhosted.org/packages/21/82/3bf86d2e2808902013132e1ce905a7da0da53790f3836c64bf44d55e24f3/pywin32-312-cp314-cp314-win_amd64.whl", hash = "sha256:a4dd3a848290ef724347b19f301045831d8e802fa4464f491b98b1e0a081432e", size = 7024157, upload-time = "2026-06-04T07:49:45.34Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/a4/0e/73f6d6800b4f27655abd9e9f6aaeaefcddb2b946e4674efa2bab184a7f7b/pywin32-312-cp314-cp314-win_ar
m64.whl", hash = "sha256:9fce94568364e0155e6dfb781ac5d95903be8baf28670632beab1b523f300daa", size = 6839598, upload-time = "2026-06-04T07:49:47.613Z" }
,
    { url = "https://files.pythonhosted.org/packages/eb/61/caa39686032d2ebdd04ff0ab5cbe163126c0066d98e00c9018646e42393b/pywin32-312-cp315-cp315-win32.
whl", hash = "sha256:5c1fbe4a937a73ae9297384a3da38518cbc694c68ad8a809b2e19acd350f03ed", size = 6471159, upload-time = "2026-06-04T07:49:50.035Z" },
    { url = "https://files.pythonhosted.org/packages/0f/cd/7e1de64a4a6f69c04214169657ccab0d93a670ea50e35eb8f489d7378249/pywin32-312-cp315-cp315-win_am
d64.whl", hash = "sha256:c2f03a0f73f804a13c2735b99392b0cd426bb4f2c4d0178e5ac966a0f21618d5", size = 7025293, upload-time = "2026-06-04T07:49:54.857Z" }
,
    { url = "https://files.pythonhosted.org/packages/23/ed/4532e9388e65fa16b46776ef47ad631a64eda1631884488af707666350ed/pywin32-312-cp315-cp315-win_ar
m64.whl", hash = "sha256:a8597d28f267b39074aef51fa593530082b39cbe5a074226096857b1fed2dfb9", size = 6840337, upload-time = "2026-06-04T07:49:57.531Z" }
,
]

[[package]]
name = "pyyaml"
version = "6.0.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/05/8e/961c0007c59b8dd7729d542c61a4d537767a59645b82a0b521206e1e25c2/pyyaml-6.0.3.tar.gz", hash
 = "sha256:d76623373421df22fb4cf8817020cbb7ef15c725b9d5e45f17e189bfc384190f", size = 130960, upload-time = "2025-09-25T21:33:16.546Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d1/33/422b98d2195232ca1826284a76852ad5a86fe23e31b009c9886b2d0fb8b2/pyyaml-6.0.3-cp312-cp312-macos
x_10_13_x86_64.whl", hash = "sha256:7f047e29dcae44602496db43be01ad42fc6f1cc0d8cd6c83d342306c32270196", size = 182063, upload-time = "2025-09-25T21:32:
11.445Z" },
    { url = "https://files.pythonhosted.org/packages/89/a0/6cf41a19a1f2f3feab0e9c0b74134aa2ce6849093d5517a0c550fe37a648/pyyaml-6.0.3-cp312-cp312-macos
x_11_0_arm64.whl", hash = "sha256:fc09d0aa354569bc501d4e787133afc08552722d3ab34836a80547331bb5d4a0", size = 173973, upload-time = "2025-09-25T21:32:12
.492Z" },
    { url = "https://files.pythonhosted.org/packages/ed/23/7a778b6bd0b9a8039df8b1b1d80e2e2ad78aa04171592c8a5c43a56a6af4/pyyaml-6.0.3-cp312-cp312-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:9149cad251584d5fb4981be1ecde53a1ca46c891a79788c0df828d2f166bda28",
 size = 775116, upload-time = "2025-09-25T21:32:13.652Z" },
    { url = "https://files.pythonhosted.org/packages/65/30/d7353c338e12baef4ecc1b09e877c1970bd3382789c159b4f89d6a70dc09/pyyaml-6.0.3-cp312-cp312-manyl
inux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:5fdec68f91a0c6739b380c83b951e2c72ac0197ace422360e6d5a959d8d97b2c", size
= 844011, upload-time = "2025-09-25T21:32:15.21Z" },
    { url = "https://files.pythonhosted.org/packages/8b/9d/b3589d3877982d4f2329302ef98a8026e7f4443c765c46cfecc8858c6b4b/pyyaml-6.0.3-cp312-cp312-manyl
inux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:ba1cc08a7ccde2d2ec775841541641e4548226580ab850948cbfda66a1befcdc", si
ze = 807870, upload-time = "2025-09-25T21:32:16.431Z" },
    { url = "https://files.pythonhosted.org/packages/05/c0/b3be26a015601b822b97d9149ff8cb5ead58c66f981e04fedf4e762f4bd4/pyyaml-6.0.3-cp312-cp312-musll
inux_1_2_aarch64.whl", hash = "sha256:8dc52c23056b9ddd46818a57b78404882310fb473d63f17b07d5c40421e47f8e", size = 761089, upload-time = "2025-09-25T21:3
2:17.56Z" },
    { url = "https://files.pythonhosted.org/packages/be/8e/98435a21d1d4b46590d5459a22d88128103f8da4c2d4cb8f14f2a96504e1/pyyaml-6.0.3-cp312-cp312-musll
inux_1_2_x86_64.whl", hash = "sha256:41715c910c881bc081f1e8872880d3c650acf13dfa8214bad49ed4cede7c34ea", size = 790181, upload-time = "2025-09-25T21:32
:18.834Z" },
    { url = "https://files.pythonhosted.org/packages/74/93/7baea19427dcfbe1e5a372d81473250b379f04b1bd3c4c5ff825e2327202/pyyaml-6.0.3-cp312-cp312-win32
.whl", hash = "sha256:96b533f0e99f6579b3d4d4995707cf36df9100d67e0c8303a0c55b27b5f99bc5", size = 137658, upload-time = "2025-09-25T21:32:20.209Z" },
    { url = "https://files.pythonhosted.org/packages/86/bf/899e81e4cce32febab4fb42bb97dcdf66bc135272882d1987881a4b519e9/pyyaml-6.0.3-cp312-cp312-win_a
md64.whl", hash = "sha256:5fcd34e47f6e0b794d17de1b4ff496c00986e1c83f7ab2fb8fcfe9616ff7477b", size = 154003, upload-time = "2025-09-25T21:32:21.167Z" }
,
    { url = "https://files.pythonhosted.org/packages/1a/08/67bd04656199bbb51dbed1439b7f27601dfb576fb864099c7ef0c3e55531/pyyaml-6.0.3-cp312-cp312-win_a
rm64.whl", hash = "sha256:64386e5e707d03a7e172c0701abfb7e10f0fb753ee1d773128192742712a98fd", size = 140344, upload-time = "2025-09-25T21:32:22.617Z" }
,
    { url = "https://files.pythonhosted.org/packages/d1/11/0fd08f8192109f7169db964b5707a2f1e8b745d4e239b784a5a1dd80d1db/pyyaml-6.0.3-cp313-cp313-macos
x_10_13_x86_64.whl", hash = "sha256:8da9669d359f02c0b91ccc01cac4a67f16afec0dac22c2ad09f46bee0697eba8", size = 181669, upload-time = "2025-09-25T21:32:
23.673Z" },
    { url = "https://files.pythonhosted.org/packages/b1/16/95309993f1d3748cd644e02e38b75d50cbc0d9561d21f390a76242ce073f/pyyaml-6.0.3-cp313-cp313-macos
x_11_0_arm64.whl", hash = "sha256:2283a07e2c21a2aa78d9c4442724ec1eb15f5e42a723b99cb3d822d48f5f7ad1", size = 173252, upload-time = "2025-09-25T21:32:25
.149Z" },
    { url = "https://files.pythonhosted.org/packages/50/31/b20f376d3f810b9b2371e72ef5adb33879b25edb7a6d072cb7ca0c486398/pyyaml-6.0.3-cp313-cp313-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:ee2922902c45ae8ccada2c5b501ab86c36525b883eff4255313a253a3160861c",
 size = 767081, upload-time = "2025-09-25T21:32:26.575Z" },
    { url = "https://files.pythonhosted.org/packages/49/1e/a55ca81e949270d5d4432fbbd19dfea5321eda7c41a849d443dc92fd1ff7/pyyaml-6.0.3-cp313-cp313-manyl
inux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:a33284e20b78bd4a18c8c2282d549d10bc8408a2a7ff57653c0cf0b9be0afce5", size
= 841159, upload-time = "2025-09-25T21:32:27.727Z" },
    { url = "https://files.pythonhosted.org/packages/74/27/e5b8f34d02d9995b80abcef563ea1f8b56d20134d8f4e5e81733b1feceb2/pyyaml-6.0.3-cp313-cp313-manyl
inux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:0f29edc409a6392443abf94b9cf89ce99889a1dd5376d94316ae5145dfedd5d6", si
ze = 801626, upload-time = "2025-09-25T21:32:28.878Z" },
    { url = "https://files.pythonhosted.org/packages/f9/11/ba845c23988798f40e52ba45f34849aa8a1f2d4af4b798588010792ebad6/pyyaml-6.0.3-cp313-cp313-musll
inux_1_2_aarch64.whl", hash = "sha256:f7057c9a337546edc7973c0d3ba84ddcdf0daa14533c2065749c9075001090e6", size = 753613, upload-time = "2025-09-25T21:3
2:30.178Z" },
    { url = "https://files.pythonhosted.org/packages/3d/e0/7966e1a7bfc0a45bf0a7fb6b98ea03fc9b8d84fa7f2229e9659680b69ee3/pyyaml-6.0.3-cp313-cp313-musll
inux_1_2_x86_64.whl", hash = "sha256:eda16858a3cab07b80edaf74336ece1f986ba330fdb8ee0d6c0d68fe82bc96be", size = 794115, upload-time = "2025-09-25T21:32
:31.353Z" },
    { url = "https://files.pythonhosted.org/packages/de/94/980b50a6531b3019e45ddeada0626d45fa85cbe22300844a7983285bed3b/pyyaml-6.0.3-cp313-cp313-win32
.whl", hash = "sha256:d0eae10f8159e8fdad514efdc92d74fd8d682c933a6dd088030f3834bc8e6b26", size = 137427, upload-time = "2025-09-25T21:32:32.58Z" },
    { url = "https://files.pythonhosted.org/packages/97/c9/39d5b874e8b28845e4ec2202b5da735d0199dbe5b8fb85f91398814a9a46/pyyaml-6.0.3-cp313-cp313-win_a
md64.whl", hash = "sha256:79005a0d97d5ddabfeeea4cf676af11e647e41d81c9a7722a193022accdb6b7c", size = 154090, upload-time = "2025-09-25T21:32:33.659Z" }
,
    { url = "https://files.pythonhosted.org/packages/73/e8/2bdf3ca2090f68bb3d75b44da7bbc71843b19c9f2b9cb9b0f4ab7a5a4329/pyyaml-6.0.3-cp313-cp313-win_a
rm64.whl", hash = "sha256:5498cd1645aa724a7c71c8f378eb29ebe23da2fc0d7a08071d89469bf1d2defb", size = 140246, upload-time = "2025-09-25T21:32:34.663Z" }
,
    { url = "https://files.pythonhosted.org/packages/9d/8c/f4bd7f6465179953d3ac9bc44ac1a8a3e6122cf8ada906b4f96c60172d43/pyyaml-6.0.3-cp314-cp314-macos
x_10_13_x86_64.whl", hash = "sha256:8d1fab6bb153a416f9aeb4b8763bc0f22a5586065f86f7664fc23339fc1c1fac", size = 181814, upload-time = "2025-09-25T21:32:
35.712Z" },
    { url = "https://files.pythonhosted.org/packages/bd/9c/4d95bb87eb2063d20db7b60faa3840c1b18025517ae857371c4dd55a6b3a/pyyaml-6.0.3-cp314-cp314-macos
x_11_0_arm64.whl", hash = "sha256:34d5fcd24b8445fadc33f9cf348c1047101756fd760b4dacb5c3e99755703310", size = 173809, upload-time = "2025-09-25T21:32:36
.789Z" },
    { url = "https://files.pythonhosted.org/packages/92/b5/47e807c2623074914e29dabd16cbbdd4bf5e9b2db9f8090fa64411fc5382/pyyaml-6.0.3-cp314-cp314-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:501a031947e3a9025ed4405a168e6ef5ae3126c59f90ce0cd6f2bfc477be31b7",
 size = 766454, upload-time = "2025-09-25T21:32:37.966Z" },
    { url = "https://files.pythonhosted.org/packages/02/9e/e5e9b168be58564121efb3de6859c452fccde0ab093d8438905899a3a483/pyyaml-6.0.3-cp314-cp314-manyl
inux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:b3bc83488de33889877a0f2543ade9f70c67d66d9ebb4ac959502e12de895788", size
= 836355, upload-time = "2025-09-25T21:32:39.178Z" },
    { url = "https://files.pythonhosted.org/packages/88/f9/16491d7ed2a919954993e48aa941b200f38040928474c9e85ea9e64222c3/pyyaml-6.0.3-cp314-cp314-manyl
inux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:c458b6d084f9b935061bc36216e8a69a7e293a2f1e68bf956dcd9e6cbcd143f5", si
ze = 794175, upload-time = "2025-09-25T21:32:40.865Z" },
    { url = "https://files.pythonhosted.org/packages/dd/3f/5989debef34dc6397317802b527dbbafb2b4760878a53d4166579111411e/pyyaml-6.0.3-cp314-cp314-musll
inux_1_2_aarch64.whl", hash = "sha256:7c6610def4f163542a622a73fb39f534f8c101d690126992300bf3207eab9764", size = 755228, upload-time = "2025-09-25T21:3
2:42.084Z" },
    { url = "https://files.pythonhosted.org/packages/d7/ce/af88a49043cd2e265be63d083fc75b27b6ed062f5f9fd6cdc223ad62f03e/pyyaml-6.0.3-cp314-cp314-musll
inux_1_2_x86_64.whl", hash = "sha256:5190d403f121660ce8d1d2c1bb2ef1bd05b5f68533fc5c2ea899bd15f4399b35", size = 789194, upload-time = "2025-09-25T21:32
:43.362Z" },
    { url = "https://files.pythonhosted.org/packages/23/20/bb6982b26a40bb43951265ba29d4c246ef0ff59c9fdcdf0ed04e0687de4d/pyyaml-6.0.3-cp314-cp314-win_a
md64.whl", hash = "sha256:4a2e8cebe2ff6ab7d1050ecd59c25d4c8bd7e6f400f5f82b96557ac0abafd0ac", size = 156429, upload-time = "2025-09-25T21:32:57.844Z" }
,
    { url = "https://files.pythonhosted.org/packages/f4/f4/a4541072bb9422c8a883ab55255f918fa378ecf083f5b85e87fc2b4eda1b/pyyaml-6.0.3-cp314-cp314-win_a
rm64.whl", hash = "sha256:93dda82c9c22deb0a405ea4dc5f2d0cda384168e466364dec6255b293923b2f3", size = 143912, upload-time = "2025-09-25T21:32:59.247Z" }
,
```

```
    { url = "https://files.pythonhosted.org/packages/7c/f9/07dd09ae774e4616edf6cda684ee78f97777bdd15847253637a6f052a62f/pyyaml-6.0.3-cp314-cp314t-macosx_10_13_x86_64.whl", hash = "sha256:02893d100e99e03eda1c8fd5c441d8c60103fd175728e23e431db1b589cf5ab3", size = 189108, upload-time = "2025-09-25T21:32:44.377Z" },
    { url = "https://files.pythonhosted.org/packages/4e/78/8d08c9fb7ce09ad8c38ad533c1191cf27f7ae1effe5bb9400a46d9437fcf/pyyaml-6.0.3-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:c1ff362665ae507275af2853520967820d9124984e0f7466736aea23d8611fba", size = 183641, upload-time = "2025-09-25T21:32:45.407Z" },
    { url = "https://files.pythonhosted.org/packages/7b/5b/3babb19104a46945cf816d047db2788bcaf8c94527a805610b0289a01c6b/pyyaml-6.0.3-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:6adc77889b628398debc7b65c073bcb99c4a0237b248cacaf3fe8a557563ef6c", size = 831901, upload-time = "2025-09-25T21:32:48.83Z" },
    { url = "https://files.pythonhosted.org/packages/8b/cc/dff0684d8dc44da4d22a13f35f073d558c268780ce3c6ba1b87055bb0b87/pyyaml-6.0.3-cp314-cp314t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:a80cb027f6b349846a3bf6d73b5e95e782175e52f22108cfa17876aaeff93702", size = 861132, upload-time = "2025-09-25T21:32:50.149Z" },
    { url = "https://files.pythonhosted.org/packages/b1/5e/f77dc6b9036943e285ba76b49e118d9ea929885becb0a29ba8a7c75e29fe/pyyaml-6.0.3-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:00c4bdeba853cc34e7dd471f16b4114f4162dc03e6b7afcc2128711f0eca823c", size = 839261, upload-time = "2025-09-25T21:32:51.808Z" },
    { url = "https://files.pythonhosted.org/packages/ce/88/a9db1376aa2a228197c58b37302f284b5617f56a5d959fd1763fb1675ce6/pyyaml-6.0.3-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:66e1674c3ef6f541c35191caae2d429b967b99e02040f5ba928632d9a7f0f065", size = 805272, upload-time = "2025-09-25T21:32:52.941Z" },
    { url = "https://files.pythonhosted.org/packages/da/92/1446574745d74df0c92e6aa4a7b0b3130706a4142b2d1a5869f2eaa423c6/pyyaml-6.0.3-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:16249ee61e95f858e83976573de0f5b2893b3677ba71c9dd36b9cf8be9ac6d65", size = 829923, upload-time = "2025-09-25T21:32:54.537Z" },
    { url = "https://files.pythonhosted.org/packages/f0/7a/1c7270340330e575b92f397352af856a8c06f230aa3e76f86b39d01b416a/pyyaml-6.0.3-cp314-cp314t-win_amd64.whl", hash = "sha256:4ad1906908f2f5ae4e5a8ddfce73c320c2a1429ec52eafd27138b7f1cbe341c9", size = 174062, upload-time = "2025-09-25T21:32:55.767Z" },
    { url = "https://files.pythonhosted.org/packages/f1/12/de94a39c2ef588c7e6455cfbe7343d3b2dc9d6b6b2f40c4c6565744c873d/pyyaml-6.0.3-cp314-cp314t-win_arm64.whl", hash = "sha256:ebc55a14a21cb14062aa4162f906cd962b28e2e9ea38f9b4391244cd8de4ae0b", size = 149341, upload-time = "2025-09-25T21:32:56.828Z" },
]

[[package]]
name = "redis"
version = "8.1.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a8/99/604f0b666d4c616d891cf77ebb9db6bb21601344c051aebf1b72b9ff915f/redis-8.1.0.tar.gz", hash = "sha256:6e1a19beef9225c83efd689c7e6b7da2d5215b1f42cd13b7fc3714d0a09c7b25", size = 5254356, upload-time = "2026-07-30T08:51:00.269Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/66/9d/c5731f6e3608663d4d3656fd8d3aecee8b509c3082818f5a13eae925baea/redis-8.1.0-py3-none-any.whl", hash = "sha256:a4fe1aac3d3b3cc791d4b3d5931c5a956045dc951ee74d1c913ee3ac4d2ee9fb", size = 560618, upload-time = "2026-07-30T08:50:58.497Z" },
]

[[package]]
name = "referencing"
version = "0.37.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "attrs" },
    { name = "rpds-py" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/22/f5/df4e9027acead3ecc63e50fe1e36aca1523e1719559c499951bb4b53188f/referencing-0.37.0.tar.gz", hash = "sha256:44aefc3142c5b842538163acb373e24cce6632bd54bdb01b21ad5863489f50d8", size = 78036, upload-time = "2025-10-13T15:30:48.871Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/2c/58/ca301544e1fa93ed4f80d724bf5b194f6e4b945841c5bfd555878eea9fcb/referencing-0.37.0-py3-none-any.whl", hash = "sha256:381329a9f99628c9069361716891d34ad94af76e461dcb0335825aecc7692231", size = 26766, upload-time = "2025-10-13T15:30:47.625Z" },
]

[[package]]
name = "regex"
version = "2026.7.19"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/20/98/04b13f1ddfb63158025291c02e03eb42fbb7acb51d091d541050eb4e35e8/regex-2026.7.19.tar.gz", hash = "sha256:7e77b324909c1617cbb4c668677e2c6ae13f44d7c1de0d4f15f2e3c10f3315b5", size = 416440, upload-time = "2026-07-19T00:19:48.923Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/3b/b9/d11d7e501ac8fd7d617684423ebb9561e0b998481c1e4cbc0cb212c5d74a/regex-2026.7.19-cp312-cp312-macosx_10_13_universal2.whl", hash = "sha256:2cc3460cedf7579948486eab03bc9ad7089df4d7281c0f47f4afe03e8d13f02d", size = 496778, upload-time = "2026-07-19T00:17:05.677Z" },
    { url = "https://files.pythonhosted.org/packages/3f/a9/a5ab6f312f24318019170dc485d5421fe4f89e43a98640da50d95a8a7041/regex-2026.7.19-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:0e9554c8785eac5cffe6300f69a91f58ba72bc88a5f8d661235ad7c6aa5b8ccd", size = 297122, upload-time = "2026-07-19T00:17:07.59Z" },
    { url = "https://files.pythonhosted.org/packages/b3/63/4cab4d7f2d384a144d420b763d97674cb70619c878ea6fcd7640d0e62143/regex-2026.7.19-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:d7da47a0f248977f08e2cb659ff3c17ddc13a4d39b3a7baa0a81bf5b415430f6", size = 292009, upload-time = "2026-07-19T00:17:09.648Z" },
    { url = "https://files.pythonhosted.org/packages/22/85/102a81b218298957d4ea7d2f084fae537a71add9d6ff93c8e67284c5f45e/regex-2026.7.19-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:93db40c8de0815baab96a06e08a984bac71f989d13bab789e382158c5d426797", size = 796708, upload-time = "2026-07-19T00:17:11.542Z" },
    { url = "https://files.pythonhosted.org/packages/78/b5/dc136af5629938a037cd2b304c12240e132ec92f38be8ff9cc89af2a1f2d/regex-2026.7.19-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:66bd62c59a5427746e8c44becae1d9b99d22fb13f30f492083dfb9ad7c45cc18", size = 865651, upload-time = "2026-07-19T00:17:13.312Z" },
    { url = "https://files.pythonhosted.org/packages/e0/75/67402ae3cd9c8c988a4c805d15ee3eef015e7ca4cb112cf3e640fc1f4153/regex-2026.7.19-cp312-cp312-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:1649eb39fcc9ea80c4d2f110fde2b8ab2aef3877b98f02ab9b14e961f418c511", size = 911756, upload-time = "2026-07-19T00:17:15.015Z" },
    { url = "https://files.pythonhosted.org/packages/2a/8e/096d00c7c480ef2ff4265349b14e2261d4ab787ba1f74e2e80d1c58079c3/regex-2026.7.19-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9dce8ec9695f531a1b8a6f314fd4b393adcccf2ea861db480cdf97a301d01a68", size = 801798, upload-time = "2026-07-19T00:17:17.208Z" },
    { url = "https://files.pythonhosted.org/packages/f0/41/e7ecac6edb5722417f85cc67eaf386322fbe8acf6918ec2fdc37c20dd9d0/regex-2026.7.19-cp312-cp312-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:3080a7fd38ef049bd489e01c970c97dd84ff446a885b0f1f6b26d9b1ad13ce11", size = 776933, upload-time = "2026-07-19T00:17:19.347Z" },
    { url = "https://files.pythonhosted.org/packages/6f/69/03c9b3f058d66403e0ca2c938696e81d51cd4c6d47ec5265f02f96948d9a/regex-2026.7.19-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:1d793a7988e04fcb1e2e135567443d82173225d657419ec09414a9b5a145b986", size = 784338, upload-time = "2026-07-19T00:17:21.057Z" },
    { url = "https://files.pythonhosted.org/packages/f6/f7/b38ab3d43f284afbb618fcd15d0e77eb786ae461ce1f6bc7494619ddc0f2/regex-2026.7.19-cp312-cp312-musllinux_1_2_ppc64le.whl", hash = "sha256:e8b0abe7d870f53ca5143895fef7d1041a0c831a140d3dc2c760dd7ba25d4a8b", size = 860452, upload-time = "2026-07-19T00:17:23.119Z" },
    { url = "https://files.pythonhosted.org/packages/15/5c/ff60ef0571121714f3cf9920bc183071e384a10b556d042e0fdb06cc07a5/regex-2026.7.19-cp312-cp312-musllinux_1_2_riscv64.whl", hash = "sha256:4e5413bd5f13d3a4e3539ca98f70f75e7fca92518dd7f117f030ebedd10b60cb", size = 765958, upload-time = "2026-07-19T00:17:24.81Z" },
    { url = "https://files.pythonhosted.org/packages/aa/0f/bd34021162c0ab47f9a315bd56cd5642e920c8e5668a75ef6c6a6fca590d/regex-2026.7.19-cp312-cp312-musllinux_1_2_s390x.whl", hash = "sha256:73b133a9e6fb512858e7f065e96f1180aa46646bc74a83aea62f1d314f3dd035", size = 851765, upload-time = "2026-07-19T00:17:26.993Z" },
    { url = "https://files.pythonhosted.org/packages/2a/20/a2ca43edade0595cccfdc98636739f536d9e26898e7dbddc2b9e98898953/regex-2026.7.19-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:dbe6493fbd27321b1d1f2dd4f5c7e5bd4d8b1d7cab7f32fd67db3d0b2ed8248a", size = 789714, upload-time = "2026-07-19T00:17:28.699Z" },
    { url = "https://files.pythonhosted.org/packages/5d/47/e02db4015d424fc83c00ea0ac8c5e5ec14397943de9abf909d5ce3a25931/regex-2026.7.19-cp312-cp312-win32.whl", hash = "sha256:ddd67571c10869f65a5d7dde536d1e066e306cc90de57d7de4d5f34802428bb5", size = 267157, upload-time = "2026-07-19T00:17:31.051Z" },
    { url = "https://files.pythonhosted.org/packages/08/8e/c780c131f79b42ed22d1bd7da4096c2c35f813e835acd02ef0f018bd892c/regex-2026.7.19-cp312-cp312-wi
```

n_amd64.whl", hash = "sha256:e30d40268a28d54ce043703175049700 4c22602b8e3ab891f759b795a003b312", size = 277777, upload-time = "2026-07-19T00:17:32.848Z
" },
    { url = "https://files.pythonhosted.org/packages/3e/4c/e4d7e086449bdf379d89774bf1f89dc4a41943f3c5a6125a03905b34b5fb/regex-2.6.7.19-cp312-cp312-wi
n_arm64.whl", hash = "sha256:de9208bb427130c82a5dbfd104f92c8876fc9559278c880b3002755bbbe9c83d", size = 277136, upload-time = "2026-07-19T00:17:34.803Z
" },
    { url = "https://files.pythonhosted.org/packages/5d/3d/84165e4299ff76f3a40fe1f2abf939e976f693383a08d2beea6af62bd2c1/regex-2.6.7.19-cp313-cp313-ma
cosx_10_13_universal2.whl", hash = "sha256:f035d9dc1d25eff9d361456572231c7d27b5ccd473ca7dc0adfce732bd006d40", size = 496552, upload-time = "2026-07-19
T00:17:36.808Z" },
    { url = "https://files.pythonhosted.org/packages/02/a2/a65293e6e4cf28eb7ee1be5335a5386c40d6742e9f47fafc8fec785e16c7/regex-2.6.7.19-cp313-cp313-ma
cosx_10_13_x86_64.whl", hash = "sha256:c42572142ed0b9d5d261ba727157c426510da78e20828b66bbb855098b8a4e38", size = 296983, upload-time = "2026-07-19T00:
17:38.816Z" },
    { url = "https://files.pythonhosted.org/packages/95/47/2d0564e93d87bc48618360ddca232a2ca612bbdf53ce8465d45ca5ce14ee/regex-2.6.7.19-cp313-cp313-ma
cosx_11_0_arm64.whl", hash = "sha256:40b34dd88658e4fedd2fddbf0275ac970d00614b731357f425722a3ed1983d11", size = 291832, upload-time = "2026-07-19T00:17
:40.726Z" },
    { url = "https://files.pythonhosted.org/packages/07/cd/42dfbabff3dfc9603c501c0e2e2c5adbb09d127b267bf5348de0af338c15/regex-2.6.7.19-cp313-cp313-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:0c41c63992bf1874cebb6e7f56fd7d3c007924659a604ae3d90e427d40d4fd1
3", size = 796775, upload-time = "2026-07-19T00:17:42.382Z" },
    { url = "https://files.pythonhosted.org/packages/df/5d/f6a4839f2b934e3eed5973fd07f5929ee97d4c98939fb275ea23c274ee16/regex-2.6.7.19-cp313-cp313-ma
nylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:1d3372064506b94dd2c67c845f2db8062e9e9ba84d04e33cb96d7d33c11fe1a
e", size = 865687, upload-time = "2026-07-19T00:17:44.185Z" },
    { url = "https://files.pythonhosted.org/packages/14/b0/b47d6c36049bc59806a50bd4c86ced70bbe058d787f80281b1d7a9b0e024/regex-2.6.7.19-cp313-cp313-ma
nylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:fce7760bf283405b2c7999cab3da4e72f7deca6396013115e3f7a955db9760da", si
ze = 911962, upload-time = "2026-07-19T00:17:46.442Z" },
    { url = "https://files.pythonhosted.org/packages/2a/be/ff61f28f9273658cfe23acbbac5217221f6519960ed401e61dfdab12bc35/regex-2.6.7.19-cp313-cp313-ma
nylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:c0d702548d89d572b2929879bc883bb7a4c4709efafe4512cadee56c55c9bd15",
 size = 801817, upload-time = "2026-07-19T00:17:48.25Z" },
    { url = "https://files.pythonhosted.org/packages/c3/bb/8b4f7f26b333f9f79e1b453613c39bb4776f51d38ae66dd0ba31d6b354ca/regex-2.6.7.19-cp313-cp313-ma
nylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:d446c6ac40bb6e05025ccee55b84d80fe9bf8e93010ffc4bb9484f13d498835f", size = 776908, upl
oad-time = "2026-07-19T00:17:50.183Z" },
    { url = "https://files.pythonhosted.org/packages/09/13/610110fc5921d380516d03c26b652555f08aa0d23ea78a771231873c3638/regex-2.6.7.19-cp313-cp313-mu
sllinux_1_2_aarch64.whl", hash = "sha256:4c3501bfa814ab07b5580741f9bf78dfdfe146a04057f82df9e2402d2a975939", size = 784426, upload-time = "2026-07-19T0
0:17:52.454Z" },
    { url = "https://files.pythonhosted.org/packages/ca/f5/1ef9e2a83a5947c57ebff0b377cb5727c3d5ec1992317a320d035cd0dbb6/regex-2.6.7.19-cp313-cp313-mu
sllinux_1_2_ppc64le.whl", hash = "sha256:c4585c3e64b4f9e583b4d2683f18f5d5d872b3d71dcf24594b74ecc23602fa96", size = 860600, upload-time = "2026-07-19T0
0:17:54.229Z" },
    { url = "https://files.pythonhosted.org/packages/a0/02/073af33a3ec149241d11c80acea91e722aa0adbf05addd50f251c4fe89c3/regex-2.6.7.19-cp313-cp313-mu
sllinux_1_2_riscv64.whl", hash = "sha256:571fde9741eb0ccde23dd4e0c1d50fbae910e901fa7e629faf39b2dda740d220", size = 765950, upload-time = "2026-07-19T0
0:17:56.041Z" },
    { url = "https://files.pythonhosted.org/packages/81/a9/d1e9f819dc394a568ef370cd56cf25394e957a2235f8370f23b576e5a475/regex-2.6.7.19-cp313-cp313-mu
sllinux_1_2_s390x.whl", hash = "sha256:15b364b9b98d6d2fe1a85034c23a3180ff913f46caddc3895f6fd65186255ccc", size = 851794, upload-time = "2026-07-19T00:
17:57.897Z" },
    { url = "https://files.pythonhosted.org/packages/03/3a/8ae83eda7579feacdf984e71fb9e70635fb6f832eeddca58427ec4fca926/regex-2.6.7.19-cp313-cp313-mu
sllinux_1_2_x86_64.whl", hash = "sha256:ffd8893ccc1c2fce6e0d6ca402d716fe1b29db70c7132609a05955e31b2aa8f2", size = 789845, upload-time = "2026-07-19T00
:17:59.97Z" },
    { url = "https://files.pythonhosted.org/packages/4b/23/c195cbfe5a75fdec64d8f6554fd15237b837919d2c61bdc141d7c807b08b/regex-2.6.7.19-cp313-cp313-wi
n32.whl", hash = "sha256:f0fa4fa9c3632d708742baf2282f2055c11d888a790362670a403cbf48a2c404", size = 267135, upload-time = "2026-07-19T00:18:01.958Z" },
    { url = "https://files.pythonhosted.org/packages/b2/80/a11de8404b7272b70acb45c1c05987cce60b45d5693da2e176f0e390d564/regex-2.6.7.19-cp313-cp313-wi
n_amd64.whl", hash = "sha256:d51ffd3427640fa2da6ade574ceba932f210ad095f65fcc450a2b0a0d454868e", size = 277747, upload-time = "2026-07-19T00:18:04.121Z
" },
    { url = "https://files.pythonhosted.org/packages/d1/29/0f5c8eff1b4f1f3d83276d365fccecf666afcc7d947420943bf394d07adb/regex-2.6.7.19-cp313-cp313-wi
n_arm64.whl", hash = "sha256:c670fe7be5b6020b76bc6e8d2196074657e1327595bca93a389e1a76ab130ad8", size = 277129, upload-time = "2026-07-19T00:18:05.821Z
" },
    { url = "https://files.pythonhosted.org/packages/dc/4c/44b74742052cedda40f9ae469532a037112f7311a36669a891fba8984bb0/regex-2.6.7.19-cp313-cp313t-m
acosx_10_13_universal2.whl", hash = "sha256:db47b561c9afd884baa1f96f797c9ca369872c4b65912bc691cfa99e68340af2", size = 501134, upload-time = "2026-07-1
9T00:18:07.567Z" },
    { url = "https://files.pythonhosted.org/packages/f0/45/bbd038b5e39ee5613a5a689290145b40058cc152c41de9cc23639d2b9734/regex-2.6.7.19-cp313-cp313t-m
acosx_10_13_x86_64.whl", hash = "sha256:65dcd28d3eba2ab7c2fd906485cc301392b47cc2234790d27d4e4814e02cdfda", size = 299418, upload-time = "2026-07-19T00
:18:09.38Z" },
    { url = "https://files.pythonhosted.org/packages/65/38/c5bde94b4cedfd5850d64c3f08222d8e1600e84f6ee71d9b44b4b8163f74/regex-2.6.7.19-cp313-cp313t-m
acosx_11_0_arm64.whl", hash = "sha256:f2e7f8e2ab6c2922be02c7ec45185aa5bd771e2e57b95455ee343a44d8130dff", size = 294486, upload-time = "2026-07-19T00:1
8:11.188Z" },
    { url = "https://files.pythonhosted.org/packages/d7/6a/2f5e107cb26c960b781967178899daf2787a7ab151844ed3c01d6fc95474/regex-2.6.7.19-cp313-cp313t-m
anylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:fe31f28c94402043161876a258a9c6f757cb485905c7614ce8d6cd40e6b7bd
c1", size = 811643, upload-time = "2026-07-19T00:18:12.975Z" },
    { url = "https://files.pythonhosted.org/packages/37/d4/a2f963406d7d73a62eed84ba05a258afb6cad1b21aa4517443ce40506b78/regex-2.6.7.19-cp313-cp313t-m
anylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:f8f6fa298bb4f7f58a33334406218ba74716e68feddf5e4e54cd5d8082705a
bf", size = 871081, upload-time = "2026-07-19T00:18:14.733Z" },
    { url = "https://files.pythonhosted.org/packages/45/a3/44be546340bedb15f13063f5e7fe16793ea4d9ea2e805d09bd174ac27724/regex-2.6.7.19-cp313-cp313t-m
anylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:cc1b2440423a851fad781309dd87843868f4f66a6bcd1ddb9225cf4ec2c84732", s
ize = 917372, upload-time = "2026-07-19T00:18:16.724Z" },
    { url = "https://files.pythonhosted.org/packages/f8/f6/e0870b0fd2a40dba0074e4b76e514b21313d37946c9248453e34ec43923e/regex-2.6.7.19-cp313-cp313t-m
anylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:8ac59a0900474a52b7c04af8196affc22bd9842acb0950df12f7b813e983609a"
, size = 816089, upload-time = "2026-07-19T00:18:18.617Z" },
    { url = "https://files.pythonhosted.org/packages/ae/27/957e8e22690ad6634572b39b71f130a6105f4d0718bb16849eac00fff147/regex-2.6.7.19-cp313-cp313t-m
anylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:4896db1f4ce0576765b8272aa922df324e0f5b9bb2c3d03044ff32a7234a9aba", size = 785206, up
load-time = "2026-07-19T00:18:20.464Z" },
    { url = "https://files.pythonhosted.org/packages/76/a4/186e410941e731037c01166069ab86da9f65e8f8110c18009ccf4bd623ee/regex-2.6.7.19-cp313-cp313t-m
usllinux_1_2_aarch64.whl", hash = "sha256:4e6883a021db30511d9fb8cfb0f222ce1f2c369f7d4d8b0448f449a93ba0bdfc", size = 800431, upload-time = "2026-07-19T
00:18:22.716Z" },
    { url = "https://files.pythonhosted.org/packages/73/9f/e4e10e023d291d64a33e246610b724493bf1ce98e0e59c9b7c837e5acfb7/regex-2.6.7.19-cp313-cp313t-m
usllinux_1_2_ppc64le.whl", hash = "sha256:09523a592938aa9f587fb74467c63ff0cf88fc3df14c82ab0f0517dcf76aaa62", size = 864906, upload-time = "2026-07-19T
00:18:24.772Z" },
    { url = "https://files.pythonhosted.org/packages/24/57/ccb20b6be5f1f52a053d1ba2a8f7a077edb9d918248b8490d7506c6832b3/regex-2.6.7.19-cp313-cp313t-m
usllinux_1_2_riscv64.whl", hash = "sha256:1ebac3474b8589fce2f9b225b650afd61448f7c73a5d0255a10cc6366471aed1", size = 773559, upload-time = "2026-07-19T
00:18:27.008Z" },
    { url = "https://files.pythonhosted.org/packages/a3/82/f3b263cf8fad927dc102891da8502e718b7ff9d19af7a2a07c03865d7188/regex-2.6.7.19-cp313-cp313t-m
usllinux_1_2_s390x.whl", hash = "sha256:4a0530bb1b8c1c985e7e2122e2b4d3aedd8a3c21c6bfddae6767c4405668b56e", size = 857739, upload-time = "2026-07-19T00
:18:29.107Z" },
    { url = "https://files.pythonhosted.org/packages/47/2e/1687bd1b6c2aed5e672ccf845fc11557821fe7366d921b50889ea5ce57bf/regex-2.6.7.19-cp313-cp313t-m
usllinux_1_2_x86_64.whl", hash = "sha256:2ef7eeb108c47ce7bcc9513e51bcb1bf57e8f483d52fce68a8642e3527141ae0", size = 804522, upload-time = "2026-07-19T0
0:18:31.362Z" },
    { url = "https://files.pythonhosted.org/packages/76/7c/cc4e7655181b2d9235b704f2c5e19d8eff002bbc437bae59baee0e381aca/regex-2.6.7.19-cp313-cp313t-w
in32.whl", hash = "sha256:64b6ca7391a1395c2638dd5c7456d67bea44fc6c5e8e92c5dc8aa6a8f23292b4", size = 269141, upload-time = "2026-07-19T00:18:33.479Z" }
,
    { url = "https://files.pythonhosted.org/packages/bb/14/961b4c7b05a2391c32dbc85e27773076671ef8f97f36cec70fe414734c02/regex-2.6.7.19-cp313-cp313t-w
in_amd64.whl", hash = "sha256:f04b9f56b0e0614c0126be12c2c2d9f8850c1e57af302bd0a63bed379d4af974", size = 280036, upload-time = "2026-07-19T00:18:35.419
Z" },
    { url = "https://files.pythonhosted.org/packages/ce/67/795644550d788ddbb6dc458c95895f8009978ea6d6ea76b005eb3f45e8c9/regex-2.6.7.19-cp313-cp313t-w
in_arm64.whl", hash = "sha256:fcee38cd8e5089d6d4f048ba1233b3ad76e5954f545382180889112ff5cb712d", size = 279394, upload-time = "2026-07-19T00:18:37.454
Z" },
    { url = "https://files.pythonhosted.org/packages/d2/25/0c4c452f8ef3efe456745b2f33195f5904b573fb4c2ff3f0cb9ec188461e/regex-2.6.7.19-cp314-cp314-ma
cosx_10_13_universal2.whl", hash = "sha256:a81758ed242b861b72e778ba34d41366441a2e10b16b472784c88da2dea7e2dd", size = 496750, upload-time = "2026-07-19
T00:18:39.633Z" },
    { url = "https://files.pythonhosted.org/packages/24/9e/b70ca6c1704f6c7cd32a9e143c86cc5968d10981eca284bad670c245ea7d/regex-2.6.7.19-cp314-cp314-ma
cosx_10_13_x86_64.whl", hash = "sha256:4aa5435cdb3eb6f55fe98a171b05e3fbcd95fadaa4aa32acf62afd9b0cfdbcac", size = 297093, upload-time = "2026-07-19T00:

```
18:41.583Z" },
    { url = "https://files.pythonhosted.org/packages/87/74/0b692da2520d51fbff19c88b83d97e4c702909dd02386c585998b7e2dbed/regex-2026.7.19-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:60be8693a1dadc210bbcbc0db3e26da5f7d01d1d5a3da594e99b4fa42df404f5", size = 292043, upload-time = "2026-07-19T00:18:43.347Z" },
    { url = "https://files.pythonhosted.org/packages/e3/a7/1d478e614016045a33feae57446215f9fd65b665a5ceb2f891fb3183bc52/regex-2026.7.19-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:d19662dbedbe783d323196312d38f5ba53cf562963782521719 85da6899887d3", size = 797214, upload-time = "2026-07-19T00:18:45.362Z" },
    { url = "https://files.pythonhosted.org/packages/aa/ae/11b9c9411d92c30e3d2db32df5a31133e4a99a8fc397a604fd08f6c4bffb/regex-2026.7.19-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:d15df07081d91b76ff20d43f94592ee110330152d617b730fdbe5ef9fb680053", size = 866433, upload-time = "2026-07-19T00:18:47.315Z" },
    { url = "https://files.pythonhosted.org/packages/b1/62/2b2efc4992f91d6d204b24c647c9f9412e85379d92b7c0ab9fdae622327e/regex-2026.7.19-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:56ad4d9f77df871a99e25c37091052a02528ec0eb059de928ee33956b854b45b", size = 911360, upload-time = "2026-07-19T00:18:49.588Z" },
    { url = "https://files.pythonhosted.org/packages/14/71/986ceea9aa3da548bf1357cad89b63915ec6d21ec957c8113b29ece567df/regex-2026.7.19-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:7322ec6cc9fba9d49ab888bb82d67ac5625627aa168f0165139b17018df3fb8a", size = 801275, upload-time = "2026-07-19T00:18:51.767Z" },
    { url = "https://files.pythonhosted.org/packages/15/be/ce9d9534b2cda96eab32c548261224b9b4e220a4126f098f60f42ae7b4cd/regex-2026.7.19-cp314-cp314-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:9c7472192ebfad53a6be7c4a8bfb2d64b81c0e93a1fc8c57e1dd0b638297b5d1", size = 777131, upload-time = "2026-07-19T00:18:54.053Z" },
    { url = "https://files.pythonhosted.org/packages/61/2b/58b5c710f2c3929515a25f3a1ca0dad0dcd4518d4fff3cf23bc7adb8dcd2/regex-2026.7.19-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:c10b82c2634df08dfb13b1f04e38fe310d086ee092f4f69c0c8da234251e556e", size = 785020, upload-time = "2026-07-19T00:18:56.579Z" },
    { url = "https://files.pythonhosted.org/packages/84/03/5fe091935b74f15fe0f97998c215cae418d1c0413f6258c7d4d2e83aa37f/regex-2026.7.19-cp314-cp314-musllinux_1_2_ppc64le.whl", hash = "sha256:17ed5692f6acc4183e98331101a5f9e4f64d72fe58b753da4d444a2c77d05b12", size = 861263, upload-time = "2026-07-19T00:18:58.64Z" },
    { url = "https://files.pythonhosted.org/packages/d8/fa/d60bf82e10841eef62a9e32aac401468f05fddfbcb2942e342b1ba3d2433/regex-2026.7.19-cp314-cp314-musllinux_1_2_riscv64.whl", hash = "sha256:22a992de9a0d91bda927bf02b94351d737a0302905432c88a53de7c4b9ce62e2", size = 766199, upload-time = "2026-07-19T00:19:00.705Z" },
    { url = "https://files.pythonhosted.org/packages/bf/5d/11e64d151b0662b81d6bf644c74dc118d461df85bdf2577fadbbf751788a/regex-2026.7.19-cp314-cp314-musllinux_1_2_s390x.whl", hash = "sha256:618a0aed532be87294c4477b0481f3aa0f1520f4014a4374dd4cf789b4cd2c97", size = 851317, upload-time = "2026-07-19T00:19:03.015Z" },
    { url = "https://files.pythonhosted.org/packages/7c/34/532efb87488d90807bae6a443d357ee5e2728a478c597619c8aaa17cc0bd/regex-2026.7.19-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:2ce9e679f776649746729b6c86382da519ef649c8e34cc41df0d2e5e0f6c36d4", size = 789557, upload-time = "2026-07-19T00:19:05.338Z" },
    { url = "https://files.pythonhosted.org/packages/d6/90/3a8d5ca977171ec3ae21a71207d2228b2663bde14d7f7ef0e6363ecf9290/regex-2026.7.19-cp314-cp314-win32.whl", hash = "sha256:73f272fba87b8ccfe70a137d02a54af386f6d27aa509fbffdd978f5947aae1aa", size = 272531, upload-time = "2026-07-19T00:19:07.487Z" },
    { url = "https://files.pythonhosted.org/packages/96/e1/8862885e70409de70e8c005f57fb2e7be8d9ef0317250d60f4c9660a300d/regex-2026.7.19-cp314-cp314-win_amd64.whl", hash = "sha256:d721e53758b2cca74990185eb0671dd466d7a388a1a45d0c6f4c13cef41a68ac", size = 280831, upload-time = "2026-07-19T00:19:09.46Z" },
    { url = "https://files.pythonhosted.org/packages/08/82/2693e53e29f9104d9de95d37ce4dd826bd32d5f9c0085d3aa6ac042675c4/regex-2026.7.19-cp314-cp314-win_arm64.whl", hash = "sha256:65fa6cb38ed5e9c3637e68e544f598b39c3b86b808ed0627a67b68320384b459", size = 281099, upload-time = "2026-07-19T00:19:11.398Z" },
    { url = "https://files.pythonhosted.org/packages/92/b7/9a01aa16461a18cde9d7b9c3ab21e501db2ce33725f53014342b91df2b0a/regex-2026.7.19-cp314-cp314t-macosx_10_13_universal2.whl", hash = "sha256:5a2721c8720e2cb3c209925dfb9200199b4b07361c9e01d321719404b21458b3", size = 501121, upload-time = "2026-07-19T00:19:13.425Z" },
    { url = "https://files.pythonhosted.org/packages/f3/5e/bbaeca815dc9191c424c94a4fdc5c87c75748a64a6271821212ebdd4e1a3/regex-2026.7.19-cp314-cp314t-macosx_10_13_x86_64.whl", hash = "sha256:199535629f25caf89698039af3d1ad5fcae7f933e2112c73f1cdf49165c99518", size = 299415, upload-time = "2026-07-19T00:19:15.43Z" },
    { url = "https://files.pythonhosted.org/packages/cd/d6/0dd1a321afaab95eb7ff44aa0f637301786f1dc71c6b797b9ed236ed8890/regex-2026.7.19-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:9b60d7814174f059e5de4ab98271cc5ba9259cfea55273a81544dceea32dc8d9", size = 294483, upload-time = "2026-07-19T00:19:17.879Z" },
    { url = "https://files.pythonhosted.org/packages/92/5f/40bacf91d0904f812e13bbbab3864604c463eced8afdc54aeaa50492ea95/regex-2026.7.19-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:dbece16025afda5e3031af0c4059207e61dcf73ef13af844964f57f387d1c435", size = 811833, upload-time = "2026-07-19T00:19:20.102Z" },
    { url = "https://files.pythonhosted.org/packages/94/7c/4902744261f775aeede8b5627314b38482da29cf49a57b66a6fb753246c5/regex-2026.7.19-cp314-cp314t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:d24ecb4f5e009ea0bd275ee37ad9953b32005e2e5e60f8bbae16da0dbbf0d3a0", size = 871270, upload-time = "2026-07-19T00:19:22.365Z" },
    { url = "https://files.pythonhosted.org/packages/16/70/6980c9be6bf21c0a60ed3e0aea39cf419ecf3b08d1d9947bc56e196ef186/regex-2026.7.19-cp314-cp314t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:8cae6fd77a5b72dae505084b1a2ee0360139faf72fedbab667cd7cc65aae7a6a", size = 917534, upload-time = "2026-07-19T00:19:24.529Z" },
    { url = "https://files.pythonhosted.org/packages/52/92/8b2bd872782ce8c42691e39acb38eb8efe014e5ddb78ad7d943d6f197ce9/regex-2026.7.19-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9724e6cb5e478cd7d8cabf027826178739cb18cf0e117d0e32814d479fa02276", size = 816135, upload-time = "2026-07-19T00:19:26.919Z" },
    { url = "https://files.pythonhosted.org/packages/de/2d/33a602f657bdc4041f17d79f92ab18261d255d91a06117a6e29df023e5e2/regex-2026.7.19-cp314-cp314t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:572fc57b0009c735ee56c175ea021b637a15551a312f56734277f923d6fd0f6c", size = 785492, upload-time = "2026-07-19T00:19:29.192Z" },
    { url = "https://files.pythonhosted.org/packages/9e/36/0987cf4cb271680064a70d24a475873775a151d0b7058698a006cb0cae4a/regex-2026.7.19-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:20568e182eb82d39a6bf7cff3fd58566f14c75c6f74b2c8c96537eecf9010e3a", size = 800658, upload-time = "2026-07-19T00:19:31.392Z" },
    { url = "https://files.pythonhosted.org/packages/a8/24/c14f31c135e1ba55fa4f9a58ca98d0842512bf6188230763c31c8f449e3b/regex-2026.7.19-cp314-cp314t-musllinux_1_2_ppc64le.whl", hash = "sha256:1d58561843f0ff7dc78b4c28b5e2dc388f3eff94ebc8a232a3adba961fc00009", size = 865073, upload-time = "2026-07-19T00:19:33.485Z" },
    { url = "https://files.pythonhosted.org/packages/14/85/181a12211f22469f24d2de1ebddfe397d2396e2c29013b9a58134a91069a/regex-2026.7.19-cp314-cp314t-musllinux_1_2_riscv64.whl", hash = "sha256:61bb1bd45520aacd56dd80943bd34991fb5350afdd1f36f2282230fd5154a218", size = 773684, upload-time = "2026-07-19T00:19:35.599Z" },
    { url = "https://files.pythonhosted.org/packages/23/58/bd1a0c1a62251366f8d21f41b1ea3c76994962071b8b6ea42f72d505c0f0/regex-2026.7.19-cp314-cp314t-musllinux_1_2_s390x.whl", hash = "sha256:cd3584591ea4429026cdb931b054342c2bcf189b44ff367f8d5c15bc092a2966", size = 857769, upload-time = "2026-07-19T00:19:37.738Z" },
    { url = "https://files.pythonhosted.org/packages/e4/4f/f7e2dad6756b2fe1fe75dd90a628c3b45f249d39f948dd90cd2476325417/regex-2026.7.19-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:5cc26a66e212fa5d6c6170c3a40d99d888db3020c6fdab1523250d4341382e44", size = 804546, upload-time = "2026-07-19T00:19:40.229Z" },
    { url = "https://files.pythonhosted.org/packages/2b/d7/01d31d5bdb09bc026fab77f59a371fdf8f9b292e4810546c56182ca70498/regex-2026.7.19-cp314-cp314t-win32.whl", hash = "sha256:2c4e61e2e1be56f63ec3cc618aa9e0de81ef6f43d177205451840022e24f5b78", size = 274526, upload-time = "2026-07-19T00:19:42.398Z" },
    { url = "https://files.pythonhosted.org/packages/52/0e/cea4ce73bc0a8247a0748228ae6669984c7e1f8134b6fa66e59c0572e0ea/regex-2026.7.19-cp314-cp314t-win_amd64.whl", hash = "sha256:c639ea314df70a7b2811e8020448c75af8c9445f5a60f8a4ced81c306a9380c2", size = 283763, upload-time = "2026-07-19T00:19:44.644Z" },
    { url = "https://files.pythonhosted.org/packages/6f/b6/26e41975febae63b7a6e3e02f32cff6cff2e4f10d19c929082f56aebf7c6/regex-2026.7.19-cp314-cp314t-win_arm64.whl", hash = "sha256:9a15e785f244f3e07847b984ce8773fc3da10a9f3c131cc49a4c5b4d672b4547", size = 283451, upload-time = "2026-07-19T00:19:46.639Z" },
]

[[package]]
name = "requests"
version = "2.34.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "certifi" },
    { name = "charset-normalizer" },
    { name = "idna" },
    { name = "urllib3" },
]
sdist = { url = "https://files.pythonhosted.org/packages/ac/c3/e2a2b89f2d3e2179abd6d00ebd70bff6273f37fb3e0cc209f48b39d00cbf/requests-2.34.2.tar.gz", hash = "sha256:f288924cae4e29463698d6d60bc6a4da69c89185ad1e0bcc4104f584e960b9ed", size = 142856, upload-time = "2026-05-14T19:25:27.735Z" }
wheels = [
```

```
    { url = "https://files.pythonhosted.org/packages/a0/f4/c67b0b3f1b9245e8d266f0f112c500d50e5b4e83cb6f3b71b6528104182a/requests-2.34.2-py3-none-any.whl", hash = "sha256:2a0d60c172f83ac6ab31e4554906c0f3b3588d37b5cb939b1c061f4907e278e0", size = 73075, upload-time = "2026-05-14T19:25:26.443Z" },
]

[[package]]
name = "requests-toolbelt"
version = "1.0.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "requests" },
]
sdist = { url = "https://files.pythonhosted.org/packages/f3/61/d7545dafb7ac2230c70d38d31cbfe4cc64f7144dc41f6e4e4b78ecd9f5bb/requests-toolbelt-1.0.0.tar.gz", hash = "sha256:7681a0a3d047012b5bdc0ee37d7f8f07ebe76ab08caeccfc3921ce23c88d5bc6", size = 206888, upload-time = "2023-05-01T04:11:33.229Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/3f/51/d4db610ef29373b879047326cbf6fa98b6c1969d6f6dc423279de2b1be2c/requests_toolbelt-1.0.0-py2.py3-none-any.whl", hash = "sha256:cccfdd665f0a24fcf4726e690f65639d272bb0637b9b92dfd91a5568ccf6bd06", size = 54481, upload-time = "2023-05-01T04:11:28.427Z" },
]

[[package]]
name = "rpds-py"
version = "2026.6.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/aa/2a/9618a122aeb2a169a28b03889a2995fe297588964333d4a7d67bdf46e147/rpds_py-2026.6.3.tar.gz",
hash = "sha256:1cebd1337c242e4ec2293e541f712b2da849b29f48f0c293684b71c0632625d4", size = 64051, upload-time = "2026-06-30T07:17:53.009Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/5c/be/2e8974163072e7bab7df1a5acd54c4498e75e35d6d18b864d3a9d5dadc92/rpds_py-2026.6.3-cp312-cp312-macosx_10_12_x86_64.whl", hash = "sha256:a0811d33247c3d6128a3001d763f2aa056bb3425204335400ac54f89eec3a0d0", size = 343691, upload-time = "2026-06-30T07:15:14.96Z" },
    { url = "https://files.pythonhosted.org/packages/a4/73/319dfa745dd668efe89309141ded489126461fcecd2b8f3a3cda185129b6/rpds_py-2026.6.3-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:538949e262e46caa31ac01bdb3c1e8f642622922cacbabbae6a8445d9dc33eaf", size = 338542, upload-time = "2026-06-30T07:15:16.267Z" },
    { url = "https://files.pythonhosted.org/packages/21/63/4239893be1c4d09b709b1a8f6be4188f0870084ff547f46606b8a75f1b03/rpds_py-2026.6.3-cp312-cp312-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:55927d532399c2c646100ff7feb48eaa940ad70f42cd68e1328f3ded9f81ca24", size = 368180, upload-time = "2026-06-30T07:15:17.62Z" },
    { url = "https://files.pythonhosted.org/packages/1c/ca/9c5de382225234ceb37b1844ebdb140db12b2a278bb9efe2fcd19f6c82ce/rpds_py-2026.6.3-cp312-cp312-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:f56f1695bc5c0871cbc33dc0130fcf503aab0c57dcc5a6700a4f49eba4f2652e", size = 375067, uploaded-time = "2026-06-30T07:15:18.952Z" },
    { url = "https://files.pythonhosted.org/packages/87/dc/863f69d1bf04ade34b7fe0d59b9fdf6f0135fe2d7cbca74f1d665589559d/rpds_py-2026.6.3-cp312-cp312-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:270b293dae9058fc9fcedab50f13cebf46fb8ed1d1d54e0521a9da5d6b211975", size = 490509, upload-time = "2026-06-30T07:15:20.434Z" },
    { url = "https://files.pythonhosted.org/packages/ce/ef/eac16a12048b45ec7c7fa94f2be3438a5f26bf9cc8580b18a1cfd609b7f6/rpds_py-2026.6.3-cp312-cp312-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:127565fead0a10943b282957bd5447804ff3160ad79f2ad2635e6d249e380680", size = 382754, upload-time = "2026-06-30T07:15:21.831Z" },
    { url = "https://files.pythonhosted.org/packages/04/8f/d2f3f532616be4d06c316ef119683e832bd3d41e112bf3a88f4151c95b17/rpds_py-2026.6.3-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:ecabd69db66de867690f9797f2f8fa27ba501bbc24540cbdbdc649cd15888ba6", size = 366189, uploaded-time = "2026-06-30T07:15:23.371Z" },
    { url = "https://files.pythonhosted.org/packages/e3/29/41a7b0e98a4b44cd676ab7598419623373eb43b20be68c084935c1a8cf88/rpds_py-2026.6.3-cp312-cp312-manylinux_2_31_riscv64.whl", hash = "sha256:58eadac9cd119677b60e1cf8ac4052f35949d71b8a9e5556efccbe82533cf22a", size = 377750, upload-time = "2026-06-30T07:15:24.659Z" },
    { url = "https://files.pythonhosted.org/packages/2e/05/ecda0bec46f9a1565090bcdc941d023f6a25aff85fda28f89f8d19878152/rpds_py-2026.6.3-cp312-cp312-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:7491ee23305ac3eb59e492b6945881f5cd77a6f731061a3f25b77fd40f9e99a4", size = 395576, upload-time = "2026-06-30T07:15:25.987Z" },
    { url = "https://files.pythonhosted.org/packages/68/a8/6ed52f03ee6cb854ce78785cc9a9a672eb880e83fd7224d471f667d151f1/rpds_py-2026.6.3-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:2c99f7e8ccb3dd6e3e4bfeac657a7b208c9bac8075f4b078c02d7404c34107fa", size = 543807, upload-time = "2026-06-30T07:15:27.356Z" },
    { url = "https://files.pythonhosted.org/packages/8f/d6/156c0d3eea27ba09b92562ba2364ba124c0a061b199e17eac637cd25a5e2/rpds_py-2026.6.3-cp312-cp312-musllinux_1_2_i686.whl", hash = "sha256:62698275682bf121181861295c9181e789030a2d516071f5b8f3c23c170cd0fc", size = 611187, upload-time = "2026-06-30T07:15:28.931Z" },
    { url = "https://files.pythonhosted.org/packages/f1/31/774212ed989c62f7f310220089f9b0a3fb8f40f5443d1727abd5d9f52bc9/rpds_py-2026.6.3-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:a214c993455f99a89aaeadc9b21241900037adc9d97203e374d75513c5911822", size = 573030, upload-time = "2026-06-30T07:15:30.553Z" },
    { url = "https://files.pythonhosted.org/packages/c9/50/22f73127a41f1ce4f87fe39aadfb9a126345801c274aa93ae88456249327/rpds_py-2026.6.3-cp312-cp312-win32.whl", hash = "sha256:501f9f04a588d6a09179368c57071301445191767c64e4b52a6aa9871f1ef5ed", size = 202185, upload-time = "2026-06-30T07:15:32.027Z" },
    { url = "https://files.pythonhosted.org/packages/04/3a/f0ee4d4dde9d3b69dedf1b5f74e7a40017046d55052d173e418c6a94f960/rpds_py-2026.6.3-cp312-cp312-win_amd64.whl", hash = "sha256:2c958bf94822e9290a40aaf2a822d4bc5c88099093e3948ad6c571eca9272e5f", size = 220394, upload-time = "2026-06-30T07:15:33.359Z" },
    { url = "https://files.pythonhosted.org/packages/f3/83/3382fe37f809b59f02aac04dbc4e765b480b46ee0227ed516e3bdc4d3dfc/rpds_py-2026.6.3-cp312-cp312-win_arm64.whl", hash = "sha256:22bffe6042b9bcb0822bcd1955ec00e245daf17b4344e4ed8e9551b976b63e96", size = 215753, upload-time = "2026-06-30T07:15:34.778Z" },
    { url = "https://files.pythonhosted.org/packages/a4/9e/b818ee580026ec578138e961027a68820c40afeb1ec8f6819b54fb99e196/rpds_py-2026.6.3-cp313-cp313-macosx_10_12_x86_64.whl", hash = "sha256:3cfe765c1da0072636ca06628261e0ea05688e160d5c8a03e0217c3854037223", size = 343012, upload-time = "2026-06-30T07:15:36.005Z" },
    { url = "https://files.pythonhosted.org/packages/f3/6b/686d9dc4359a8f163cfbbf89ee0b4e586431de22fe8248edb63a8cf50d49/rpds_py-2026.6.3-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:f4d78253f6996be4901669ad25319f842f740eccf4d58e3c7f3dd39e6dde1d8f", size = 338203, upload-time = "2026-06-30T07:15:37.462Z" },
    { url = "https://files.pythonhosted.org/packages/9e/9b/069aa329940f8207615e091f5eedbbd40e1e15eac68a0790fd05ccdf796c/rpds_py-2026.6.3-cp313-cp313-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:54f45a148e28767bf343d33a684693c70e451c6f4c0e9904709a723fafbdfc1f", size = 367984, upload-time = "2026-06-30T07:15:39.008Z" },
    { url = "https://files.pythonhosted.org/packages/14/db/34c203e4becff3703e4d3bc121842c00b8689197f398161203a880052f4e/rpds_py-2026.6.3-cp313-cp313-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:842e7b070435622248c7a2c44ae53fa1440e073cc3023bc919fed570884097a7", size = 374815, uploaded-time = "2026-06-30T07:15:40.253Z" },
    { url = "https://files.pythonhosted.org/packages/ee/7d/8071067d2cc453d916ad836e828c943f575e8a44612537759002a1e07381/rpds_py-2026.6.3-cp313-cp313-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:8020133a74bd81b4572dd8e4be028a6b1ebcd70e6726edc3918008c08bee6ee6", size = 490545, upload-time = "2026-06-30T07:15:41.729Z" },
    { url = "https://files.pythonhosted.org/packages/a3/42/da06c5aa8f0484ff07f270787434204d9f4535e2f8c3b51ed402267e63c3/rpds_py-2026.6.3-cp313-cp313-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:cdc7e35386f3847df728fbcb5e887e2d79c19e2fa1eba9e51b6621d23e3243af", size = 382828, upload-time = "2026-06-30T07:15:43.327Z" },
    { url = "https://files.pythonhosted.org/packages/57/d7/fe978efc2ae50abe48eb7464668ea99f53c010c60aeebb7b35ad27f23661/rpds_py-2026.6.3-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:acac386b453c2516111b50985d60ce46e7fadb5ea71ae7b25f4c946935bf27cf", size = 365678, uploaded-time = "2026-06-30T07:15:44.992Z" },
    { url = "https://files.pythonhosted.org/packages/69/9d/1d8922e1990b2a6eb532b6ff53d3e73d2b3bbffc84116c75826bee73dfc6/rpds_py-2026.6.3-cp313-cp313-manylinux_2_31_riscv64.whl", hash = "sha256:425560c6fa0415f27261727bb20bd097568485e5eb0c121f1949417d1c516885", size = 377811, upload-time = "2026-06-30T07:15:46.523Z" },
    { url = "https://files.pythonhosted.org/packages/b1/3d/198dceafb4fb034a6a47347e1b0735d34e0bd4a50be4e898d408ee66cb14/rpds_py-2026.6.3-cp313-cp313-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:a550fb4950a06dde3beb4721f5ad4b25bf4513784665b0a8522c792e2bd822a4", size = 395382, upload-time = "2026-06-30T07:15:47.955Z" },
    { url = "https://files.pythonhosted.org/packages/1f/f1/13968e49655d40b6b19d8b9140296bbc6f1d86b3f0f6c346cf9f1adddf4b/rpds_py-2026.6.3-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:4f4bca01b63096f606e095734dd56e74e175f94cfbf24ff3d63281cec61f7bb7", size = 543832, upload-time = "2026-06-30T07:15:49.33Z" },
    { url = "https://files.pythonhosted.org/packages/ac/ab/289bcb1b90bd3e40a2900c561fa0e2087345ecbb094f0b870f2345142b7c/rpds_py-2026.6.3-cp313-cp313-musllinux_1_2_i686.whl", hash = "sha256:ccffae9a092a00deb7efd545fe5e2c33c33b88e7c054337e9a74c179347d0b7d", size = 611011, upload-time = "2026-06-30T07:15:50.847Z" },
    { url = "https://files.pythonhosted.org/packages/1e/16/5043105e679436ccfbc8e5e0dd2d663ed18a8b8113515fd06a5e5d77c83e/rpds_py-2026.6.3-cp313-cp313-m
```

usllinux_1_2_x86_64.whl", hash = "sha256:1cf01971c4f2c5553b772a542e4aaf191789cd331bc2cd4ff0e6e65ba49e1e97", size = 572431, upload-time = "2026-06-30T07:15:52.394Z" },
{ url = "https://files.pythonhosted.org/packages/85/ed/adab103321c0a6565d5ae1c2998349bc3ee175b82ccc5ae8fc04cc413075/rpds_py-2026.6.3-cp313-cp313-win32.whl", hash = "sha256:8c3d1e9c15b9d51ca0391e13da1a25a0a4df3c58a37c9dc368e0736cf7f69df0", size = 201710, upload-time = "2026-06-30T07:15:53.894Z" },
{ url = "https://files.pythonhosted.org/packages/7b/ed/a03b09668e74e5dabbf2e211f6468e1820c0552f7b0500082da31841bf7b/rpds_py-2026.6.3-cp313-cp313-win_amd64.whl", hash = "sha256:9250a9a0a6fd4648b3f868da8d91a4c52b5811a62df58e753d50ae4454a36f80", size = 219454, upload-time = "2026-06-30T07:15:55.25Z" },
{ url = "https://files.pythonhosted.org/packages/27/17/b8642c12930b71bc2b25831f6708ccf0f75abcd11883932ec9ce54ba3a78/rpds_py-2026.6.3-cp313-cp313-win_arm64.whl", hash = "sha256:900a67df3fd1660b035a4761c4ce73c382ea6b35f90f9863c36c6fd8bf8b09bb", size = 215063, upload-time = "2026-06-30T07:15:56.573Z" },
{ url = "https://files.pythonhosted.org/packages/b6/36/7fbe9dcdaf857fb3f63c2a2284b62492d95f5e8334e947e5fb6e7f68c9be/rpds_py-2026.6.3-cp314-cp314-macosx_10_12_x86_64.whl", hash = "sha256:931908d9fc855d8f74783377822be318edb6dcb19e47169dc038f9a1bf60b06e", size = 344510, upload-time = "2026-06-30T07:15:57.921Z" },
{ url = "https://files.pythonhosted.org/packages/ba/54/f785cc3d3f60839ca57a5af4927a9f347b07b2799c373fc20f7949f87c7e/rpds_py-2026.6.3-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:d7469697dce35be237db177d42e2a2ee26e6dcc5fc052078a6fefabd288c6edd", size = 339495, upload-time = "2026-06-30T07:15:59.238Z" },
{ url = "https://files.pythonhosted.org/packages/63/ef/d4cdaf309e6b095b43597103cf8c0b951d6cca2acce68c474f75ec12e0c7/rpds_py-2026.6.3-cp314-cp314-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:bcfbcf66006befb9fd2aeaa9e01feaf881b4dc330a02ba07d2322b1c11be7b5d", size = 369454, upload-time = "2026-06-30T07:16:01.021Z" },
{ url = "https://files.pythonhosted.org/packages/96/4a/9559a68b7ee15db09d7981212e8c2e219d2a1d6d4faa0391d813c3496a36/rpds_py-2026.6.3-cp314-cp314-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:847927daf4cffbd4e90e42bc890069897101edd015f956cb8721b3473372edda", size = 374583, uploaded-time = "2026-06-30T07:16:02.287Z" },
{ url = "https://files.pythonhosted.org/packages/ef/75/8964aa7d2c6e8ac43eba8eb6e6b0fdda1f46d39f2fc3e6aa9f2cb17f485d/rpds_py-2026.6.3-cp314-cp314-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:aca6c1ef08a82bfe327cc156da694660f599923e2e6665b6d81c9c2d0ac9ffc8", size = 492919, upload-time = "2026-06-30T07:16:03.723Z" },
{ url = "https://files.pythonhosted.org/packages/8f/97/6908094ac804115e65aedfd90f1b5fee4eebebd3f6c4cfc5419939267565/rpds_py-2026.6.3-cp314-cp314-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:ae50181a047c871561212bb97f7932a2d45fb53e947bd9b57ebad85b529cbc53", size = 383725, upload-time = "2026-06-30T07:16:05.305Z" },
{ url = "https://files.pythonhosted.org/packages/d1/9c/0d1fdc2e7aba23e290d603bc494e97bd205bae262ce33c6b32a69768ed5e/rpds_py-2026.6.3-cp314-cp314-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:dc319e5a1de4b6913aac94bf6a2f9e847371e0a140a43dd4991db1a09bc2d504", size = 367255, uploaded-time = "2026-06-30T07:16:07.086Z" },
{ url = "https://files.pythonhosted.org/packages/c4/fe/f0209ca4a9ed074bc8acb44dfd0e81c3122e94c9689f5645b7973a866719/rpds_py-2026.6.3-cp314-cp314-manylinux_2_31_riscv64.whl", hash = "sha256:e4316bf32babbed84e691e352faf967ce2f0f024174a8643c37c94a1080374fc", size = 379060, upload-time = "2026-06-30T07:16:08.525Z" },
{ url = "https://files.pythonhosted.org/packages/c6/8d/f1cc54c616b9d8897de8738aac148d20afca93f68187475fe194d09a71b9/rpds_py-2026.6.3-cp314-cp314-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:8c6e5a2f750cc71c3e3b11d71661f21d6f9bc6cebc6564b1466417a1ec03ec77", size = 395960, upload-time = "2026-06-30T07:16:09.989Z" },
{ url = "https://files.pythonhosted.org/packages/fb/04/aafff00f73aeca2945f734f1d483c64ab8f472d0864ab02377fd8e89c3b2/rpds_py-2026.6.3-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:4470ce197d4090875cf6affbf1f853338387428df97c4fb7b7106317b8214698", size = 545356, upload-time = "2026-06-30T07:16:11.816Z" },
{ url = "https://files.pythonhosted.org/packages/fd/cc/e229663b9e4ddac5a4acbe9085dd80a71af2a5d356b8b39d6bff233f24b0/rpds_py-2026.6.3-cp314-cp314-musllinux_1_2_i686.whl", hash = "sha256:ea964164cc9afa72d4d9b23cc28dafae93693c0a53e0b42acbff15b22c3f9ddd", size = 612319, upload-time = "2026-06-30T07:16:13.586Z" },
{ url = "https://files.pythonhosted.org/packages/e3/7a/8a0e6d3e6cd066af108b71b43122c3fe158dd9eb86acac626593a2582eb1/rpds_py-2026.6.3-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:639c8929aa0afe81be836b04de888460d6bed38b9c54cfc18da8f6bfabf5af5d", size = 573508, upload-time = "2026-06-30T07:16:15.23Z" },
{ url = "https://files.pythonhosted.org/packages/87/03/2a69ab618a789cf6cf85c86bb844c62d090e700ab1a2aa676b3741b6c516/rpds_py-2026.6.3-cp314-cp314-win32.whl", hash = "sha256:882076c00c0a608b131187055ddc5ae29f2e7eaf870d6168980420d58528a5c8", size = 202504, upload-time = "2026-06-30T07:16:16.893Z" },
{ url = "https://files.pythonhosted.org/packages/85/62/a3892ba945f4e24c78f352e5de3c7620d8479f73f211406a97263d13c7d2/rpds_py-2026.6.3-cp314-cp314-win_amd64.whl", hash = "sha256:0be972be84cfcaf46c8c6edf690ca0f154ac17babf1f6a955a51579b34ad2dc5", size = 220380, upload-time = "2026-06-30T07:16:18.108Z" },
{ url = "https://files.pythonhosted.org/packages/3d/e7/c2bd44dc831931815ad11ebb5f430b5a0a4d3caa9de837107876c30c3432/rpds_py-2026.6.3-cp314-cp314-win_arm64.whl", hash = "sha256:2a9c6f195058cb45335e8cc3802745c603d716eb96bc9625950c1aac71c0c703", size = 215976, upload-time = "2026-06-30T07:16:19.654Z" },
{ url = "https://files.pythonhosted.org/packages/79/9c/fff7b74bce9a091ec9a012a03f9ff5f69364eaf9451060dfc4486da2ffdd/rpds_py-2026.6.3-cp314-cp314t-macosx_10_12_x86_64.whl", hash = "sha256:f90938e92afda60266da758ee7d363447f7f0138c9559f9e1811629580582d90", size = 346840, upload-time = "2026-06-30T07:16:21.268Z" },
{ url = "https://files.pythonhosted.org/packages/e9/44/77bcb1168b33704908295533d27f10eb811e9e3e193e8993dc99572211d3/rpds_py-2026.6.3-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:ec829541c45bca16e61c7ae50c20501f213605beb75d1aba91a6ee37fbbb56a4", size = 340282, upload-time = "2026-06-30T07:16:22.875Z" },
{ url = "https://files.pythonhosted.org/packages/87/3c/7a9081c7c9e645b39efe19e4ffbeccd80add246327cd9b888aecffd72317/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:afd70d95892096cdb26f15a00c45907b17817577aa8d1c76b2dcc2788391f9e9", size = 370403, upload-time = "2026-06-30T07:16:24.415Z" },
{ url = "https://files.pythonhosted.org/packages/f7/69/af47021eb7dad6ff3396cb001c08f0f3c4d06c20253f75be6421a59fe6b7/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:29dfa0533a5d4c94d4dfa1b694fcb56c9c63aad8330ffdd816fd225d0a7a162f", size = 376055, uploaded-time = "2026-06-30T07:16:26.111Z" },
{ url = "https://files.pythonhosted.org/packages/81/fc/a3bcf517084396a6dd258c592567a3c011ba4557f2fde23dceaf26e74f2e/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:af05d726809bff6b141be124d4c7ce998f9c9c7f30edb1f46c07aa103d540b41", size = 494419, upload-time = "2026-06-30T07:16:27.596Z" },
{ url = "https://files.pythonhosted.org/packages/c9/eb/13d529d1788135425c7bf207f8463458ca5d92e43f3f701365b83e9dffc1/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:9826217f048f620d9a712672818bf231442c1b35d96b227a07eabd11b4bb6945", size = 384848, upload-time = "2026-06-30T07:16:29.183Z" },
{ url = "https://files.pythonhosted.org/packages/8e/f4/b7ac49f30013aba8f7b9566b1dd07e81de95e708c1374b7bacc5b9bc5c9c/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:536bceea4fa4acf7e1c61da2b5786304367c816c8895be71b8f537c480b0ea1f", size = 371369, upload-time = "2026-06-30T07:16:30.912Z" },
{ url = "https://files.pythonhosted.org/packages/31/86/6260bafa622f788b07ddec0e52d810305c8b9b0b8c27f58a2ab04bf62b4f/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_31_riscv64.whl", hash = "sha256:bc0011654b91cc4fb2ae701bec0a0ba1e552c0714247fa7af6c59e0ccfa3a4e1", size = 379673, upload-time = "2026-06-30T07:16:32.486Z" },
{ url = "https://files.pythonhosted.org/packages/19/c3/03f1ee79a047b48daeca157c89a18509cde22b6b951d642b9b0af1be660a/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:539d75de9e0d536c84ff18dfeb805398e58227001ce09231a26a08b9aed1ee0e", size = 397500, upload-time = "2026-06-30T07:16:34.471Z" },
{ url = "https://files.pythonhosted.org/packages/f0/95/8ed0cd8c377dca12aea498f119fe639fc474d1461545c39d2b5872eb1c0f/rpds_py-2026.6.3-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:166cf54d9f44fc6ceb53c7860258dde44a81406646de79f8ed3234fca3b6e538", size = 545978, upload-time = "2026-06-30T07:16:36.45Z" },
{ url = "https://files.pythonhosted.org/packages/d3/f2/0eb57f0eaa83f8fc152a7e03de968ab77e1f00732bebc892b190c6eebde7/rpds_py-2026.6.3-cp314-cp314t-musllinux_1_2_i686.whl", hash = "sha256:d34c20167764fbcf927194d532dd7e0c56772f0a5f943fa5ef9e9afbba8fb9db", size = 613350, upload-time = "2026-06-30T07:16:38.213Z" },
{ url = "https://files.pythonhosted.org/packages/5b/de/e0674bdbc3ef7634989b3f854c3f34bc1f587d36e5bfdc5c378d57034619/rpds_py-2026.6.3-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:ea7bb13b7c9a29791f87a0387ba7d3ad3a6d783d827e4d3f27b40a0ff44495e2", size = 576486, upload-time = "2026-06-30T07:16:39.797Z" },
{ url = "https://files.pythonhosted.org/packages/f2/f6/21101359743cd136ada781e8210a85769578422ba460672eea0e29739200/rpds_py-2026.6.3-cp314-cp314t-win32.whl", hash = "sha256:6de4744d05bd1aa1be4ed7ea1189e3979196808008113bbbf899a460966b925e", size = 201068, upload-time = "2026-06-30T07:16:41.316Z" },
{ url = "https://files.pythonhosted.org/packages/a6/b2/9574d4d44f7760c2aa32d92a0a4f41698e33f5b204a0bf5c9758f52c79d5/rpds_py-2026.6.3-cp314-cp314t-win_amd64.whl", hash = "sha256:c7b9a2f8f4d8e90af72571d3d495deebdd7e3c75451f5b41719aee166e940fc2", size = 220600, upload-time = "2026-06-30T07:16:43.091Z" },
{ url = "https://files.pythonhosted.org/packages/08/ae/f23a2697e6ee6340a578b0f136be6483657bef0c6f9497b752bb5c0964bb/rpds_py-2026.6.3-cp315-cp315-macosx_10_12_x86_64.whl", hash = "sha256:e059c5dde6452b44424bd1834557556c226b57781dee1227af23518459722b13", size = 344726, upload-time = "2026-06-30T07:16:44.5Z" },
{ url = "https://files.pythonhosted.org/packages/c3/63/e7b3a1a5358dd32c930a1062d8e15b67fd6e8922e81df9e91706d66ee5c8/rpds_py-2026.6.3-cp315-cp315-macosx_11_0_arm64.whl", hash = "sha256:2f7c26fbc5acd2522b95d4177fe4710ffd8e9b20529e703ffbf8db4d93903f05", size = 339587, upload-time = "2026-06-30T07:16:46.255Z" },
{ url = "https://files.pythonhosted.org/packages/ec/64/10a85681916ca55fffb91b0a211f84e34297c109243484dd6394660a8a7c/rpds_py-2026.6.3-cp315-cp315-m

```
anylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:a3086b538543802f84c843911242db20447de00d8752dd0efc936dbcf02218ba", size = 369585, upl
oad-time = "2026-06-30T07:16:48.101Z" },
    { url = "https://files.pythonhosted.org/packages/76/c2/baf95c7c38823e12ba34407c5f5767a89e5cf2233895e56f608167ae9493/rpds_py-2026.6.3-cp315-cp315-m
anylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:8f2e5c5ee828d42cb11760761c0af6507927bec42d0ad5458f97c9203b054617", size = 375479, uploa
d-time = "2026-06-30T07:16:49.93Z" },
    { url = "https://files.pythonhosted.org/packages/6a/94/0aad06c72d65101e11d33528d438cda99a39ce0da99466e156158f2541d3/rpds_py-2026.6.3-cp315-cp315-m
anylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:ed0c1e5d10cdc7135537988c74a0188da68e2f3c30813ba3744ab1e42e0480f9", size = 492418, upl
oad-time = "2026-06-30T07:16:51.641Z" },
    { url = "https://files.pythonhosted.org/packages/b5/17/de3f5a479a1f056535d7489819639d8cd591ea6281d700390b43b1abd745/rpds_py-2026.6.3-cp315-cp315-m
anylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:8c2642a7603ec0b16ed77da4555db3b4b472341904873788327c0b0d7b95f1bb", size = 384123, upload-
time = "2026-06-30T07:16:53.622Z" },
    { url = "https://files.pythonhosted.org/packages/46/7d/bf09bd1b145bb2671c03e1e6d1ab8651858d90d8c7dfeadd85a37a934fd8/rpds_py-2026.6.3-cp315-cp315-m
anylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:8e4320744c1ffdd95a603def63344bfab2d33edeab301c5007e7de9f9f5b3885", size = 367351, uploa
d-time = "2026-06-30T07:16:55.241Z" },
    { url = "https://files.pythonhosted.org/packages/a3/ea/1bb734f314b8be319149ddee80b18bd41372bdcfbdf88d28131c0cd37719/rpds_py-2026.6.3-cp315-cp315-m
anylinux_2_31_riscv64.whl", hash = "sha256:a9f464559036b81bbdb36b9c8e0ea0d1c3fee968c4d59db0344c14087ef143a", size = 378827, upload-time = "2026-06-30
T07:16:56.841Z" },
    { url = "https://files.pythonhosted.org/packages/4b/93/d9611e5b25e26df9a3649813ed66193ace9347a7c7fc4ab7cf70e94851c0/rpds_py-2026.6.3-cp315-cp315-m
anylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:e55d236be29255554da47abe5c577637db7c24a02b8b46f0ca9524c855801868", size = 395966, upload-time =
 "2026-06-30T07:16:58.557Z" },
    { url = "https://files.pythonhosted.org/packages/c3/cb/99d77e16e5534ae1d90629bbe419ba6ee170833a6a85e3aa1cc41726fbbc/rpds_py-2026.6.3-cp315-cp315-m
usllinux_1_2_aarch64.whl", hash = "sha256:24e9c5386e16669b674a69c156c8eeefcb578f3b3397b713b08e6d60f3c7b187", size = 545680, upload-time = "2026-06-30T
07:17:00.164Z" },
    { url = "https://files.pythonhosted.org/packages/59/15/11a29755f790cef7a2f755e8e14f4f0c33f39489e1893a632a2eee59672b/rpds_py-2026.6.3-cp315-cp315-m
usllinux_1_2_i686.whl", hash = "sha256:c60924535c75f1566b6eb75b5c31a48a43fef04fa2d0d201acbad8a9969c6107", size = 611853, upload-time = "2026-06-30T07:
17:01.962Z" },
    { url = "https://files.pythonhosted.org/packages/68/86/0c27547e21644da938fb530f7e1a8148dd24d02db07e7a5f2567a17ce710/rpds_py-2026.6.3-cp315-cp315-m
usllinux_1_2_x86_64.whl", hash = "sha256:38a2fea2787428f811719ceb9114cb78964a3138838320c29ac39526c79c16ba", size = 573715, upload-time = "2026-06-30T0
7:17:03.693Z" },
    { url = "https://files.pythonhosted.org/packages/29/71/4d8fcf700931815594bce892255bbd973b94efaf0fc1932b0590df18d886/rpds_py-2026.6.3-cp315-cp315-w
in32.whl", hash = "sha256:d483fe17f01ad64b7bf7cc38fcefff1ca9fb83f8c2b2542b68f97ffe0611b369", size = 202864, upload-time = "2026-06-30T07:17:05.746Z" }
,
    { url = "https://files.pythonhosted.org/packages/eb/62/b577562de0edbb55b2be85ce5fd09c33e386b9b13eee09833af4240fd5c4/rpds_py-2026.6.3-cp315-cp315-w
in_amd64.whl", hash = "sha256:67e3a721ffc5d8d2210d3671872298c4a84e4b8035cfe42ffd7cde35d772b146", size = 220430, upload-time = "2026-06-30T07:17:07.471
Z" },
    { url = "https://files.pythonhosted.org/packages/c8/95/d6d0b2509825141eef60669a5739eec88dbc6a48053d6c92993a5704defe/rpds_py-2026.6.3-cp315-cp315-w
in_arm64.whl", hash = "sha256:6e84adbcf4bf841aed8116a8264b9f50b4cb3e7bd89b516122e616ac56ca269e", size = 215877, upload-time = "2026-06-30T07:17:09.008
Z" },
    { url = "https://files.pythonhosted.org/packages/b7/bf/f3ea278f0afd615c1d0f19cb69043a41526e2bb600c2b536eb192218eb27/rpds_py-2026.6.3-cp315-cp315t-
macosx_10_12_x86_64.whl", hash = "sha256:ae6dd8f10bd17aad820876d24caec9efdafd80a318d16c0a48edb5e136902c6b", size = 346933, upload-time = "2026-06-30T0
7:17:10.762Z" },
    { url = "https://files.pythonhosted.org/packages/9d/29/9907bdf1c5346763cf10b7f6852aad86652168c259def904cbe0082c5864/rpds_py-2026.6.3-cp315-cp315t-
macosx_11_0_arm64.whl", hash = "sha256:bdbd97738551fca3917c1bd7188bec1920bb520104f28e7e1007f9ceb17b7690", size = 340274, upload-time = "2026-06-30T07:
17:12.266Z" },
    { url = "https://files.pythonhosted.org/packages/6f/2c/8e03767b5778ef25cebf74a7a91a2c3806f8eced4c92cb7406bbe060756d/rpds_py-2026.6.3-cp315-cp315t-
manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:8b95977e7211527ab0ba576e286d023389fbeeb32a6b7b771665d333c60e5342", size = 370763, up
load-time = "2026-06-30T07:17:14.107Z" },
    { url = "https://files.pythonhosted.org/packages/2e/e1/df2a7e1ba2efd796af26194250b8d42c821b46592311595162af9ef0528d/rpds_py-2026.6.3-cp315-cp315t-
manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:d15fde0e6fb0d88a60d221204873743e5d9f0b7d29165e62cd86d0413ad74ba6", size = 376467, uplo
ad-time = "2026-06-30T07:17:15.76Z" },
    { url = "https://files.pythonhosted.org/packages/6b/de/8a0814d1946af29cb068fb259aa8622f856df1d0bab58429448726b537f5/rpds_py-2026.6.3-cp315-cp315t-
manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:a136d453475ac0fcbda502ef1e6504bd28d6d904700915d278deeab0d00fe140", size = 496689, up
load-time = "2026-06-30T07:17:17.308Z" },
    { url = "https://files.pythonhosted.org/packages/df/f3/f19e0c852ba13694f5a79f3b719331051573cb5693feacf8a88ffffc3a71/rpds_py-2026.6.3-cp315-cp315t-
manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:f826877d462181e5eb1c26a0026b8d0cab05d99844ecb6d8bf3627a2ca0c0442", size = 385340, upload
-time = "2026-06-30T07:17:18.928Z" },
    { url = "https://files.pythonhosted.org/packages/e2/ae/7ec3a9d2d4351f99e37bcb06b6b6f954512646bfdbf9742e1de727865daf/rpds_py-2026.6.3-cp315-cp315t-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:79486287de1730dbaff3dbd124d0ca4d2ef7f9d29bf2544f1f93c09b5bcbbd12", size = 372179, uplo
ad-time = "2026-06-30T07:17:20.539Z" },
    { url = "https://files.pythonhosted.org/packages/d3/ac/9cee911dff2aaa9a5a8354f6610bf2e6a616de9197c5fff4f54f82585f1e/rpds_py-2026.6.3-cp315-cp315t-
manylinux_2_31_riscv64.whl", hash = "sha256:808345f53cb952433ca2816f1604ff3515608a81784954f38d4452acfe8e61d5", size = 379993, upload-time = "2026-06-3
0T07:17:22.212Z" },
    { url = "https://files.pythonhosted.org/packages/83/6b/7c2a07ba88d1e9a936612f7a5d067467ed03d971d5a06f7d309dff044a7e/rpds_py-2026.6.3-cp315-cp315t-
manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:1967debc37f64f2c4dc90a7f563aec558b471966e12adcac4e1c4240496b6ebf", size = 398909, upload-time
= "2026-06-30T07:17:23.66Z" },
    { url = "https://files.pythonhosted.org/packages/97/0b/776ffcb66783637b0031f6d58d6fb55913c8b5abf00aeecd46bf933fb477/rpds_py-2026.6.3-cp315-cp315t-
musllinux_1_2_aarch64.whl", hash = "sha256:f0840b5b17057f7fd918b76183a4b5a0635f43e14eb2ce60dce1d4ee4707ea00", size = 546584, upload-time = "2026-06-30
T07:17:25.264Z" },
    { url = "https://files.pythonhosted.org/packages/55/33/ba3bc04d7092bd553c9b2b195624992d2cc4f3de1f380b7b93cbee67bd79/rpds_py-2026.6.3-cp315-cp315t-
musllinux_1_2_i686.whl", hash = "sha256:faa679d19a6696fd54259ad321251ad77a13e70e03dd834daa762a44fb6196ef", size = 614357, upload-time = "2026-06-30T07
:17:26.888Z" },
    { url = "https://files.pythonhosted.org/packages/8b/71/14edf065f04630b1a8472f7653cad03f6c478bcf95ea0e6aed55451e33ea/rpds_py-2026.6.3-cp315-cp315t-
musllinux_1_2_x86_64.whl", hash = "sha256:23a439f31ccbeff1574e24889128821d1f7917470e830cf6544dced1c662262a", size = 576533, upload-time = "2026-06-30T
07:17:28.546Z" },
    { url = "https://files.pythonhosted.org/packages/ba/76/65002b08596c389105720a8c0d22298b8dc25a4baf89b2ce431343c8b1de/rpds_py-2026.6.3-cp315-cp315t-
win32.whl", hash = "sha256:913ca42ccad3f8cc6e292b587ae8ae49c8c823e5dce51a736252fc7c7cdfa577", size = 201204, upload-time = "2026-06-30T07:17:30.193Z"
},
    { url = "https://files.pythonhosted.org/packages/8c/97/d855d6b3c322d1f27e26f5241c42016b56cf01377ea8ed348285f54652f0/rpds_py-2026.6.3-cp315-cp315t-
win_amd64.whl", hash = "sha256:ae3d4fe8c0b9213624fdce7279d70e3b148b682ca20719ebd193a23ebfa47324", size = 220719, upload-time = "2026-06-30T07:17:31.78
8Z" },
]

[[package]]
name = "rsa"
version = "4.9.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "pyasn1" },
]
sdist = { url = "https://files.pythonhosted.org/packages/da/8a/22b7beea3ee0d44b1916c0c1cb0ee3af23b700b6da9f04991899d0c555d4/rsa-4.9.1.tar.gz", hash =
"sha256:e7bdbfdb5497da4c07dfd35530e1a902659db6ff241e39d9953cad06ebd0ae75", size = 29034, upload-time = "2025-04-16T09:51:18.218Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/64/8d/0133e4eb4beed9e425d9a98ed6e081a55d195481b7632472be1af08d2f6b/rsa-4.9.1-py3-none-any.whl", h
ash = "sha256:68635866661c6836b8d39430f97a996acbd61bfa49406748ea243539fe239762", size = 34696, upload-time = "2025-04-16T09:51:17.142Z" },
]

[[package]]
name = "ruff"
version = "0.16.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/70/25/7113f6d5498888c5fb7db34081cba7d5971c4cb1bfb26819966eee68f003/ruff-0.16.1.tar.gz", hash
= "sha256:fedad7c801dabd3fb9741d76aca39246e6ddd9ca446a015875207bf19f1e6bc7", size = 4877500, upload-time = "2026-07-30T19:37:01.379Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/1b/bd/694da69368e0973de65df2ddc73ab18d43c469d5963d9b150911de6bc513/ruff-0.16.1-py3-none-linux_arm
v6l.whl", hash = "sha256:58edb313b88f0c5460a26adf5f39a37a3be789494a15e3e411e35fa78b89f9a0", size = 10839126, upload-time = "2026-07-30T19:36:13.697Z"
```

```
},
    { url = "https://files.pythonhosted.org/packages/3f/f0/b626e5d5bd0dd9576263658ef12885e2288afd1029a48e26ffed65ec1ac1/ruff-0.16.1-py3-none-macosx_10_12_x86_64.whl", hash = "sha256:fde5a99e2f97479af66edd6622c6d5a2a7592c77cf4153d9e4428f5eeb55b60c", size = 11070253, upload-time = "2026-07-30T19:36:17.14Z" },
    { url = "https://files.pythonhosted.org/packages/83/63/f40acfb6b35b88623e71684942b552c3edd96035f5d98f313815f7b277de/ruff-0.16.1-py3-none-macosx_11_0_arm64.whl", hash = "sha256:e0d4c20532fca4f7fa609369161d968dd28f65d83dabbd61d8e9c7edbf7001f6", size = 10561425, upload-time = "2026-07-30T19:36:20.04Z" },
    { url = "https://files.pythonhosted.org/packages/aa/dd/14ec0e9c2b4d315547dd38765004b4863e354e1b52cb308272215d9f6f6d/ruff-0.16.1-py3-none-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:30affbcedf59ad5703d9c91f82266e02b47739f797e1a7b6e158e5526a6dae38", size = 10948879, upload-time = "2026-07-30T19:36:22.476Z" },
    { url = "https://files.pythonhosted.org/packages/33/e9/9d870cbae575030fdef595f04b4b97573c525b5497cce4f4498cf2f85446/ruff-0.16.1-py3-none-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:24e9c631573cbca9d20f1283f8f479b2afa4a8503504822bd71a293889f16743", size = 10643691, upload-time = "2026-07-30T19:36:24.914Z" },
    { url = "https://files.pythonhosted.org/packages/c4/09/12743d544e2173f53ecd27217c65f90d2bc0f8424a66a60339e56bbc0457/ruff-0.16.1-py3-none-manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:b41bdd48fb420987a9b5212e4957c26ad4abce401fa9ea9d4d85843727945f4f", size = 11435354, upload-time = "2026-07-30T19:36:28.447Z" },
    { url = "https://files.pythonhosted.org/packages/7f/89/a1652b2daee52083c9554a6333b678a8b01d0400f976827bb87857f9449a/ruff-0.16.1-py3-none-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:b0d1e1393b7648079e13669de1c1f4fde06d4583e84d8fd5c1551e0a77a2aa75", size = 12259033, upload-time = "2026-07-30T19:36:31.326Z" },
    { url = "https://files.pythonhosted.org/packages/16/96/ecdcb8c54ee7b123b487f807eb014e6e019155a0b81dfb669acd52f28ce3/ruff-0.16.1-py3-none-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:07bf434b1c95f4e093be4532068ef4fcf00924eb2ade879607598090 2d6fd54a", size = 11667981, upload-time = "2026-07-30T19:36:34.394Z" },
    { url = "https://files.pythonhosted.org/packages/cd/90/c52e12e0d862e9572f2a33aa227409143520abe53111e9a6babbac7b4af8/ruff-0.16.1-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:39897739f112253ee4fdd2e8aa9a4f9ded99fb2be367d5f31dfa4ded6025584c", size = 11468183, upload-time = "2026-07-30T19:36:37.339Z" },
    { url = "https://files.pythonhosted.org/packages/2c/6b/4ffb7ad1d83eb16cf8cbb3c8815d3f11c88460fd162d4b372a2059be1c2a/ruff-0.16.1-py3-none-manylinux_2_31_riscv64.whl", hash = "sha256:82ae3c0c0d74daf17b968a10b7b3bb3ef297ab7de0c1f749646b25e690ccb150", size = 11470071, upload-time = "2026-07-30T19:36:39.91Z" },
    { url = "https://files.pythonhosted.org/packages/9c/72/32ae7db4c0b5e32ab611787caa19d1546800676d79f7483b7100a3561bf4/ruff-0.16.1-py3-none-musllinux_1_2_aarch64.whl", hash = "sha256:4d5f2ed10f8242d83fc08d521301089364e3375375705356f20c0e31606ef3ef", size = 10919503, upload-time = "2026-07-30T19:36:42.65Z" },
    { url = "https://files.pythonhosted.org/packages/f7/ca/3d901ba6ad6fc38da39c3448fc6c59ac945679293a17c3ceb6d6c1cba13e/ruff-0.16.1-py3-none-musllinux_1_2_armv7l.whl", hash = "sha256:a4665b309891f83f3e3c25447935f1213e9abbd4b5640af7a1f2def9f8d413c1", size = 10649861, upload-time = "2026-07-30T19:36:45.18Z" },
    { url = "https://files.pythonhosted.org/packages/92/79/894ef1ced26552d5f8c9cf6d85b0687840e1128c55aeab7b9c2d54a0d880/ruff-0.16.1-py3-none-musllinux_1_2_i686.whl", hash = "sha256:26e9ca5c9bc3971f20d3cf18a957f52ffd6a5f6564ff15c4912a144dcac22494", size = 11148137, upload-time = "2026-07-30T19:36:47.936Z" },
    { url = "https://files.pythonhosted.org/packages/2d/69/3609a09fa1cb46cc28b762363e440a354204e5dff01bd0c8d7437874d6b9/ruff-0.16.1-py3-none-musllinux_1_2_x86_64.whl", hash = "sha256:67e1e1e3fa4f0c82f0e36d4cd61e661f6e7a6196cb1aa92fe0828fa7b8f257cd", size = 11559211, upload-time = "2026-07-30T19:36:50.448Z" },
    { url = "https://files.pythonhosted.org/packages/fc/8a/fb22af2fd78a736e241fabf67e30ce1799a64244026377a49e133af90762/ruff-0.16.1-py3-none-win32.whl", hash = "sha256:d31765e131295b8445caf301e3e8a85b34d1b9b211b4109b7ba457888b051806", size = 10838258, upload-time = "2026-07-30T19:36:53.298Z" },
    { url = "https://files.pythonhosted.org/packages/d4/35/e57fd9fb5d423961df087a00b12d42c0a830288dc2f3b45ecca299158b4f/ruff-0.16.1-py3-none-win_amd64.whl", hash = "sha256:09b05e8b90c2cb06ad63464350e7a45e8e44a2dfe52072ebfba6666ca8d3f596", size = 11961111, upload-time = "2026-07-30T19:36:56.107Z" },
    { url = "https://files.pythonhosted.org/packages/cb/46/240ea004bf6dc4feb40e9832f2205a476a47dd5b8a3f8211a5fc5f95e20e/ruff-0.16.1-py3-none-win_arm64.whl", hash = "sha256:dbaadaac38c70239f056d306b7476f246b0bf000fa6b3876402acbf5b227eaf8", size = 11309414, upload-time = "2026-07-30T19:36:58.79Z" },
]

[[package]]
name = "setuptools"
version = "83.0.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/34/26/f5d29e25ffdb535afef2d35cdb55b325298f96debd670da4c325e08d70f4/setuptools-83.0.0.tar.gz", hash = "sha256:025bccbbf0fa05b6192bc64ae1e7b16e001fd6d4d5de03c97b1c1ade523bef", size = 1154254, upload-time = "2026-07-04T15:31:22.699Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/5d/40/e1e72872c6354b306daef1703549e8e83b4d43cfea356311bf722a043752/setuptools-83.0.0-py3-none-any.whl", hash = "sha256:29b23c360f22f414dc7336bb39178cc7bcbf6021ed2733cde173f09dba19abb3", size = 1008090, upload-time = "2026-07-04T15:31:20.885Z" },
]

[[package]]
name = "six"
version = "1.17.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/94/e7/b2c673351809dca68a0e064b6af791aa332cf192da575fd474ed7d6f16a2/six-1.17.0.tar.gz", hash = "sha256:ff70335d468e7eb6ec65b95b99d3a2836546063f63acc5171de367e834932a81", size = 34031, upload-time = "2024-12-04T17:35:28.174Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/b7/ce/149a00dd41f10bc29e5921b496af8b574d8413afcd5e30dfa0ed46c2cc5e/six-1.17.0-py2.py3-none-any.whl", hash = "sha256:4721f391ed90541fddacab5acf947aa0d3dc7d27b2e1e8eda2be8970586c3274", size = 11050, upload-time = "2024-12-04T17:35:26.475Z" },
]

[[package]]
name = "sniffio"
version = "1.3.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a2/87/a6771e1546d97e7e041b6ae58d80074f81b7d5121207425c964ddf5cfdbd/sniffio-1.3.1.tar.gz", hash = "sha256:f4324edc670a0f49750a81b895f35c3adb843cca46f0530f79fc1babb23789dc", size = 20372, upload-time = "2024-02-25T23:20:04.057Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e9/44/75a9c9421471a6c4805dbf2356f7c181a29c1879239abab1ea2cc8f38b40/sniffio-1.3.1-py3-none-any.whl", hash = "sha256:2f6da418d1f1e0fddd844478f41680e794e6051915791a034ff65e5f100525a2", size = 10235, upload-time = "2024-02-25T23:20:01.196Z" },
]

[[package]]
name = "sqlalchemy"
version = "2.0.51"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "greenlet", marker = "platform_machine == 'AMD64' or platform_machine == 'WIN32' or platform_machine == 'aarch64' or platform_machine == 'amd64' or platform_machine == 'ppc64le' or platform_machine == 'win32' or platform_machine == 'x86_64'" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/02/f1/a7a892f18d4d224e6b26f706531eafccc41e37594d37d304786969ee13cb/sqlalchemy-2.0.51.tar.gz", hash = "sha256:804dccd8a4a6242c4e30ad961e540e18a588f6527202f2d6791b01845d59fdc9", size = 9912201, upload-time = "2026-06-15T15:41:20.012Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d5/70/e868bc5412acd101a8280f25c95f10eeae0771c4eb806b02491142810ee8/sqlalchemy-2.0.51-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:7d78702b26ba1c18b2d0fb2ea940ba7f17a9581b42e8361ff93920ebbee1235a", size = 2160291, upload-time = "2026-06-15T16:08:48.918Z" },
    { url = "https://files.pythonhosted.org/packages/e5/1c/71ee0f8a6b9d7316a1ccd30430b4c62b6c2e36adc96017a4e3a72dce49d6/sqlalchemy-2.0.51-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:581921d849d6e6f994d560389192955e80e2950e18fcdfe2ccea863e01158e6e", size = 3343835, upload-time = "2026-06-15T16:19:42.613Z" },
    { url = "https://files.pythonhosted.org/packages/2b/7c/7ab9f9aadc5944fdd06612484ed7918fe376ad871a5f50404dc1536e0194/sqlalchemy-2.0.51-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:1d21ce524ab86c23046e992a5b81cb54c21079c6df6e78b8fc77d77cac70a6b9", size = 3358470, upload-time = "2026-06-15T16:26:38.011Z" },
    { url = "https://files.pythonhosted.org/packages/d0/7d/ff77169fee6186de145a7f2b87006c39638391130abbab2b1f63ac6ea583/sqlalchemy-2.0.51-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:c5d98a2709840027f5a347c3af0a7c3d5f6c1ff93af2ca1c54494e23cba8f389", size = 3289874, upload-time = "2026-06-15T16:19:45.212Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/6f/3b/6c505903710d781b55bc3141ee34a062bf9745a6b5bc7333305b9ed63b33/sqlalchemy-2.0.51-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:1181256e0f16479691b5616d36375dc2620ad8332b25978763c3d206ad3f3f1d", size = 3321692, upload-time = "2026-06-15T16:26:39.747Z" },
    { url = "https://files.pythonhosted.org/packages/3c/b7/c5ffe50aa2f4d947c9250e1519d939260329a07fe6272edfccd784b3d007/sqlalchemy-2.0.51-cp312-cp312-win32.whl", hash = "sha256:9f380393be5abeb6815f68fd39271b95127173511b6706b0a630a9995d53f8f5", size = 2119674, upload-time = "2026-06-15T16:23:09.543Z" },
    { url = "https://files.pythonhosted.org/packages/25/dc/46a65916af68a06ef6b972c6050ba4c8f97070fe3fb33097d34229d9bef6/sqlalchemy-2.0.51-cp312-cp312-win_amd64.whl", hash = "sha256:2cf39aabdf48e87c1c2c2ed6d20d33ffa0733b3071ce9c5f66357947dd009080", size = 2146670, upload-time = "2026-06-15T16:23:11.048Z" },
    { url = "https://files.pythonhosted.org/packages/54/fe/a210d52fd1a90ecfae8a78e9d8b27e18d733d60818a8bf250ff690b75120/sqlalchemy-2.0.51-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:7c2056838b6685b72fdb36c99996cf862753461a62f2e84f4196371d3b2d6a07", size = 2157184, upload-time = "2026-06-15T16:08:50.374Z" },
    { url = "https://files.pythonhosted.org/packages/17/6b/2dce8369b199cb855110e056032f94a9f66dacc2237d3d39c115a86eac56/sqlalchemy-2.0.51-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:483b11bd46bf35fc14c52faf338b04300c9e6ce554bce9b11be85bfec3bc3195", size = 3284735, upload-time = "2026-06-15T16:19:46.934Z" },
    { url = "https://files.pythonhosted.org/packages/53/ff/dbc495b8a14da840faffb353857a72d4190113cac33727906fb997047f0f/sqlalchemy-2.0.51-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:1bed1ee8b01da6088210aa9412023326fb98a599ba502e6118308601dcbef77f", size = 3302756, upload-time = "2026-06-15T16:26:41.336Z" },
    { url = "https://files.pythonhosted.org/packages/cf/d5/fde8f4dddcf518ee15ab35a7c6a28acc32c8ba548d1d2aa451f96e6dbb0b/sqlalchemy-2.0.51-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:72ca54c952107ba5cd58854b67a5a6268631289d21651a1235396f3b98b47400", size = 3232055, upload-time = "2026-06-15T16:19:49.286Z" },
    { url = "https://files.pythonhosted.org/packages/67/d1/43d3a0ac955a58601c24fa23038b1c55ee3a1ec02c0f96ebb1eae2bcf614/sqlalchemy-2.0.51-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:b3e693d15533a45cd5906f0589f9c35090bef6ef45bf1e8195c424aa0ae06a8d", size = 3269850, upload-time = "2026-06-15T16:26:43.017Z" },
    { url = "https://files.pythonhosted.org/packages/94/df/de669c7054cd47c4439ac34b1b2ee8b804a794791fbb10720e997a2c87c7/sqlalchemy-2.0.51-cp313-cp313-win32.whl", hash = "sha256:b93ab07b5292dbe7e6b8da89475275e7042744283921344b56105f3eeb0f828b", size = 2117721, upload-time = "2026-06-15T16:23:12.36Z" },
    { url = "https://files.pythonhosted.org/packages/d0/8a/403c51d064196bae20a0bc2476577f83a3f8dd299719a97417086b7f2ec5/sqlalchemy-2.0.51-cp313-cp313-win_amd64.whl", hash = "sha256:0f053118c30e53161857a953e4de667d90e274980dccbe5dd3829bbbeece72a5", size = 2143615, upload-time = "2026-06-15T16:23:13.906Z" },
    { url = "https://files.pythonhosted.org/packages/b1/49/a739be2e1d02a96a658eb71ab45d921c874249252358ad24a5bffdd02525/sqlalchemy-2.0.51-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:6ea306caaae6bd5afd0a46050003c88f6bf33227377a49298c498c3cb88ff491", size = 2158999, upload-time = "2026-06-15T16:08:51.759Z" },
    { url = "https://files.pythonhosted.org/packages/23/6b/2e0e38cf75c8780eca78d9b2e78164f8bcfd70125e5caa588ff5cbb9c9f4/sqlalchemy-2.0.51-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:c45a496d6bc05dec41dcd4c3a2b183723f47473255c159cd80b503c8f246424d", size = 3282539, upload-time = "2026-06-15T16:19:51.065Z" },
    { url = "https://files.pythonhosted.org/packages/dd/a1/e77854cb5336fd37dc3c6ae3b71de242c98caac5725120be0b526b31cbd0/sqlalchemy-2.0.51-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:4004ada0aafe8ae1991b2cd1d99c6d9146126e123bd6f883c260d974aa012e54", size = 3287545, upload-time = "2026-06-15T16:26:44.735Z" },
    { url = "https://files.pythonhosted.org/packages/f6/ab/9e17272fd4dac8df3b83c4fbe52b998a1c9d89a843c8c35ff29b74ff7364/sqlalchemy-2.0.51-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:0f6bcad487aee1c638d707235682fc96f741de00663619881ab235400d03289e", size = 3230929, upload-time = "2026-06-15T16:19:52.625Z" },
    { url = "https://files.pythonhosted.org/packages/02/3c/52f408ea701781caee975606becc c48845f2aee8711ac29843d612c0306c/sqlalchemy-2.0.51-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:39a76529db6305693d8d4affa58ad5b5e2e18edd62daea628b29b97930b3513d", size = 3252888, upload-time = "2026-06-15T16:26:46.454Z" },
    { url = "https://files.pythonhosted.org/packages/24/16/3efd2ee6bc4ca4693a30a1dd17a91b606cae15d517d2a4746611d9b73ce8/sqlalchemy-2.0.51-cp314-cp314-win32.whl", hash = "sha256:08a204d8b5638717c26a24df18fcf40af45a6b22e35b70b1d62f0113c2e278e8", size = 2120551, upload-time = "2026-06-15T16:23:15.629Z" },
    { url = "https://files.pythonhosted.org/packages/7b/78/55b12e70f45bccc40d9e483925c065027b3b98ea4cbbdf6f8c2546feaf6c/sqlalchemy-2.0.51-cp314-cp314-win_amd64.whl", hash = "sha256:96747bfbadb055466e5b46d572618170046b45ce5a4879167f50d70a5319a499", size = 2146318, upload-time = "2026-06-15T16:23:17.108Z" },
    { url = "https://files.pythonhosted.org/packages/21/db/a9574ed40fed418924b1b1a3e54f47ee3963053b3d3d325a0d36b41f2c08/sqlalchemy-2.0.51-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:e5ea1a213be1fcd5e49d9904c3b9939211ded90bc2a64e93f4c01963474285de", size = 2178920, upload-time = "2026-06-15T15:59:56.285Z" },
    { url = "https://files.pythonhosted.org/packages/bf/90/a1bb5c7cbba76b7bc1fbd586d0a5479a7bc9c27b4a8298f22ec9423b2bb3/sqlalchemy-2.0.51-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:7c6b36ed71f41942bdcd2ad2522be46bfce09d5705be5640ecf19bbc7660e4b7", size = 3566534, upload-time = "2026-06-15T15:58:35.024Z" },
    { url = "https://files.pythonhosted.org/packages/15/4b/481f1fed30e0e9e8dd24aecbb49f29eb57fe7657ece5cf06ee9b84bb97d8/sqlalchemy-2.0.51-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:0c2c62877097e1a0db401fba5cb4debee33265e5b2a55c4ccb489c02c53b4f72", size = 3535844, upload-time = "2026-06-15T16:02:43.973Z" },
    { url = "https://files.pythonhosted.org/packages/02/71/0aa64aeda645510af0a43f7d9ee70932f0d1dc4263aed34c50ee891d9df3/sqlalchemy-2.0.51-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:0378d055e9e8cd6ce4d8dff683bdd3d7d413533c4ee51d67a2b1e0f9eacc0f23", size = 3475355, upload-time = "2026-06-15T15:58:36.592Z" },
    { url = "https://files.pythonhosted.org/packages/05/db/6061db32316446135a3abae5f308d144ab988a34234726042da3e58b1c63/sqlalchemy-2.0.51-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:6e46fc36029eff666391e0531e5387b62ce6c4f1d8e50b3fb3099eaca1b42522", size = 3486591, upload-time = "2026-06-15T16:02:45.346Z" },
    { url = "https://files.pythonhosted.org/packages/0d/c9/f14fdf71bb8957e0c7e39db69bbdf12b5c80f4ef775fdfa127bf4e0d6760/sqlalchemy-2.0.51-cp314-cp314t-win32.whl", hash = "sha256:9161cfc9efce70d1715f47d6ff40f79c6778c00d53be4fbc09d70301e4b83ba7", size = 2151313, upload-time = "2026-06-15T16:03:39.127Z" },
    { url = "https://files.pythonhosted.org/packages/6a/c6/673e618e6f4f297e126d9b56ea2f6478708f6c1af4e3223835c22e2c3697/sqlalchemy-2.0.51-cp314-cp314t-win_amd64.whl", hash = "sha256:159bb6ba32059f57ad7375a8f50d844dd2f19d14954ecf820cd33e20debd46b2", size = 2186280, upload-time = "2026-06-15T16:03:40.569Z" },
    { url = "https://files.pythonhosted.org/packages/e2/22/dbf013a12ec759e54a34a119e9e217435b3f71b2dd5c61a7ade0a25dae87/sqlalchemy-2.0.51-py3-none-any.whl", hash = "sha256:bb024d8b621d0be75f4f44ecc7c950450026e76d66dc8f791bb5331d7fed59d5", size = 1944334, upload-time = "2026-06-15T16:09:22.418Z" },
]

[package.optional-dependencies]
asyncio = [
    { name = "greenlet" },
]

[[package]]
name = "sse-starlette"
version = "3.4.8"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "starlette" },
]
sdist = { url = "https://files.pythonhosted.org/packages/f8/00/b42a44342a054d58cb1115d7c8aa9cb4290dd9442f9c1b91a4b8173dba22/sse_starlette-3.4.8.tar.gz", hash = "sha256:ed89ffbb75cbf78a5fe2f2109cd584792ee7f9dfac96f791db546df8f15f3f9c", size = 32548, upload-time = "2026-08-05T11:19:49.982Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/dd/3a/764912c58293d95b6dcdf4cc255f9d10de310580ced547b082eb9d72018c/sse_starlette-3.4.8-py3-none-any.whl", hash = "sha256:6e82314c786709a3cd9520f2285cf9fff90e181e598e8a357b0cf80f66afba0d", size = 16516, upload-time = "2026-08-05T11:19:48.748Z" },
]

[[package]]
name = "starlette"
version = "1.4.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/0f/3c/76d2fd1f1357ed0f0108d8a5aa233dcf16e2946a8559c84912fe08e01ac7/starlette-1.4.1.tar.gz", h
```

```
ash = "sha256:b7332de6e9375593a29ba9eee1e6ecfeb3eb2043e2e19a13b4b71da73ff35540", size = 2709041, upload-time = "2026-08-05T15:17:26.23Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/75/0a/67e95f21498de41433babf7b1db0eeab449eb58872dfb831b27747a70fd0/starlette-1.4.1-py3-none-any.whl", hash = "sha256:7d078e0fbefae0d2cecfb80a799d6fb84b1c0c6acd4f14ac79d17d0e7ec27f19", size = 74019, upload-time = "2026-08-05T15:17:24.357Z" },
]

[[package]]
name = "tenacity"
version = "9.1.4"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/47/c6/ee486fd809e357697ee8a44d3d69222b344920433d3b6666ccd9b374630c/tenacity-9.1.4.tar.gz", hash = "sha256:adb31d4c263f2bd041081ab33b498309a57c77f9acf2db65aadf0898179cf93a", size = 49413, upload-time = "2026-02-07T10:45:33.841Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d7/c1/eb8f9debc45d3b7918a32ab756658a0904732f75e555402972246b0b8e71/tenacity-9.1.4-py3-none-any.whl", hash = "sha256:6095a360c919085f28c6527de529e76a06ad89b23659fa881ae0649b867a9d55", size = 28926, upload-time = "2026-02-07T10:45:32.24Z" },
]

[[package]]
name = "tiktoken"
version = "0.13.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "regex" },
    { name = "requests" },
]
sdist = { url = "https://files.pythonhosted.org/packages/e4/e5/5f3cb2159769d0f4324c0e9e87f9de3c4b1cd45848a96b2eb3566ad5ca77/tiktoken-0.13.0.tar.gz", hash = "sha256:c9435714c3a84c2319499de9a300c0e604449dd0799ff246458b3bb6a7f433c1", size = 38986, upload-time = "2026-05-15T04:51:27.153Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/85/8e/144bde4e01df66b34bb865557c7cd754ed08b036217ebd79c9db5e9048a9/tiktoken-0.13.0-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:32ac870a806cfb260a02d0cb70426aef02e038297f8ad50df5040bb5af360791", size = 1034888, upload-time = "2026-05-15T04:50:31.579Z" },
    { url = "https://files.pythonhosted.org/packages/36/18/d4ac9d20956cdebca04841316660ed584c2fecdc2b81722a28bc7ad3b1e4/tiktoken-0.13.0-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:4d9980f11429ed2d737c463bb1fb78cf330caa026adf002f714aced7849a687b", size = 982970, upload-time = "2026-05-15T04:50:32.961Z" },
    { url = "https://files.pythonhosted.org/packages/74/ed/6bb8d05b9f731f749fee5c6f5ca63e981143c826a5985877330507bd13b7/tiktoken-0.13.0-cp312-cp312-manylinux_2_28_aarch64.whl", hash = "sha256:3f277ebea5edd7b8bf03c6f9431e1d67d517530115572b2dc1d465326e8f88c7", size = 1115741, upload-time = "2026-05-15T04:50:34.475Z" },
    { url = "https://files.pythonhosted.org/packages/34/de/2ca96b07a82d972b74fe4b46de055b79c904e45c7eab699354a0bfa697dc/tiktoken-0.13.0-cp312-cp312-manylinux_2_28_x86_64.whl", hash = "sha256:a116178fa7e1b4065bff05214360373a65cac22f965be7b3f73d00a0dbfe7649", size = 1136523, upload-time = "2026-05-15T04:50:35.782Z" },
    { url = "https://files.pythonhosted.org/packages/ee/dc/9dafec002c2d4424378563cf4cf5c7fb93631d2a55013c8b87554ee4012c/tiktoken-0.13.0-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:2c397ddda233208345b01bd30f2fca79ff730e55731d0108a603f9bc57f6af3b", size = 1181954, upload-time = "2026-05-15T04:50:36.99Z" },
    { url = "https://files.pythonhosted.org/packages/a1/d0/1f8578c45b2f24759b46f0b50d31878c63c73e6bf0f2227e10ec5c5408dc/tiktoken-0.13.0-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:95097e4f89b06403976e498abf61a0ee73a7497e73fb599cb211d8197a054d91", size = 1240069, upload-time = "2026-05-15T04:50:38.221Z" },
    { url = "https://files.pythonhosted.org/packages/aa/90/28d7f154888610aa9237e541986beb62b479df29d193a5a0617dbb1514d0/tiktoken-0.13.0-cp312-cp312-win_amd64.whl", hash = "sha256:8f2d16e7a7c783ad81f36e457d046d1f1c8af70b22aec8a13238efe531977c41", size = 874748, upload-time = "2026-05-15T04:50:39.587Z" },
    { url = "https://files.pythonhosted.org/packages/9c/83/b096c859c2a47c11731bf2f5885f4028b809dfe2396582883eed9cae372f/tiktoken-0.13.0-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:5df5d1507bd245f1ccad4a074698240021239e455eb0bb4ced4e3d7181872154", size = 1034228, upload-time = "2026-05-15T04:50:40.988Z" },
    { url = "https://files.pythonhosted.org/packages/53/61/c68e123b6d753e3fc2751e9b18e732c9d8bf1e1926762e736eee935d931c/tiktoken-0.13.0-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:8fe806a50664e83a6ffd56cbd1e4f5dcc6cd32a3e7538f70dc38b1a271384545", size = 982978, upload-time = "2026-05-15T04:50:42.195Z" },
    { url = "https://files.pythonhosted.org/packages/ef/8b/96cc178cc584e65d363134500f297790b06cd48cdeb1e8fcf7bbe60f4715/tiktoken-0.13.0-cp313-cp313-manylinux_2_28_aarch64.whl", hash = "sha256:125bc05005e747f993a83dc67934249932d6e4209854452cd4c0b1d53fba3ba2", size = 1116355, upload-time = "2026-05-15T04:50:43.564Z" },
    { url = "https://files.pythonhosted.org/packages/86/f5/bab735d2c72ea55404b295d02d092644eb5f7cc6205e34d35eb9abfb9ab2/tiktoken-0.13.0-cp313-cp313-manylinux_2_28_x86_64.whl", hash = "sha256:5e6358911cab4adee6712da27d65573496a4f68cf8a2b5fca6a4ad10fc5748cf", size = 1135772, upload-time = "2026-05-15T04:50:44.782Z" },
    { url = "https://files.pythonhosted.org/packages/4e/b9/6de04ebdf904edfaad87788011b3735087a0c9ea671b9027e1e4e965e8c8/tiktoken-0.13.0-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:975cbd78d085d75d26b59660e262736dcaed1e35f8f142cd6291025c01d25486", size = 1182415, upload-time = "2026-05-15T04:50:46.422Z" },
    { url = "https://files.pythonhosted.org/packages/0d/9c/470a05f3b1caf038f44880e334d47ab674e0c80d514c66b375d14d5afa10/tiktoken-0.13.0-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:75ab9bc99fa020a4c283424590ecd7f3afd70c1c281cb3fa3192a6c3af9f9615", size = 1239879, upload-time = "2026-05-15T04:50:48.052Z" },
    { url = "https://files.pythonhosted.org/packages/42/a6/c1936d16055436cb32e6c6128d68629622e00f4768562f55653752d34768/tiktoken-0.13.0-cp313-cp313-win_amd64.whl", hash = "sha256:6b1615f0ff71953d19729ceb18865429c185b0a23c5353f1bbca34a394bf60f7", size = 874829, upload-time = "2026-05-15T04:50:49.202Z" },
    { url = "https://files.pythonhosted.org/packages/d6/07/acb5992c3772b5a36284f742cfb7a5895aa4471d1848ac31464ad50d7fdf/tiktoken-0.13.0-cp313-cp313t-macosx_10_13_x86_64.whl", hash = "sha256:6eb4a5bfbc6426938026b1a334e898ac53541360d62d8c689870160cc80abd67", size = 1033600, upload-time = "2026-05-15T04:50:50.4Z" },
    { url = "https://files.pythonhosted.org/packages/14/e9/742e9aec30f59b9f161f7ff7cd072e02ea836c9e1c0854a8076dfcd40d5c/tiktoken-0.13.0-cp313-cp313t-macosx_11_0_arm64.whl", hash = "sha256:43cee3e5400573b2046fbf092cc7a5bc30164f9e4c95ce20714da929df48737a", size = 982516, upload-time = "2026-05-15T04:50:52.03Z" },
    { url = "https://files.pythonhosted.org/packages/72/74/ca1541b053e7648254d2e4b42a253e1bb4359f2c91a0a8d49228c794e1a0/tiktoken-0.13.0-cp313-cp313t-manylinux_2_28_aarch64.whl", hash = "sha256:7de52e3f566d19b3b11bd37eea552c6c305ad74081f736882bd44d148ed4c48d", size = 1115518, upload-time = "2026-05-15T04:50:53.543Z" },
    { url = "https://files.pythonhosted.org/packages/46/e3/93825eaf5a4a504795b787e5d5dea07fbeb3dabf97aa7b450be8bde59c89/tiktoken-0.13.0-cp313-cp313t-manylinux_2_28_x86_64.whl", hash = "sha256:51384448aa508e4df84c0f7c1dc3211c7f7b8096325660ee5fc82f3e11b381ce", size = 1136867, upload-time = "2026-05-15T04:50:55.191Z" },
    { url = "https://files.pythonhosted.org/packages/8c/46/002b68de6827091d5ae90b048f326e8aad8d953520950e5ce1508879414f/tiktoken-0.13.0-cp313-cp313t-musllinux_1_2_aarch64.whl", hash = "sha256:e28157350f7ebf35008dd8e9e0fdb621f976e4230c881099c85e8cf07eaa50e2", size = 1181826, upload-time = "2026-05-15T04:50:56.296Z" },
    { url = "https://files.pythonhosted.org/packages/db/c6/d393e3185a276505182f7abd93fe714f3c444a2be9180798fa052347504e/tiktoken-0.13.0-cp313-cp313t-musllinux_1_2_x86_64.whl", hash = "sha256:165cf1820ea4a354985c2490a5205d4cc74661c934aca79dd0368232fff94e0f", size = 1239489, upload-time = "2026-05-15T04:50:57.918Z" },
    { url = "https://files.pythonhosted.org/packages/b7/4d/bc07d1f1635d4897a202acc0ae11c2886eaa7325c359ba4741b47bf8e225/tiktoken-0.13.0-cp313-cp313t-win_amd64.whl", hash = "sha256:6c43a675ca14f6f2749ba7f12075d37456015a24b859f2517b9beb4ef30807ec", size = 873820, upload-time = "2026-05-15T04:50:59.528Z" },
    { url = "https://files.pythonhosted.org/packages/8c/93/0dd6adca026a616c3a92974566b43381eea4b475ce1f36c062b8271a9ac5/tiktoken-0.13.0-cp314-cp314-macosx_10_13_x86_64.whl", hash = "sha256:eaaaef47c2406277181d2086484c317bf7fc433e2d5d03ff94f56b0dcec87471", size = 1034977, upload-time = "2026-05-15T04:51:00.957Z" },
    { url = "https://files.pythonhosted.org/packages/d9/77/5ec6e6bc5b30bed6d93f7f2162d8f6b32437b3ba27cb527cfe004f6109c9/tiktoken-0.13.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:ca8b310bd93b3772cb1b7922d915446864860f562bdfe4825c63a0aed3fb28cd", size = 983635, upload-time = "2026-05-15T04:51:02.629Z" },
    { url = "https://files.pythonhosted.org/packages/94/b0/c8ae9aff00d625c50659b4513e707a0462c4bf5d4d6cc1b802103225c02e/tiktoken-0.13.0-cp314-cp314-manylinux_2_28_aarch64.whl", hash = "sha256:32e0c12305105002c047b3bb1070b0dd9a73b0cb3b2856a8972b810e7a4f5881", size = 1116036, upload-time = "2026-05-15T04:51:04.082Z" },
    { url = "https://files.pythonhosted.org/packages/1b/ac/6a5dddd1d0a6018ecb389bd0353e6b4a515eb4d2286611bd0ace1937b9e1/tiktoken-0.13.0-cp314-cp314-manylinux_2_28_x86_64.whl", hash = "sha256:5ba5fd62507a932d1241346179e3b39bc7bf7408f03c272652d93b3bedf5db24", size = 1135544, upload-time = "2026-05-15T04:51:05.229Z" },
    { url = "https://files.pythonhosted.org/packages/f4/b8/585032b4384b2f7dcdaddcb52865c83a701a420d09e3c2b4a2be1c450c57/tiktoken-0.13.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:d108bc2d470fc53c8ecd24f2c0fd2b5f98c33e87cdb6aa2e9b8c5dced703d273", size = 1182217, upload-time = "2026-05-15T
```

```
04:51:06.517Z" },
    { url = "https://files.pythonhosted.org/packages/cd/b6/993ff1ded3958215fd341a847b8e5ffeb5de473f435296870d314fc91ac4/tiktoken-0.13.0-cp314-cp314-mu
sllinux_1_2_x86_64.whl", hash = "sha256:cb99cb5127449f58d0a2d5f5ccfb390d8dbdfd919c221246caaee29d8725ed51", size = 1239404, upload-time = "2026-05-15T0
4:51:07.843Z" },
    { url = "https://files.pythonhosted.org/packages/dd/3d/fef7e06e3b33e7538db0ced734cf9fe23b6832d2ac4990c119c377aec55e/tiktoken-0.13.0-cp314-cp314-wi
n_amd64.whl", hash = "sha256:115c4f26ffa11caac8b54eea35c2ad38c612c20a48d35dd15d70a02ac6f51f58", size = 918686, upload-time = "2026-05-15T04:51:08.925Z
" },
    { url = "https://files.pythonhosted.org/packages/c1/82/a7fc44582bc32ab00de988a2299bf77c077f59068b233109e34b7d6ca7e6/tiktoken-0.13.0-cp314-cp314t-m
acosx_10_13_x86_64.whl", hash = "sha256:472527e9132952f2fbf77cd290658bacf003d4d5a3fabc18e5fbd407cbae4d9b", size = 1034454, upload-time = "2026-05-15T0
4:51:10.035Z" },
    { url = "https://files.pythonhosted.org/packages/37/d0/24d8a890c14f432a05cea669c17bebeaa99f96a7c79523b590f564246411/tiktoken-0.13.0-cp314-cp314t-m
acosx_11_0_arm64.whl", hash = "sha256:4e2f67d27c9626cdd25fe33d9313c5cdb3d8d82da646b68d6eb8e7e9c20e6448", size = 982976, upload-time = "2026-05-15T04:5
1:11.23Z" },
    { url = "https://files.pythonhosted.org/packages/49/b7/2ab43f62788a9266187a9bfc1d3af99ad83e5eaa25fbef168a69cd5ad14f/tiktoken-0.13.0-cp314-cp314t-m
anylinux_2_28_aarch64.whl", hash = "sha256:2b920b35805cd64585a37c3dc7ce65fba4d2d36016be01e1d7942482ca29093a", size = 1115526, upload-time = "2026-05-1
5T04:51:12.608Z" },
    { url = "https://files.pythonhosted.org/packages/64/39/1494321ed323ce7a14d88e3cd6cb9058625977df1c6961ddc492bd10a9f3/tiktoken-0.13.0-cp314-cp314t-m
anylinux_2_28_x86_64.whl", hash = "sha256:493af3aa28a4aaf2e3d2600a2ee717252c9bf5ab38fff94eb5a02db5ab77e5ad", size = 1136466, upload-time = "2026-05-15
T04:51:13.926Z" },
    { url = "https://files.pythonhosted.org/packages/96/d9/dfd086aa2d918c563a140720e0ce296cada1634efd2783d5cf51e05f984e/tiktoken-0.13.0-cp314-cp314t-m
usllinux_1_2_aarch64.whl", hash = "sha256:6644c9c2b5cf3916f5a3641d7d12fdb3f006a7b3d9ff6acdaec44e29ab1ff91e", size = 1181863, upload-time = "2026-05-15
T04:51:15.025Z" },
    { url = "https://files.pythonhosted.org/packages/2f/68/a18b4f307086954fdae32714cb4f85562e34f9d34ab206e61f1816aa6018/tiktoken-0.13.0-cp314-cp314t-m
usllinux_1_2_x86_64.whl", hash = "sha256:5cb65b60b9408563676d874a3a4ee573370066f0dc4e29d84e82e989c6517424", size = 1239218, upload-time = "2026-05-15T
04:51:16.103Z" },
    { url = "https://files.pythonhosted.org/packages/16/5b/f2aa703a4fc5d2dff73460a7d46cc2f3f44aa0f3dd8eeb20d2a0ecf68862/tiktoken-0.13.0-cp314-cp314t-w
in_amd64.whl", hash = "sha256:85b78cc3a2c3d48723ca751fa981f1fedccd54194ca0471b957364353a898b07", size = 918110, upload-time = "2026-05-15T04:51:17.237
Z" },
]

[[package]]
name = "tinytag"
version = "2.3.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/9f/a4/a1d39cc10b43cbbae268127a1c38d689bc6a85cf966f9445bc9f1f5f517a/tinytag-2.3.0.tar.gz", has
h = "sha256:84850f8045424b944475b9754bc35c7e09bcae1ab08d1f88d9293aa33af39a27", size = 44379, upload-time = "2026-07-30T23:35:03.407Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/90/48/f9a955e0d27376dd0a6bcf7973818b19e5965dd847580aef2f08c6d6f64b/tinytag-2.3.0-py3-none-any.whl
", hash = "sha256:231ba5b2fb7a6db478f6dd344ebf20dfdfcd5907f142d475605d516dd7e8b9a8", size = 37155, upload-time = "2026-07-30T23:35:02.003Z" },
]

[[package]]
name = "tqdm"
version = "4.70.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "colorama", marker = "sys_platform == 'win32'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/21/3b/6c24bec5be5e743ffd99576daa5cc077722fc7d5bbc00bd133fa0c698dc6/tqdm-4.70.0.tar.gz", hash
= "sha256:55b0b0dbd97462d06ebee91e4dac24ed4d4702be82b24f07e6c1d27e08cea220", size = 795438, upload-time = "2026-07-27T11:33:15.271Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/f9/1c/01bfd571a64e7f270e6bab5e33777debe0edc56759233ce84f27dec92d14/tqdm-4.70.0-py3-none-any.whl",
 hash = "sha256:7f585706bfddbdebf89daac705b2dfcc16890130727d3197ca62c732b4310953", size = 80184, upload-time = "2026-07-27T11:33:13.167Z" },
]

[[package]]
name = "truststore"
version = "0.10.4"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/53/a3/1585216310e344e8102c22482f6060c7a6ea0322b63e026372e6dcefcfd6/truststore-0.10.4.tar.gz",
 hash = "sha256:9d91bd436463ad5e4ee4aba766628dd6cd7010cf3e2461756b3303710eebc301", size = 26169, upload-time = "2025-08-12T18:49:02.73Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/19/97/56608b2249fe206a67cd573bc93cd9896e1efb9e98bce9c163bcdc704b88/truststore-0.10.4-py3-none-any
.whl", hash = "sha256:adaeaecf1cbb5f4de3b1959b42d41f6fab57b2b1666adb59e89cb0b53361d981", size = 18660, upload-time = "2025-08-12T18:49:01.46Z" },
]

[[package]]
name = "typing-extensions"
version = "4.16.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/f6/cc/6253133b5bb138fc3306cebfbda2c520f545d36b5be2c7255cc528bb45d6/typing_extensions-4.16.0.t
ar.gz", hash = "sha256:dc983d19a509c94dba722ee6abd33940f7c05a89e243c47e907eb4db6f1a43e5", size = 113555, upload-time = "2026-07-02T08:40:05.92Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/49/d3/b8441a820a491ddfc024b0b0cf0393375b75ea13866d9c66727e54c2fc80/typing_extensions-4.16.0-py3-n
one-any.whl", hash = "sha256:481caa481374e813c1b176ada14e97f1f67a4539ce9cfeb3f350d78d6370c2e8", size = 45571, upload-time = "2026-07-02T08:40:04.659Z"
 },
]

[[package]]
name = "typing-inspect"
version = "0.9.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "mypy-extensions" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/dc/74/1789779d91f1961fa9438e9a8710cdae6bd138c80d7303996933d117264a/typing_inspect-0.9.0.tar.g
z", hash = "sha256:b23fc42ff6f6ef6954e4852c1fb512cdd18dbea03134f91f856a95ccc9461f78", size = 13825, upload-time = "2023-05-24T20:25:47.612Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/65/f3/107a22063bf27bdccf2024833d3445f4eea42b2e598abfbd46f6a63b6cb0/typing_inspect-0.9.0-py3-none-
any.whl", hash = "sha256:9ee6fc59062311ef8547596ab6b955e1b8aa46242d854bfc78f4f6b0eff35f9f", size = 8827, upload-time = "2023-05-24T20:25:45.287Z" },
]

[[package]]
name = "typing-inspection"
version = "0.4.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/55/e3/70399cb7dd41c10ac53367ae42139cf4b1ca5f36bb3dc6c9d33acdb43655/typing_inspection-0.4.2.ta
r.gz", hash = "sha256:ba561c48a67c5958007083d386c3295464928b01faa735ab8547c5692e87f464", size = 75949, upload-time = "2025-10-01T02:14:41.687Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/dc/9b/47798a6c91d8bdb567fe2698fe81e0c6b7cb7ef4d13da4114b41d239f65d/typing_inspection-0.4.2-py3-no
ne-any.whl", hash = "sha256:4ed1cacbdc298c220f1bd249ed5287caa16f34d44ef4e9c3d0cbad5b521545e7", size = 14611, upload-time = "2025-10-01T02:14:40.154Z"
```

```
},
]

[[package]]
name = "tzdata"
version = "2026.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/92/ff/5a28bdfd8c3ebec42564ac7d0e54ca3db65044a9314a97f9564fa7a1e926/tzdata-2026.3.tar.gz", has
h = "sha256:4a1518b8993086a7982523e071643f3c0e5f213e75b21318e78bcabfff9d1415", size = 198674, upload-time = "2026-07-10T08:50:37.887Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e5/6d/b53b99a9f2766d095985947a5782f1702cabb129a34f7a802d7197af832f/tzdata-2026.3-py2.py3-none-any
.whl", hash = "sha256:dc096730c87af6cab1b171c9d532be840741ff5d459015e7f6947bd7d7e54931", size = 348168, upload-time = "2026-07-10T08:50:36.46Z" },
]

[[package]]
name = "urllib3"
version = "2.7.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/53/0c/06f8b233b8fd13b9e5ee11424ef85419ba0d8ba0b3138bf360be2ff56953/urllib3-2.7.0.tar.gz", has
h = "sha256:231e0ec3b63ceb14667c67be60f2f2c40a518cb38b03af60abc813da26505f4c", size = 433602, upload-time = "2026-05-07T16:13:18.596Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/7f/3e/5db95bcf282c52709639744ca2a8b149baccf648e39c8cc87553df9eae0c/urllib3-2.7.0-py3-none-any.whl
", hash = "sha256:9fb4c81ebbb1ce9531cce37674bbc6f1360472bc18ca9a553ede278ef7276897", size = 131087, upload-time = "2026-05-07T16:13:17.151Z" },
]

[[package]]
name = "uuid-utils"
version = "0.17.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/e7/91/63938e0e7e7876658e5e40178e7c0735b53527886fe11797a11699c55edd/uuid_utils-0.17.0.tar.gz",
 hash = "sha256:abb5667a36119019b3fa320c4d10c21ebccfcc87c8a739e6a0056cee7f48dde2", size = 43220, upload-time = "2026-07-09T13:49:58.433Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/20/80/a7e685968e3cec99d6fe2fb25d0f5726310e1bba356da68c13dfd8b7d140/uuid_utils-0.17.0-cp312-cp312-
macosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:9205068badf453d2f0821fd5d340389b4679992d7ff79d4f3e5608996dd1b287",
size = 556403, upload-time = "2026-07-09T13:48:27.022Z" },
    { url = "https://files.pythonhosted.org/packages/56/47/3102d93bcb7b0bfe6bede63ff8f221a7f91348e10a37f682773be27c56d9/uuid_utils-0.17.0-cp312-cp312-
macosx_10_12_x86_64.whl", hash = "sha256:0fcca4e838af9ac9243b3358d7c14afa4dca286a87781124c272d6c4cad9c968", size = 285608, upload-time = "2026-07-09T1
3:48:28.769Z" },
    { url = "https://files.pythonhosted.org/packages/55/fb/d59695f0f8db065b93c63316eaafa05a22d75a0486978a33736c52c646d5/uuid_utils-0.17.0-cp312-cp312-
manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:0f3729e839209f3457d0d8b6a35a376fdf65577a5aecaf4cc3587d3305759ba6", size = 319926, up
load-time = "2026-07-09T13:48:29.965Z" },
    { url = "https://files.pythonhosted.org/packages/5a/03/62fabcd1e990e07a0e220e8d552af45bc16f107fa8e55c2014a706bb1a1e/uuid_utils-0.17.0-cp312-cp312-
manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:3dac0ad0cd9a2818d1775215365a4e8c2f8ada215529dd26f3f8cceeb67a6988", size = 327172, uplo
ad-time = "2026-07-09T13:48:31.187Z" },
    { url = "https://files.pythonhosted.org/packages/d9/37/a5081391338b459e2f8d8b12581f00f8caa6317fab510e0e85c18c59e938/uuid_utils-0.17.0-cp312-cp312-
manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:e671b2322ef09106ecb1ca0f4c398b134d5e2c1f80d7a4f3336847a3072c0e94", size = 439075, up
load-time = "2026-07-09T13:48:32.295Z" },
    { url = "https://files.pythonhosted.org/packages/59/30/91795bd01e17a13661280d4899fbf38fb05e3f38e873f9aaec106ec30aa0/uuid_utils-0.17.0-cp312-cp312-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:8eb3e5caca8d3a6f72ea4cce024583f989f6f2e9186f98800213fff0176e8bcc", size = 320247, uplo
ad-time = "2026-07-09T13:48:33.64Z" },
    { url = "https://files.pythonhosted.org/packages/e5/11/09102b78303e4eb62069d6d88ef9fd661dc523e8f429e1fd67eaa78a6f44/uuid_utils-0.17.0-cp312-cp312-
manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:8b72c2002202038666bf647f9a790906214c7c11cd0d6efef77b7d07bef3034a", size = 344738, upload-time
= "2026-07-09T13:48:34.786Z" },
    { url = "https://files.pythonhosted.org/packages/74/f9/be95bad6954b60328878c3800258f01a6accd24fd75112d13f023462d53f/uuid_utils-0.17.0-cp312-cp312-
musllinux_1_2_aarch64.whl", hash = "sha256:4e2ac1c0b56f2c91b6f158e29ed96b1503223fe8aa6e79b1be1dc55bd8a5131c", size = 496845, upload-time = "2026-07-09
T13:48:36.057Z" },
    { url = "https://files.pythonhosted.org/packages/2d/02/8a19a34e0530d987488a068a71576a236f5c8c746630b870b57f71eb24ef/uuid_utils-0.17.0-cp312-cp312-
musllinux_1_2_armv7l.whl", hash = "sha256:6c142bd0cb4dba31c10babe00d59f7ef6460f0ef55eaa9c1a9da270684af996a", size = 603233, upload-time = "2026-07-09T
13:48:37.512Z" },
    { url = "https://files.pythonhosted.org/packages/f4/a8/b1abab36ff73b0248d82179816467f6d39a2e80fd64329a895ca94f3508e/uuid_utils-0.17.0-cp312-cp312-
musllinux_1_2_i686.whl", hash = "sha256:e252db239eb41c32248e096e0d170bce5896a4fd3405556362bc3dd83d912206", size = 561401, upload-time = "2026-07-09T13
:48:38.977Z" },
    { url = "https://files.pythonhosted.org/packages/61/91/70e7b528b351cc03a9ca43e6116371cdde31bb12bcead7ca2ca1367366cc/uuid_utils-0.17.0-cp312-cp312-
musllinux_1_2_x86_64.whl", hash = "sha256:237722b6581bb5b4eb4cefbcbe5c6e2980a440aabe781fbe50ebf1cb71eee4cc", size = 525314, upload-time = "2026-07-09T
13:48:40.599Z" },
    { url = "https://files.pythonhosted.org/packages/d6/f6/9167e90cf9937d6558f92d022ff3024a69d938a514d9c8faa4080f73b001/uuid_utils-0.17.0-cp312-cp312-
win32.whl", hash = "sha256:46a73cacdf512f473a81f65dbf84186e08cfe6e9118fa582b6c6b33a8288a30d", size = 166831, upload-time = "2026-07-09T13:48:41.862Z"
},
    { url = "https://files.pythonhosted.org/packages/5c/7d/0b889654d9ee3413f810cf4685e241285f650d98a4103ac9f3c6bcc95f29/uuid_utils-0.17.0-cp312-cp312-
win_amd64.whl", hash = "sha256:e59b60a0a4cb7541480e02090d37dc2df3b72df4c2e776fff64ce3a4e3dd4637", size = 172944, upload-time = "2026-07-09T13:48:42.99
2Z" },
    { url = "https://files.pythonhosted.org/packages/be/35/8c6e1bf65e4d400352885dadc656ad6d0af96e89231e3f04686bc2197128/uuid_utils-0.17.0-cp312-cp312-
win_arm64.whl", hash = "sha256:d561a4c5747a1e6c7fa7c49a0292e78b4e8c456332caa084fc7abad8de828652", size = 172459, upload-time = "2026-07-09T13:48:44.27
1Z" },
    { url = "https://files.pythonhosted.org/packages/d2/dd/614fb9912157ac0128e6050859ccf06d9f13df9a944a803e8f80f6157e38/uuid_utils-0.17.0-cp313-cp313-
macosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:d11a7bc1e02da8984d32e6de9e0826c6edac00eac17de270f372bf32f9a0af63",
size = 557259, upload-time = "2026-07-09T13:48:45.664Z" },
    { url = "https://files.pythonhosted.org/packages/3e/11/d072711704de3d21bec08b6c2f36a215200ca1d5e01a390ea1ac434080a0/uuid_utils-0.17.0-cp313-cp313-
macosx_10_12_x86_64.whl", hash = "sha256:7a49f47ac26df3e431c56b825c1bae8e6d3d591fdbb7438c227cc9845a7e3d73", size = 286271, upload-time = "2026-07-09T1
3:48:47.018Z" },
    { url = "https://files.pythonhosted.org/packages/18/6d/8a63e5eb2d5a6ba69a6c2036e305075bd6f5a022e7ea25fc6ce0eb7c51d2/uuid_utils-0.17.0-cp313-cp313-
manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:32df1944808877702ceea398c103881c09a679bb672a215e01c2a84231266bf9", size = 320025, up
load-time = "2026-07-09T13:48:48.208Z" },
    { url = "https://files.pythonhosted.org/packages/f7/2d/bdc2caf9719d9090d7c46043242ae6136cba4f7a7ee384992ab905ad9aa1/uuid_utils-0.17.0-cp313-cp313-
manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:98c88d3edd08e7245562e9815996dbc6f0bd4745e1c76462f24af5ae4e187dd1", size = 327931, uplo
ad-time = "2026-07-09T13:48:49.673Z" },
    { url = "https://files.pythonhosted.org/packages/b6/33/9219d09d51ead282b578b2a4e0a515c2cce3ec52076cada8bfb7e35727d5/uuid_utils-0.17.0-cp313-cp313-
manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:5a4370089c8b2e42f1db51d76408c7fa8eaa2934bf854d17983d16179c07c098", size = 438537, up
load-time = "2026-07-09T13:48:50.842Z" },
    { url = "https://files.pythonhosted.org/packages/d8/79/e8e0f8b3955f2081c116157119d87659937893242eb834aa170da04d660b/uuid_utils-0.17.0-cp313-cp313-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:09a55b7a5ae764985cb46467496a1787678d0a1400356157a080ad95b1a36869", size = 320656, uplo
ad-time = "2026-07-09T13:48:52.164Z" },
    { url = "https://files.pythonhosted.org/packages/d5/5e/d1ceddc430ff04b6e21704b2030d4438074a2f478b265dab43da957791c1/uuid_utils-0.17.0-cp313-cp313-
manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:56aa6488b931246fae11924e4bd0e2b32677e63945eecb71c29e3c2ca0dc3131", size = 345310, upload-time
= "2026-07-09T13:48:54.076Z" },
    { url = "https://files.pythonhosted.org/packages/d5/62/89438e12f389a843e626b7e37691319a057b3d6b80914609106891faadda/uuid_utils-0.17.0-cp313-cp313-
musllinux_1_2_aarch64.whl", hash = "sha256:309a35f12d99dde19032bc2259cda6431c85eeac0879134dc777cc3087d7e1cb", size = 496771, upload-time = "2026-07-09
T13:48:55.365Z" },
    { url = "https://files.pythonhosted.org/packages/87/d2/eedcd99f522d60e238ead03844f0d51743ba84d33044959e230b756bf212/uuid_utils-0.17.0-cp313-cp313-
musllinux_1_2_armv7l.whl", hash = "sha256:21c79b61ff750abcf057163dd764ccb6196cde7a26cda1b31b45cd97769e03b3", size = 603631, upload-time = "2026-07-09T
13:48:56.746Z" },
    { url = "https://files.pythonhosted.org/packages/0e/a8/bb1b38aaddd7243b6e562c6694f499bf094800918316192fd8cb2cdc2620/uuid_utils-0.17.0-cp313-cp313-
musllinux_1_2_i686.whl", hash = "sha256:4134353bfe3026ddab8e886002dc52bc5a0ab04611aabb0eaae23c32e6e57f64", size = 562008, upload-time = "2026-07-09T13
:48:58.241Z" },
    { url = "https://files.pythonhosted.org/packages/b4/77/5f7ed930dc105e293845c09e4d5bd84076318a12f45a46783e1af64906d7/uuid_utils-0.17.0-cp313-cp313-
```

musllinux_1_2_x86_64.whl", hash = "sha256:7c89359affecebe2e39e6a116d069b363c936511a9572b308402489a26957d89", size = 525527, upload-time = "2026-07-09T13:48:59.784Z" },
    { url = "https://files.pythonhosted.org/packages/fd/25/1b55697adf6811a6f92cff6340e6b03e31fd6bc51066a5c10698c29b3679/uuid_utils-0.17.0-cp313-cp313-pyemscripten_2025_0_wasm32.whl", hash = "sha256:6a019a31bc4db89a0903a3e4f6b218571f3a6ff0ad4b3d3fe1c8f91a05ff6e3e", size = 97965, upload-time = "2026-07-09T13:49:01.217Z" },
    { url = "https://files.pythonhosted.org/packages/26/bf/cd729343de4684230be8a966bad7bfc2cf10ce3e643b1189a8b5370dbe35/uuid_utils-0.17.0-cp313-cp313-win32.whl", hash = "sha256:b3131a82d0c7611f0aa480a6d36929e001a3f54ba0fc029a8118a5863cce513c", size = 167316, upload-time = "2026-07-09T13:49:02.354Z" },
    { url = "https://files.pythonhosted.org/packages/76/f0/e602ae0a1b139a7826e5189b93d91902564def06d5006324fd2faf82c8fc/uuid_utils-0.17.0-cp313-cp313-win_amd64.whl", hash = "sha256:9e311f908d2f842fca4c7dcebc4f10306b8089b204ef04cf6704b4332c9ff6ff", size = 173630, upload-time = "2026-07-09T13:49:03.529Z" },
    { url = "https://files.pythonhosted.org/packages/1a/52/024ebece265b387154115dc4f1d9727174ef82623069f4bec8b7ed7e73f7/uuid_utils-0.17.0-cp313-cp313-win_arm64.whl", hash = "sha256:c351737e2e65497c7200ab4ffb8af97e9f48be6488309abdd265fe08d66ee92f", size = 173214, upload-time = "2026-07-09T13:49:04.836Z" },
    { url = "https://files.pythonhosted.org/packages/56/44/e2fd3fdf356e1b55d2acf1b956b4f3f29ffb215a99c387eba04b1c5fba66/uuid_utils-0.17.0-cp314-cp314-macosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:673d89cc434cc9b97a0b4cf61272f6fca70a81f64eb0afbface2a0d9f77f06cd", size = 562232, upload-time = "2026-07-09T13:49:06.201Z" },
    { url = "https://files.pythonhosted.org/packages/19/28/65e0980d668a6d44e699f59d1acf43d6b5d4893592c115ce7c680bb4dfa1/uuid_utils-0.17.0-cp314-cp314-macosx_10_12_x86_64.whl", hash = "sha256:387cf7437c94ddec08651a0f1081381299c7075bc48a6251d8922bf39973378a", size = 287858, upload-time = "2026-07-09T13:49:07.452Z" },
    { url = "https://files.pythonhosted.org/packages/8f/8d/5e97bcebc90fb6a10f98af3dc1ba552e04183aba59e2edc0b9cf486dd998/uuid_utils-0.17.0-cp314-cp314-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:220b52746d99e11964badac3c0869016e0c24bafb70a7dd5c2c072a6be3da9cc", size = 321587, upload-time = "2026-07-09T13:49:09.489Z" },
    { url = "https://files.pythonhosted.org/packages/8b/d7/88b2a2370cc3d455ba0515fb6f5c8f7ac0c0f55a86801b6e56a432f22c17/uuid_utils-0.17.0-cp314-cp314-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:0ab4a66e7a035ad6625cfc1fbdb34f5c2d25a80ae1ef4bfee458ea2036333c6d", size = 328964, upload-time = "2026-07-09T13:49:11.292Z" },
    { url = "https://files.pythonhosted.org/packages/bd/0f/181c5da673953dfc0958cb4fb3a4984a9098673ddb05cac68e994bc8511b/uuid_utils-0.17.0-cp314-cp314-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:5641071337eb11d61a001ea08793bf72216f3241f0a433ed2764804b2a3e3cc7", size = 442909, upload-time = "2026-07-09T13:49:12.644Z" },
    { url = "https://files.pythonhosted.org/packages/ec/38/5c5e665af542884a8fd3c61725c38453239e13940326b5b70f3ef8881a97/uuid_utils-0.17.0-cp314-cp314-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:9082e709014946b1f6e96ae6ecd93652efca2d2a6a3ab67dbe151c8b4bf193a4", size = 323076, upload-time = "2026-07-09T13:49:13.897Z" },
    { url = "https://files.pythonhosted.org/packages/f5/35/7de97de18cbf226c2a4f2104ad15e56ca4491717c81c0b71795c0c585b4e/uuid_utils-0.17.0-cp314-cp314-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:1fd6f0e8a162dc0e9255b6aebe3cd175e76c33202f1bf39da9e6294b93db0099", size = 347360, upload-time = "2026-07-09T13:49:15.237Z" },
    { url = "https://files.pythonhosted.org/packages/26/a1/9915d5dd59fdd1957ded5d188c0ea0b9db5a1d84d42c8d8828a7b83b366e/uuid_utils-0.17.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:d63010803d7c368963bbe6f7ec379593e76dd581d7db0f29118d88713c9e0354", size = 499267, upload-time = "2026-07-09T13:49:16.774Z" },
    { url = "https://files.pythonhosted.org/packages/c0/05/88108405262ec850cea0f95733445d6873e5772af3292baabd9ef8457740/uuid_utils-0.17.0-cp314-cp314-musllinux_1_2_armv7l.whl", hash = "sha256:a46bedc273b6f58f11dee816ff74999625ef8d007890f411b7a4975bf1c89330", size = 604940, upload-time = "2026-07-09T13:49:18.147Z" },
    { url = "https://files.pythonhosted.org/packages/89/d5/6dbcd300de47cc443cff2656cd5327a385751213dcb2101cfee7388170b2/uuid_utils-0.17.0-cp314-cp314-musllinux_1_2_i686.whl", hash = "sha256:405233a5f625b3d995648f4647fa6befa4567cf3f74e1f6b9837e16f7310f0e0", size = 564172, upload-time = "2026-07-09T13:49:19.593Z" },
    { url = "https://files.pythonhosted.org/packages/ab/94/e8057f2288a415fba8a978bca4b589f5cb6b91a028a5dc07a1775938b33f/uuid_utils-0.17.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:b6c5d2d71e1f17329150ad9427d27f4a3f29a01792e7ecdc64a98ac5368fc4d5", size = 528533, upload-time = "2026-07-09T13:49:21.075Z" },
    { url = "https://files.pythonhosted.org/packages/f0/6b/31713148c77e48e62f51aa042a98a54a8be0396912ea5130f83f52ae722d/uuid_utils-0.17.0-cp314-cp314-pyemscripten_2026_0_wasm32.whl", hash = "sha256:f7e9b8728ba07a3cb2f29d5aa1a266c2664eb8ef0fd43afa34627c92f7fac8f0", size = 99197, upload-time = "2026-07-09T13:49:22.351Z" },
    { url = "https://files.pythonhosted.org/packages/f3/f3/ca6f6ac5428312df8ed632f6dd9f9e6aba23090471fcdeae53eab027e8b3/uuid_utils-0.17.0-cp314-cp314-win32.whl", hash = "sha256:58838921e377791ef22c64cc92141bfae030f43651ff9272f0f28a208a9e6a5a", size = 169540, upload-time = "2026-07-09T13:49:23.563Z" },
    { url = "https://files.pythonhosted.org/packages/c6/cd/7ede0db66411fa09817d79b680f7454ea9bee2d374e1922e4efd065760a3/uuid_utils-0.17.0-cp314-cp314-win_amd64.whl", hash = "sha256:42275ebd0e8e74e32cdbfb8bd88fc99576567d51d54a508020611fd8f4f463a0", size = 175984, upload-time = "2026-07-09T13:49:24.703Z" },
    { url = "https://files.pythonhosted.org/packages/f0/81/533b5f80cd4918c0693f4e1b7b90ceb1caa45f4266ae8b528135d7ecca5d/uuid_utils-0.17.0-cp314-cp314-win_arm64.whl", hash = "sha256:b5d11cccba076a32321ef1380dea956821f0b51794ef59df64e58fb1cd543aae", size = 174749, upload-time = "2026-07-09T13:49:25.886Z" },
    { url = "https://files.pythonhosted.org/packages/a0/13/f400ac39d06fd8be5b099c09e41bb975205926722a3e8d53348817cb7ff9/uuid_utils-0.17.0-cp314-cp314t-macosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:fae8b282f0cb22a5de222999f7723f4e5ec04f6fcdf4aaef879b5b36625ae2b0", size = 562610, upload-time = "2026-07-09T13:49:27.374Z" },
    { url = "https://files.pythonhosted.org/packages/03/8c/c71c8312304c56f6d0bcba87cd402fa79bec35d18ffc8c41954196ca68e5/uuid_utils-0.17.0-cp314-cp314t-macosx_10_12_x86_64.whl", hash = "sha256:967955620df45e6cffe2e9950cb9903cb455649396f896b26b04363a91a5054b", size = 289473, upload-time = "2026-07-09T13:49:28.989Z" },
    { url = "https://files.pythonhosted.org/packages/bb/cd/522117e2e5184ca1d4f0f85ee833e9e21bd8c6b99eff8a4d1a8e5a194e33/uuid_utils-0.17.0-cp314-cp314t-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:375cde148430d60a4a07c03abaa0774c4fddfdd90de99b4ba02f24088bc9d750", size = 321600, upload-time = "2026-07-09T13:49:30.4Z" },
    { url = "https://files.pythonhosted.org/packages/6d/f4/0d81f9bd346fc717bc561c08fa6457e0328966eb76e536b938fe77d56459/uuid_utils-0.17.0-cp314-cp314t-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:975c17da26c5b9d46c336b03c52a057ac28378d6f9d98b58d32a038589bb3912", size = 329569, upload-time = "2026-07-09T13:49:31.732Z" },
    { url = "https://files.pythonhosted.org/packages/5e/41/26e1363f36a94c9e8ec2dd21d5f63088d3e7c723adbb12dcc8fdc77be417/uuid_utils-0.17.0-cp314-cp314t-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:3150d836290c88f1d26eb59c4db280d87417dd3bfaadd2889c77416c8f0ff6fa", size = 442051, upload-time = "2026-07-09T13:49:33.024Z" },
    { url = "https://files.pythonhosted.org/packages/2b/a7/2c1ed1b34d7df7fdcc11c28fd26d94d44843b37d9af2435ff9fd8abdbc08/uuid_utils-0.17.0-cp314-cp314t-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:9472a8de37faf8bd216c628e0e68c8f6bef730d3ba0a5060f3b0fa460c992ac2", size = 324372, upload-time = "2026-07-09T13:49:34.554Z" },
    { url = "https://files.pythonhosted.org/packages/78/bf/328d3c6bb22c496944a1b3b732207d71aa6964eb604e5e3b9dcb91ed0a00/uuid_utils-0.17.0-cp314-cp314t-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:d27c531edb8d1f38ca2eddaa1fa24913a460aeb721f2efd4ef42a124ce94e354", size = 348548, upload-time = "2026-07-09T13:49:35.898Z" },
    { url = "https://files.pythonhosted.org/packages/3e/76/a07de5cb7b90582fdbbc830fd19be129cbbb9897cfe239fef469d7bd2d09/uuid_utils-0.17.0-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:5670c52a438e21483ce715776144914a4e2a2a5c62d9dee15f8a3e90cf128ae6", size = 498985, upload-time = "2026-07-09T13:49:37.142Z" },
    { url = "https://files.pythonhosted.org/packages/f4/62/9966e46ae34fcec6b06119631fb3c09705ea78835035ce3a82d3348eb61a/uuid_utils-0.17.0-cp314-cp314t-musllinux_1_2_armv7l.whl", hash = "sha256:6f29689a76fe7a49cbd629a794d0ec1eab48814e323a00a146a741b0195bde68", size = 605183, upload-time = "2026-07-09T13:49:38.648Z" },
    { url = "https://files.pythonhosted.org/packages/d7/4e/bb962ba0fe31e903b199f22cf4c1a6cba35a8987aef526d287277ab8ca8b/uuid_utils-0.17.0-cp314-cp314t-musllinux_1_2_i686.whl", hash = "sha256:4441600447d340ae103a353f01dbcd22ff680e5ee1a22988efe8d7b791d8fdb3", size = 565412, upload-time = "2026-07-09T13:49:40.115Z" },
    { url = "https://files.pythonhosted.org/packages/ce/9e/122adfeeeae8a84ccfd43bce627b104d12a2180a93bffd2c0e1b54dad7a6/uuid_utils-0.17.0-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:e7b04935a79c03c41ad08d0a5f390aac968bfb561f1268897bc5b0f077971efd", size = 529885, upload-time = "2026-07-09T13:49:41.513Z" },
    { url = "https://files.pythonhosted.org/packages/b3/4f/257304dded339dc35fc9bf35722ac68fd4fdb930f255b8f7bccdf74ebba9/uuid_utils-0.17.0-cp314-cp314t-win32.whl", hash = "sha256:239d8a281fe10bae33205b5d43185834d556b18434e0a113b5dc1dfb2fd97e91", size = 169472, upload-time = "2026-07-09T13:49:42.871Z" },
    { url = "https://files.pythonhosted.org/packages/35/c8/e78c06db7e9ce317ce7b8759ff2058333eac75caa8c22b75f0059589c9be/uuid_utils-0.17.0-cp314-cp314t-win_amd64.whl", hash = "sha256:e288a06cbbbcd01b44386e767985c9e21d2ad9bf59829aa7058d9a2a494804ab", size = 176271, upload-time = "2026-07-09T13:49:44.105Z" },
    { url = "https://files.pythonhosted.org/packages/a7/11/bd1c70e1ad3301163cebe66c8d26de26e6814d52f642a849448bd2833626/uuid_utils-0.17.0-cp314-cp314t-win_arm64.whl", hash = "sha256:1776a80d16369999b21627028cc5dbce819be83e1e079fdd7a51b587d2916db9", size = 175004, upload-time = "2026-07-09T13:49:45.591Z" },
]

[[package]]
name = "uvicorn"

```
version = "0.52.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "click" },
    { name = "h11" },
]
sdist = { url = "https://files.pythonhosted.org/packages/03/18/ccce41535dee1be77735592bd19965f3972c82e07ee703d324709496b716/uvicorn-0.52.1.tar.gz", hash = "sha256:112ec661814189acbccd3f7b86460147cc065fc92c0821afa78918780e4354dd", size = 100571, upload-time = "2026-08-01T18:19:30.732Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/c7/d5/68e6e9bca63c0badf67002890a46d3784c958de45b65e1275ec583ca1f06/uvicorn-0.52.1-py3-none-any.whl", hash = "sha256:e4403f9d93188cf9d1088e9f40e3acd12630e2df8675316704379a7fc20fff6a", size = 79859, upload-time = "2026-08-01T18:19:29.294Z" },
]

[package.optional-dependencies]
standard = [
    { name = "httptools" },
    { name = "python-dotenv" },
    { name = "pyyaml" },
    { name = "uvloop", marker = "platform_python_implementation != 'PyPy' and sys_platform != 'cygwin' and sys_platform != 'win32'" },
    { name = "watchfiles" },
    { name = "websockets" },
]

[[package]]
name = "uvloop"
version = "0.22.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/06/f0/18d39dbd1971d6d62c4629cc7fa67f74821b0dc1f5a77af43719de7936a7/uvloop-0.22.1.tar.gz", hash = "sha256:6c84bae345b9147082b17371e3dd5d42775bddce91f885499017f4607fdaf39f", size = 2443250, upload-time = "2025-10-16T22:17:19.342Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/3d/ff/7f72e8170be527b4977b033239a83a68d5c881cc4775fca255c677f7ac5d/uvloop-0.22.1-cp312-cp312-macosx_10_13_universal2.whl", hash = "sha256:fe94b4564e865d968414598eea1a6de60adba0c040ba4ed05ac1300de402cd42", size = 1359936, upload-time = "2025-10-16T22:16:29.436Z" },
    { url = "https://files.pythonhosted.org/packages/c3/c6/e5d433f88fd54d81ef4be58b2b7b0cea13c442454a1db703a1eea0db1a59/uvloop-0.22.1-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:51eb9bd88391483410daad430813d982010f9c9c89512321f5b60e2cddbdddd6", size = 752769, upload-time = "2025-10-16T22:16:30.493Z" },
    { url = "https://files.pythonhosted.org/packages/24/68/a6ac446820273e71aa762fa21cdcc09861edd3536ff47c5cd3b7afb10eeb/uvloop-0.22.1-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:700e674a166ca5778255e0e1dc4e9d79ab2acc57b9171b79e65feba7184b3370", size = 4317413, upload-time = "2025-10-16T22:16:31.644Z" },
    { url = "https://files.pythonhosted.org/packages/5f/6f/e62b4dfc7ad6518e7eff2516f680d02a0f6eb62c0c212e152ca708a0085e/uvloop-0.22.1-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:7b5b1ac819a3f946d3b2ee07f09149578ae76066d70b44df3fa990add49a82e4", size = 4426307, upload-time = "2025-10-16T22:16:32.917Z" },
    { url = "https://files.pythonhosted.org/packages/90/60/97362554ac21e20e81bcef1150cb2a7e4ffdaf8ea1e5b2e8bf7a053caa18/uvloop-0.22.1-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:e047cc068570bac9866237739607d1313b9253c3051ad84738cbb095be0537b2", size = 4131970, upload-time = "2025-10-16T22:16:34.015Z" },
    { url = "https://files.pythonhosted.org/packages/99/39/6b3f7d234ba3964c428a6e40006340f53ba37993f46ed6e111c6e9141d18/uvloop-0.22.1-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:512fec6815e2dd45161054592441ef76c830eddaad55c8aa30952e6fe1ed07c0", size = 4296343, upload-time = "2025-10-16T22:16:35.149Z" },
    { url = "https://files.pythonhosted.org/packages/89/8c/182a2a593195bfd39842ea68ebc084e20c850806117213f5a299dfc513d9/uvloop-0.22.1-cp313-cp313-macosx_10_13_universal2.whl", hash = "sha256:561577354eb94200d75aca23fbde86ee11be36b00e52a4eaf8f50fb0c86b7705", size = 1358611, upload-time = "2025-10-16T22:16:36.833Z" },
    { url = "https://files.pythonhosted.org/packages/d2/14/e301ee96a6dc95224b6f1162cd3312f6d1217be3907b79173b06785f2fe7/uvloop-0.22.1-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:1cdf5192ab3e674ca26da2eada35b288d2fa49fdd0f357a19f0e7c4e7d5077c8", size = 751811, upload-time = "2025-10-16T22:16:38.275Z" },
    { url = "https://files.pythonhosted.org/packages/b7/02/654426ce265ac19e2980bfd9ea6590ca96a56f10c76e63801a2df01c0486/uvloop-0.22.1-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:6e2ea3d6190a2968f4a14a23019d3b16870dd2190cd69c8180f7c632d21de68d", size = 4288562, upload-time = "2025-10-16T22:16:39.375Z" },
    { url = "https://files.pythonhosted.org/packages/15/c0/0be24758891ef825f2065cd5db8741aaddabe3e248ee6acc5e8a80f04005/uvloop-0.22.1-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:0530a5fbad9c9e4ee3f2b33b148c6a64d47bbad8000ea63704fa8260f4cf728e", size = 4366890, upload-time = "2025-10-16T22:16:40.547Z" },
    { url = "https://files.pythonhosted.org/packages/d2/53/8369e5219a5855869bcee5f4d317f6da0e2c669aecf0ef7d371e3d084449/uvloop-0.22.1-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:bc5ef13bbc10b5335792360623cc378d52d7e62c2de64660616478c32cd0598e", size = 4119472, upload-time = "2025-10-16T22:16:41.694Z" },
    { url = "https://files.pythonhosted.org/packages/f8/ba/d69adbe699b768f6b29a5eec7b47dd610bd17a69de51b251126a801369ea/uvloop-0.22.1-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:1f38ec5e3f18c8a10ded09742f7fb8de0108796eb673f30ce7762ce1b8550cad", size = 4239051, upload-time = "2025-10-16T22:16:43.224Z" },
    { url = "https://files.pythonhosted.org/packages/90/cd/b62bdeaa429758aee8de8b00ac0dd26593a9de93d302bff3d21439e9791d/uvloop-0.22.1-cp314-cp314-macosx_10_13_universal2.whl", hash = "sha256:3879b88423ec7e97cd4eba2a443aa26ed4e59b45e6b76aabf13fe2f27023a142", size = 1362067, upload-time = "2025-10-16T22:16:44.503Z" },
    { url = "https://files.pythonhosted.org/packages/0d/f8/a132124dfda0777e489ca86732e85e69afcd1ff7686647000050ba670689/uvloop-0.22.1-cp314-cp314-macosx_10_13_x86_64.whl", hash = "sha256:4baa86acedf1d62115c1dc6ad1e17134476688f08c6efd8a2ab076e815665c74", size = 752423, upload-time = "2025-10-16T22:16:45.968Z" },
    { url = "https://files.pythonhosted.org/packages/a3/94/94af78c156f88da4b3a733773ad5ba0b164393e357cc4bd0ab2e2677a7d6/uvloop-0.22.1-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:297c27d8003520596236bdb2335e6b3f649480bd09e00d1e3a99144b691d2a35", size = 4272437, upload-time = "2025-10-16T22:16:47.451Z" },
    { url = "https://files.pythonhosted.org/packages/b5/35/60249e9fd07b32c665192cec7af29e06c7cd96fa1d08b84f012a56a0b38e/uvloop-0.22.1-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:c1955d5a1dd43198244d47664a5858082a3239766a839b2102a269aaff7a4e25", size = 4292101, upload-time = "2025-10-16T22:16:49.318Z" },
    { url = "https://files.pythonhosted.org/packages/02/62/67d382dfcb25d0a98ce73c11ed1a6fba5037a1a1d533dcbb7cab033a2636/uvloop-0.22.1-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:b31dc2fccbd42adc73bc4e7cdbae4fc5086cf378979e53ca5d0301838c5682c6", size = 4114158, upload-time = "2025-10-16T22:16:50.517Z" },
    { url = "https://files.pythonhosted.org/packages/f0/7a/f1171b4a882a5d13c8b7576f348acfe6074d72eaf52cccef752f748d4a9f/uvloop-0.22.1-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:93f617675b2d03af4e72a5333ef89450dfaa5321303ede6e67ba9c9d26878079", size = 4177360, upload-time = "2025-10-16T22:16:52.646Z" },
    { url = "https://files.pythonhosted.org/packages/79/7b/b01414f31546caf0919da80ad57cbfe24c56b151d12af68cee1b04922ca8/uvloop-0.22.1-cp314-cp314t-macosx_10_13_universal2.whl", hash = "sha256:37554f70528f60cad66945b885eb01f1bb514f132d92b6eeed1c90fd54ed6289", size = 1454790, upload-time = "2025-10-16T22:16:54.355Z" },
    { url = "https://files.pythonhosted.org/packages/d4/31/0bb232318dd838cad3fa8fb0c68c8b40e1145b32025581975e18b11fab40/uvloop-0.22.1-cp314-cp314t-macosx_10_13_x86_64.whl", hash = "sha256:b76324e2dc033a0b2f435f33eb88ff9913c156ef78e153fb210e03c13da746b3", size = 796783, upload-time = "2025-10-16T22:16:55.906Z" },
    { url = "https://files.pythonhosted.org/packages/42/38/c9b09f3271a7a723a5de69f8e237ab8e7803183131bc57c890db0b6bb872/uvloop-0.22.1-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:badb4d8e58ee08dad957002027830d5c3b06aea446a6a3744483c2b3b745345c", size = 4647548, upload-time = "2025-10-16T22:16:57.008Z" },
    { url = "https://files.pythonhosted.org/packages/c1/37/945b4ca0ac27e3dc4952642d4c900edd030b3da6c9634875af6e13ae80e5/uvloop-0.22.1-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:b91328c72635f6f9e0282e4a57da7470c7350ab1c9f48546c0f2866205349d21", size = 4467065, upload-time = "2025-10-16T22:16:58.206Z" },
    { url = "https://files.pythonhosted.org/packages/97/cc/48d232f33d60e2e2e0b42f4e73455b146b76ebe216487e862700457fbf3c/uvloop-0.22.1-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:daf620c2995d193449393d6c62131b3fbd40a63bf7b307a1527856ace637fe88", size = 4328384, upload-time = "2025-10-16T22:16:59.36Z" },
    { url = "https://files.pythonhosted.org/packages/e4/16/c1fd27e9549f3c4baf1dc9c20c456cd2f822dbf8de9f463824b0c0357e06/uvloop-0.22.1-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:6cde23eeda1a25c75b2e07d39970f3374105d5eafbaab2a4482be82f272d5a5e", size = 4296730, upload-time = "2025-10-16T22:17:00.744Z" },
]

[[package]]
```

```
name = "watchfiles"
version = "1.2.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
]
sdist = { url = "https://files.pythonhosted.org/packages/cd/41/5e1a4bb12aac5f1493fa1bdc11154eca3b258ca4eba65d39c473fe19d8e9/watchfiles-1.2.0.tar.gz",
hash = "sha256:c995fba777f1ea992f090f9236e9284cf7a5d1a0130dd5a3d82c598cacd76838", size = 108252, upload-time = "2026-05-18T04:32:04.251Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/b8/2f/e42c992d2afda3108ea1c02acecc991b9f31d05c14adc2a7cee9ee211fc4/watchfiles-1.2.0-cp312-cp312-m
acosx_10_12_x86_64.whl", hash = "sha256:bc13eb17538be00c874699dc0abe4ee2bc8d50bb1166a6b9e175ef3fd7eb8f26", size = 400115, upload-time = "2026-05-18T04
:32:02.06Z" },
    { url = "https://files.pythonhosted.org/packages/5f/8f/6af2ea19065c91d8b0ea3516fdfc8c0d349f407e8e9fbf4e5a17360de8ad/watchfiles-1.2.0-cp312-cp312-m
acosx_11_0_arm64.whl", hash = "sha256:2d95ddc1eb6914154253d239089900813f6a767e174b8e6a50e7fdacb7e4236c", size = 393659, upload-time = "2026-05-18T04:3
0:50.951Z" },
    { url = "https://files.pythonhosted.org/packages/13/01/b32a967c56fb3e3e5be3db52c3d3b87fa4513aa367d8ed1ad96d42952e5f/watchfiles-1.2.0-cp312-cp312-m
anylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:8f70d8b291ef6e88d19b1f297a6905ddb978888d9272b0d05e6f53309856bcfc", size = 453207, upl
oad-time = "2026-05-18T04:31:04.231Z" },
    { url = "https://files.pythonhosted.org/packages/04/98/97557a812180338cb1abd32e1cffcc4588f59b5f23e0cb006b2ba95ba64a/watchfiles-1.2.0-cp312-cp312-m
anylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:56d8641cf834c2836922899105bd3ce3d0dfc69291d52edf0b4d0436829b34c0", size = 459273, uploa
d-time = "2026-05-18T04:31:50.377Z" },
    { url = "https://files.pythonhosted.org/packages/e8/a8/b4b08dcb7653b8087c6586f7ce649505900e866bbcfe40dc9587af02e686/watchfiles-1.2.0-cp312-cp312-m
anylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:2581a94056e55d7d0a31a823ea92bf73749c489ca2285bfdc0fbe6b2bb49d50c", size = 489927, upload-ti
me = "2026-05-18T04:31:42.485Z" },
    { url = "https://files.pythonhosted.org/packages/50/94/3dceea03545d2e5ddfd839f0ddd5e1cecbf1697b5a428d5ba11cef6af95d/watchfiles-1.2.0-cp312-cp312-m
anylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:41bc1199f7523b3f82843c88cbb979180c949caef0342cf90968f178e5d49b01", size = 570476, upl
oad-time = "2026-05-18T04:31:03.071Z" },
    { url = "https://files.pythonhosted.org/packages/cc/f2/d39a5450c3532092b91f81d274360e613c2371bc874a89c7a1a3c5e8d138/watchfiles-1.2.0-cp312-cp312-m
anylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:7571e4464cb6e434958f867f7f730b8ab0b75e3f8e5eac0499168486ab3c33a8", size = 465650, upload-
time = "2026-05-18T04:30:12.701Z" },
    { url = "https://files.pythonhosted.org/packages/22/24/ed72f68cbc1333ca9b9f2200aa048bb6658ae41709bc1caad4310f4bdffd/watchfiles-1.2.0-cp312-cp312-m
anylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:e53a384f76b631c3ae5334ce6a52f0baa3a911eb94a4eac7f160079868b716d5", size = 456398, uploa
d-time = "2026-05-18T04:30:13.784Z" },
    { url = "https://files.pythonhosted.org/packages/0d/64/982ef4a4e5bab5b6e5b6becc8cd5e732f6130a78b855f0abec6439a9a135/watchfiles-1.2.0-cp312-cp312-m
anylinux_2_31_riscv64.whl", hash = "sha256:d20029a60a71a052a24c4db7673bc4de39ab89adbaccbfb5d67987c5d73f424d", size = 465140, upload-time = "2026-05-18
T04:31:52.111Z" },
    { url = "https://files.pythonhosted.org/packages/a0/0c/95282abf4ed680b6096010bcfc30c5fa7a041fc5aa5a2ad17a2cc6c75bba/watchfiles-1.2.0-cp312-cp312-m
usllinux_1_1_aarch64.whl", hash = "sha256:2cb93af48550faf1cea04c303107c8b75833de7013e57ce27d3b8d21d8d0f58c", size = 630259, upload-time = "2026-05-18T
04:31:25.676Z" },
    { url = "https://files.pythonhosted.org/packages/30/45/607c1de1530c4bdcf2cf1d1ecc2505ddba5d96bd43ba9f2b0e79876f850f/watchfiles-1.2.0-cp312-cp312-m
usllinux_1_1_x86_64.whl", hash = "sha256:2995c176de7692b86a2e4c58d9ec718f753150a979cb4a754e2b4ffa38e70906", size = 659859, upload-time = "2026-05-18T0
4:30:24.333Z" },
    { url = "https://files.pythonhosted.org/packages/fa/08/d9e2e0f9e8e6791d33aefc694ad7eefa7f901f63caff84a81ded38692f9c/watchfiles-1.2.0-cp312-cp312-w
in32.whl", hash = "sha256:7a2cffd17d27d2ecbb310c2b1d8174f222a5495b1a721894afa88ec11e25b898", size = 275480, upload-time = "2026-05-18T04:30:31.307Z" }
,
    { url = "https://files.pythonhosted.org/packages/1c/e6/9d42569c0102645cc8cea5d8c7d8a1e9d4ada2cb7f05f75e554b8aa2202a/watchfiles-1.2.0-cp312-cp312-w
in_amd64.whl", hash = "sha256:f155b3a1b2a5fc89cdc70d47ee5d54e3b75e88efa34982028a35daef9ba00379", size = 288718, upload-time = "2026-05-18T04:32:10.745
Z" },
    { url = "https://files.pythonhosted.org/packages/0a/26/88e0dc6ee3898169d7fa22bb6a69cabf2502d2ee25cb8c876d1262d204f8/watchfiles-1.2.0-cp312-cp312-w
in_arm64.whl", hash = "sha256:8fa585ede612ee9f9e91b18bebf9ba11b9ae29a4e3a0d0cf6fca3e382133f0d5", size = 281026, upload-time = "2026-05-18T04:30:22.23Z
" },
    { url = "https://files.pythonhosted.org/packages/d1/4d/70a7feced9f87e2ff26dba42667290f41694fc64646c67261fbb8cab5d5c/watchfiles-1.2.0-cp313-cp313-m
acosx_10_12_x86_64.whl", hash = "sha256:01ea8d66f0693b9b60a6541c8d10263091ca9a9060d242f3c1f3143f9aad2c98", size = 399730, upload-time = "2026-05-18T04
:31:38.162Z" },
    { url = "https://files.pythonhosted.org/packages/31/3a/0da302f2307aee316922806ebd5726c542cbd787c938271cf14a074c7daf/watchfiles-1.2.0-cp313-cp313-m
acosx_11_0_arm64.whl", hash = "sha256:7ba0480b9a74af058f43b337e937a451e109295c420916d68ad24e3dc02f5e44", size = 392842, upload-time = "2026-05-18T04:3
0:27.051Z" },
    { url = "https://files.pythonhosted.org/packages/db/ef/d5bdb705c224dbc256aa0c1ec47bf4e61ec52558f2afb44a71a1fe4d7015/watchfiles-1.2.0-cp313-cp313-m
anylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:4f34e26a19f91f710c08e0183429f0d1d15df734e6bc78c31e77b9ea9c433658", size = 452989, upl
oad-time = "2026-05-18T04:31:11.945Z" },
    { url = "https://files.pythonhosted.org/packages/71/29/5495f2c1661949ef7a35e4d71111d129cfe7606414a26887a919d0a55406/watchfiles-1.2.0-cp313-cp313-m
anylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:b4e77f6a55f858504069abd35d336a637555c09bca453dde1ee1e5ada8a6a1fb", size = 458978, uploa
d-time = "2026-05-18T04:30:52.606Z" },
    { url = "https://files.pythonhosted.org/packages/d5/8c/7f9c07c433811c2fffd93e13fdfb7135de9aab5f2ae41be08960fa0047dc/watchfiles-1.2.0-cp313-cp313-m
anylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:0cb4d80e212f116474a545c21c912b445f16bb0cef9e6a73a498164223e14e2f", size = 490248, upload-ti
me = "2026-05-18T04:31:36.003Z" },
    { url = "https://files.pythonhosted.org/packages/3c/11/d93632febc52fbc21be90231bb7c17fd5387f46c9076fd40a5f9c2ae6910/watchfiles-1.2.0-cp313-cp313-m
anylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:b974946a10af379d425e2eef5b62f5c6ebeaccf91d45eaad6f5b27ecd4f91aa0", size = 571847, upl
oad-time = "2026-05-18T04:31:10.862Z" },
    { url = "https://files.pythonhosted.org/packages/55/b4/383173e73aabb07ad1d9c7aa859d95437ac46a6d6a1e11005facda0c9d19/watchfiles-1.2.0-cp313-cp313-m
anylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:86bc13c25a8d1fcd70b51d0ce7c9b65e90de5666fcbfd3e34957cc73ee19aeb5", size = 465974, upload-
time = "2026-05-18T04:30:17.006Z" },
    { url = "https://files.pythonhosted.org/packages/a7/6c/89b1a230a78f57c52dd8893adb1f92f94411721b6ec12596c56d98c74356/watchfiles-1.2.0-cp313-cp313-m
anylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:ca148d73dea36c9763aaa351e4d7a51780ec1584217c45276f4fe8239c768b71", size = 454782, uploa
d-time = "2026-05-18T04:30:35.656Z" },
    { url = "https://files.pythonhosted.org/packages/24/62/1732118367cfff0a9fce3bf62ff4bfded09ef5df21d9d446b858b3f70a96/watchfiles-1.2.0-cp313-cp313-m
anylinux_2_31_riscv64.whl", hash = "sha256:c525543d91961c6955b2636b308569e84a1d1c5f5f2932041ab9ef46422f43e3", size = 465182, upload-time = "2026-05-18
T04:30:20.846Z" },
    { url = "https://files.pythonhosted.org/packages/28/96/716f7e5f51339bf22963f3345f9f27d7f3b30e2eadc597e257c881dd3c53/watchfiles-1.2.0-cp313-cp313-m
usllinux_1_1_aarch64.whl", hash = "sha256:a204794696ffb8f9b10fba6f7cb5216d42f3b2b71860ccac6b6e42f5f10973b0", size = 629841, upload-time = "2026-05-18T
04:31:05.397Z" },
    { url = "https://files.pythonhosted.org/packages/4c/fe/c40783950fd771ccf66ab3ec2722d188a9af1c7f96c6e811f36e40c6e03f/watchfiles-1.2.0-cp313-cp313-m
usllinux_1_1_x86_64.whl", hash = "sha256:10d86db20695afe7997ac9e1717637d6714a8d0220458c33f3d2061f54cec427", size = 658028, upload-time = "2026-05-18T0
4:31:48.22Z" },
    { url = "https://files.pythonhosted.org/packages/71/72/4508db1856d1d87fcbb3b63f4839bab1b5682cb0e8d224d122263c09654a/watchfiles-1.2.0-cp313-cp313-w
in32.whl", hash = "sha256:eb283ee99e21ad6443c8cdb06ac5b34b1308c329cbdf03fa02b445363714c799", size = 275183, upload-time = "2026-05-18T04:30:59.57Z" },
    { url = "https://files.pythonhosted.org/packages/f9/36/14b76ca57652e5cc5fd1c11f32a261292c08a0d19a00351013c2549cbfb2/watchfiles-1.2.0-cp313-cp313-w
in_amd64.whl", hash = "sha256:a0f27f01bee51861392bb6b7c4fdb290b27d1eb194e9e28788d68102a0e898d9", size = 288059, upload-time = "2026-05-18T04:32:07.937
Z" },
    { url = "https://files.pythonhosted.org/packages/1b/8d/0a85e395398d8d20fadfe5c5d32c726eee17a519e78fb356f2cf7531bffe/watchfiles-1.2.0-cp313-cp313-w
in_arm64.whl", hash = "sha256:3651aa7058595e9cfb75d35dd5ada2bf9f48a5b8a0f3562821d3e210c507e077", size = 280186, upload-time = "2026-05-18T04:31:54.484
Z" },
    { url = "https://files.pythonhosted.org/packages/37/68/36db056f1fdcc5f07302f56e631774d6835bcd6fa3ace402304621d5f9e5/watchfiles-1.2.0-cp313-cp313t-
macosx_10_12_x86_64.whl", hash = "sha256:faea288b6f0ab1902ef08f4ca6de005dccf856c4e0c4f21b8c5fce02d90a1b08", size = 399031, upload-time = "2026-05-18T0
4:30:44.576Z" },
    { url = "https://files.pythonhosted.org/packages/c1/64/01a9d6f66a82a5c101ce939274106cc72759d62427e153f01edd2b9f87c2/watchfiles-1.2.0-cp313-cp313t-
macosx_11_0_arm64.whl", hash = "sha256:01859b11fd9fbca670f4d5da00fbac282cfea9bd67a2125d8b2833a3b5617ea9", size = 391205, upload-time = "2026-05-18T04:
30:25.413Z" },
    { url = "https://files.pythonhosted.org/packages/84/2c/0a44fe058cb4bb7b8ede6b6670698bbb7c0400740e378d00022189b7b31d/watchfiles-1.2.0-cp313-cp313t-
manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:fff610d7bb2256a317bb1e96f0d7862c7aa8076733ee5df0fd41bbe76a24a4f4", size = 451892, up
load-time = "2026-05-18T04:32:14.005Z" },
    { url = "https://files.pythonhosted.org/packages/67/a1/351e0d56cd35e6488b5c8b4fb11a809a5bc923e8fe8fed9faf8920be0c89/watchfiles-1.2.0-cp313-cp313t-
manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:b141a4891c995a039cd89e9a49e62df1dc8a559a5d1a6e4c7106d16c12777a55", size = 458867, uplo
ad-time = "2026-05-18T04:31:22.279Z" },
    { url = "https://files.pythonhosted.org/packages/d5/7d/9d09605187f1b838998624049fcf8bf47b73c1a3b76901fcac1782f62277/watchfiles-1.2.0-cp313-cp313t-
```

```
manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:f22943b7770483f6ea0721c6b11d022947a98eb0acae14694de034f4d0d38925", size = 490217, upload-time = "2026-05-18T04:31:43.657Z" },
    { url = "https://files.pythonhosted.org/packages/60/5d/a17a16eccb182f04188cd308ec24b1a71a9b5c4e7098269cf35d9fa56d02/watchfiles-1.2.0-cp313-cp313t-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:1bc6195825b7dcd217968bb1f801a60fd4c16e8eeab5bedc7fe917d7d5995ab4", size = 571458, upload-time = "2026-05-18T04:32:11.875Z" },
    { url = "https://files.pythonhosted.org/packages/d3/3d/4dd457062083ab1938e5dfd45032eb425cee2ac817287ca8ff4356183e5d/watchfiles-1.2.0-cp313-cp313t-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:d4a4b147f5dca2a5d325a06a832fb43f345751adfbc63204aec30e0d9ca965a2", size = 464707, upload-time = "2026-05-18T04:30:43.492Z" },
    { url = "https://files.pythonhosted.org/packages/c6/71/ea8c57b128f5383de74d0c7d2d9c57ad7c9a65a930c451bd25d524b295b7/watchfiles-1.2.0-cp313-cp313t-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:4543579a9bdb0c9560039b4ffddbdb39545707659fbc430ce4c10f3f68d557f9", size = 454663, upload-time = "2026-05-18T04:30:16.061Z" },
    { url = "https://files.pythonhosted.org/packages/53/fd/2e812bf938406d7db351f0703ddd3fc6c061cf30d96153a77bc79a943a44/watchfiles-1.2.0-cp313-cp313t-manylinux_2_31_riscv64.whl", hash = "sha256:20aa0e708b920bde876a4aa82dc7dd6ebea228a63a67cda6632c2fc87b787efa", size = 463537, upload-time = "2026-05-18T04:31:44.9Z" },
    { url = "https://files.pythonhosted.org/packages/86/56/d17a7f1dd1bc3035f1072694a551301272f1739c2d8e319c927cb9e29b38/watchfiles-1.2.0-cp313-cp313t-musllinux_1_1_aarch64.whl", hash = "sha256:d413349d565dab74297f2a63e84a097936be69bf8f3b3801f27f380e32040f44", size = 629194, upload-time = "2026-05-18T04:31:14.141Z" },
    { url = "https://files.pythonhosted.org/packages/be/06/f1ff66bf5cae50aa4062779a0ecd0bbaf15e466195719074078947d9a17d/watchfiles-1.2.0-cp313-cp313t-musllinux_1_1_x86_64.whl", hash = "sha256:f28b2725eb8cce327b9b3ab02415c853011dc55c95832fe90de6bc56f5315f72", size = 656194, upload-time = "2026-05-18T04:31:47.14Z" },
    { url = "https://files.pythonhosted.org/packages/e7/54/a9c7ea9a82a4ac65e7004c0a03920b5cdd2f9c3b678757d9cd425aa51d53/watchfiles-1.2.0-cp314-cp314-macosx_10_12_x86_64.whl", hash = "sha256:b8c8358484d5fa12ef34f05b7f4168eaf1932f408725ff6d023c33ec17bd79d4", size = 400205, upload-time = "2026-05-18T04:32:05.153Z" },
    { url = "https://files.pythonhosted.org/packages/aa/5d/c9ab3534374a4a67450696905d6ef16a04405448b8dc52bd752ae50423d4/watchfiles-1.2.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:9f04b092229ad2c50126dd3c922c8822e51e605993764a33058d4a791ab42281", size = 392508, upload-time = "2026-05-18T04:30:54.849Z" },
    { url = "https://files.pythonhosted.org/packages/26/ca/1ad30103535cf0cecd7b993e8d50edc5351b1820e38f2d22e3df58962feb/watchfiles-1.2.0-cp314-cp314-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:7a7ce236284f002a156f70add88efe5c70879cccbb658be0822c54b1306fc09d", size = 452448, upload-time = "2026-05-18T04:30:53.727Z" },
    { url = "https://files.pythonhosted.org/packages/37/a1/ceee2cdf2afbd715fa07758d39c9859513eae411b23196f7fd039e5feedd/watchfiles-1.2.0-cp314-cp314-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:b9909cc2b48468b575eefa944919e1fe8a36c5849d5c7c168f80a8c1db69398e", size = 459605, uploaded-time = "2026-05-18T04:30:23.312Z" },
    { url = "https://files.pythonhosted.org/packages/e8/f6/421e30fd1cb3907a84ed92ab3f1983e37ba2dca015e9a894a048418417a2/watchfiles-1.2.0-cp314-cp314-manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:0a37faaed405c67e28e6be45a1fa4f206ef5a2860f27c237db9fa30704c38242", size = 490757, upload-time = "2026-05-18T04:30:47.358Z" },
    { url = "https://files.pythonhosted.org/packages/41/b0/55ed1b97ed08be7bba6f9a541cac15f2a858e1d74d2b07b6da70a82aab00/watchfiles-1.2.0-cp314-cp314-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:9649193aa27bd9ff2e80ff29bfaa93085496c7a3a377592823cc58b77ee88add", size = 568672, upload-time = "2026-05-18T04:30:38.915Z" },
    { url = "https://files.pythonhosted.org/packages/d1/cf/d8ae8a80dd7bafab395ea7681c10237311bbf34d37704a8c744e7cf31fc7/watchfiles-1.2.0-cp314-cp314-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:4e4ff8e37f99cf1da89e255e07c9c4b37c214038c4283707bdec308cb1b0ea1f", size = 464197, upload-time = "2026-05-18T04:30:09.914Z" },
    { url = "https://files.pythonhosted.org/packages/7c/8a/3076c496ca8dafe0e8cd03fcebdfc47be4b1174b4e5b24ff6e396e6b3af2/watchfiles-1.2.0-cp314-cp314-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:054dc20fd2e3132b4c3883b4a00d72fd6e1f56fdaf89fccd12e8057d74cd74d7", size = 453181, upload-time = "2026-05-18T04:30:14.829Z" },
    { url = "https://files.pythonhosted.org/packages/e5/10/9745e17c98e7b8a86454df0a3c7b5686bd650383f1e9f26e4ebcbd6cc0c0/watchfiles-1.2.0-cp314-cp314-manylinux_2_31_riscv64.whl", hash = "sha256:e140ed30ebde76796b686e67c182cff10ea2fbab186fafd1560f74bb5a473a6e", size = 465109, upload-time = "2026-05-18T04:30:28.123Z" },
    { url = "https://files.pythonhosted.org/packages/8f/95/8ef4a95481d3e0cb52d62a06fa6e972e81424be2d9698b91a2fecca9904c/watchfiles-1.2.0-cp314-cp314-musllinux_1_1_aarch64.whl", hash = "sha256:bb7e52ecf68ba46d22df23467b87cffeb2146908aa523ebfe803019618cfda06", size = 630653, upload-time = "2026-05-18T04:31:49.304Z" },
    { url = "https://files.pythonhosted.org/packages/fd/e4/3b3bf36b0f829b50c6ebcb8d031583863c59f923d6a6af3d485e470d0fac/watchfiles-1.2.0-cp314-cp314-musllinux_1_1_x86_64.whl", hash = "sha256:23282a321c8baf9b3a3c4afff673f9fe65eb7fdc2338d765ccad9d3d1916a5ba", size = 657838, upload-time = "2026-05-18T04:31:06.497Z" },
    { url = "https://files.pythonhosted.org/packages/21/b1/6cbbb50c1f3002ab568777d44aa21206dfb8807a840990c4037523b51812/watchfiles-1.2.0-cp314-cp314-win32.whl", hash = "sha256:c0db965c5f79aa49fe672d297cf1febc5ad149b658594944f49a54a2b96270a7", size = 275108, upload-time = "2026-05-18T04:30:06.891Z" },
    { url = "https://files.pythonhosted.org/packages/92/45/190ce6db8dcb4536682cf75d3889ff1a27182a58cb519d343cb6d9ea63d8/watchfiles-1.2.0-cp314-cp314-win_amd64.whl", hash = "sha256:71283b39fd17e5408eb123bd37aeecfd9d54c81fc184421943208aadb879d103", size = 288441, upload-time = "2026-05-18T04:32:12.901Z" },
    { url = "https://files.pythonhosted.org/packages/74/0d/3eae1c2313ab08378431d907c3f8095ecca00f3eda33111cf4f0f2591799/watchfiles-1.2.0-cp314-cp314-win_arm64.whl", hash = "sha256:c5c19526f4e54a00f2666a6c0e9e40d582c09e865055ea7378bf0009aab857b3", size = 280684, upload-time = "2026-05-18T04:31:26.902Z" },
    { url = "https://files.pythonhosted.org/packages/b1/75/fb64e6c25d6b5ca636d03df34ffb1c6e9873303e76d27967e045f8df088f/watchfiles-1.2.0-cp314-cp314t-macosx_10_12_x86_64.whl", hash = "sha256:d73a585accffa5ae39c17264c36ec3166d2fad7000c780f5ef83b2722afb9dd2", size = 398857, upload-time = "2026-05-18T04:32:17.108Z" },
    { url = "https://files.pythonhosted.org/packages/73/4e/9f7adf01754cbf81843722ccfec169d8f26c69778281a302855cecd2ee08/watchfiles-1.2.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:ae99b14c5f21e026e0e9d96f40e07d8570ebee6cafd9d8fc318354606daa7a28", size = 392413, upload-time = "2026-05-18T04:31:07.911Z" },
    { url = "https://files.pythonhosted.org/packages/47/c8/bec626bcc2d69f44b9acb24ce7d60ed7b16b73628eea747fcbd169d8edda/watchfiles-1.2.0-cp314-cp314t-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:4429f3b105524a10b72c3a819b091c495d2811d419c1e1e8df773a5a5974f831", size = 452409, upload-time = "2026-05-18T04:31:20.142Z" },
    { url = "https://files.pythonhosted.org/packages/00/b7/b6362068e81e7c556d155a34c35d40ac3ef42d747b06d7f6e5bf58e359c2/watchfiles-1.2.0-cp314-cp314t-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:43d818978d06062d9b22c4fab2ebe44cf5213d42dc8e62bda8c2760cfa2eeb33", size = 458827, upload-time = "2026-05-18T04:32:06.219Z" },
    { url = "https://files.pythonhosted.org/packages/67/f8/9a813fa42afb1e0b4625e75f0479826644d3ee8dc287e093799bc01f390c/watchfiles-1.2.0-cp314-cp314t-manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:b9f732dc58b2dbe69e464ccf8fff7a03b0dd0be439da4c0720d3558527d3d6b4", size = 490104, upload-time = "2026-05-18T04:31:56.034Z" },
    { url = "https://files.pythonhosted.org/packages/2f/bf/27dfb6094ca4c9aad21298b5525b6c53cb36121ee454331d05161e58d130/watchfiles-1.2.0-cp314-cp314t-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:8f200104103feb097de4cab8fe4f5dd18a2026934c7dea98c55a2f5fd6d5a33b", size = 571360, upload-time = "2026-05-18T04:31:57.133Z" },
    { url = "https://files.pythonhosted.org/packages/fb/39/44a096d67270ea93df91d33877dbe91fbda3aa4f8ec2edf799d93eda8736/watchfiles-1.2.0-cp314-cp314t-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:63ac26eefbf4af1741247d6fb68b11c49a25b2f7413fbd318a83a12aaa9cf666", size = 464644, upload-time = "2026-05-18T04:30:57.33Z" },
    { url = "https://files.pythonhosted.org/packages/0e/80/c7472203bad6268e3ef1ad26073970484789893 8ad7ea8b63a5131f46b50/watchfiles-1.2.0-cp314-cp314t-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:0c4997d4e4a55f0d02b6cde327322daf3a0400e5df6c6b15948994bf72497925", size = 454771, upload-time = "2026-05-18T04:30:48.736Z" },
    { url = "https://files.pythonhosted.org/packages/51/cf/3b10b268b4b7f0fc26e9debb5eef1998b515887840f444cd3ec80c688755/watchfiles-1.2.0-cp314-cp314t-manylinux_2_31_riscv64.whl", hash = "sha256:4c887eba18b7945ac73067a8b4a66f21cd46c2539b2bc68588f7be6c7eb6d26b", size = 463494, upload-time = "2026-05-18T04:31:33.826Z" },
    { url = "https://files.pythonhosted.org/packages/3d/3e/a4302545cd589262a0dc7d140e86f7688eba3f9c72776c27f7e23b8864c4/watchfiles-1.2.0-cp314-cp314t-musllinux_1_1_aarch64.whl", hash = "sha256:3416ff151bb6b5a8d8d11664974fbef4d9305b9b2957839ab5a270468fd8df30", size = 629383, upload-time = "2026-05-18T04:31:15.596Z" },
    { url = "https://files.pythonhosted.org/packages/db/99/d5649df0a9a410d45b7c882304d0b790903ac9b6e8f2cfd12114e0c6b9f2/watchfiles-1.2.0-cp314-cp314t-musllinux_1_1_x86_64.whl", hash = "sha256:0e831a271c035d89789cffc386b6aa1375f39f1cd25eb7ca0997e4970d152fc5", size = 656093, upload-time = "2026-05-18T04:31:58.707Z" },
    { url = "https://files.pythonhosted.org/packages/92/b9/3627025392750 19a54dd2e94511b31a9b89c5f9e6a21966de7eb692549fc/watchfiles-1.2.0-cp315-cp315-macosx_10_12_x86_64.whl", hash = "sha256:37a6721cdf3f65dbb13aa9503510ccb4451603ac837e44d265d7992a597e1374", size = 400109, upload-time = "2026-05-18T04:31:16.879Z" },
    { url = "https://files.pythonhosted.org/packages/8f/75/71d5ba62db781e5587bded1d944c675374bc4aa37ff33d5018d98e8b6538/watchfiles-1.2.0-cp315-cp315-macosx_11_0_arm64.whl", hash = "sha256:2b37d10b5a63bd4d87e18472d80fa525bd670586fae62e5dd580452764879b65", size = 392167, upload-time = "2026-05-18T04:31:28.058Z" },
    { url = "https://files.pythonhosted.org/packages/3c/01/c66dd95d0423fe30d31820e2d1d5bda773764131bbb6ac0cb1cf303ac328/watchfiles-1.2.0-cp315-cp315-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:0a105bc2283f67e8fbec74253ec2d94925de92ed72c0393f1206bf326b7b7b69", size = 452372, upload-time = "2026-05-18T04:31:00.836Z" },
    { url = "https://files.pythonhosted.org/packages/91/15/2fe99557e72f85627c6a8eed50d889e8d101623e060a22ad75b875cb932d/watchfiles-1.2.0-cp315-cp315-m
```

```
anylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:5327989a465505f05cfe06f04fa9d0c2fd5432bb243e10e6f012b1bdca3c8579", size = 459596, uploa
d-time = "2026-05-18T04:31:34.96Z" },
    { url = "https://files.pythonhosted.org/packages/ed/23/d4acfa0023367428ed48351b3b9b267893037b6cadae55620c61c24bcfd4/watchfiles-1.2.0-cp315-cp315-m
anylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:ecb47f183a8025b2aa18b546725c3657e542112ae9c0613a2af79b4fa8d04ad7", size = 490869, upload-ti
me = "2026-05-18T04:31:59.923Z" },
    { url = "https://files.pythonhosted.org/packages/a4/5f/3164cbdce06c9fb95c4f7b9e2f9760b5e2797af43a9ecc317ef42a23a278/watchfiles-1.2.0-cp315-cp315-m
anylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:8520a4ab0e37f770afc34459c4f8f7019e153f9124dc101c15538365875d1ab2", size = 571641, upl
oad-time = "2026-05-18T04:32:00.948Z" },
    { url = "https://files.pythonhosted.org/packages/41/e6/85d3731c55e65cd7690f3f803d24c139588aaf863e4bf2148fe7a7fa1a19/watchfiles-1.2.0-cp315-cp315-m
anylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:71cd71740ed2c15211ebb237ced4e39a1cdf6f80566e5fe95428da1626f4fde6", size = 464444, upload-
time = "2026-05-18T04:30:34.298Z" },
    { url = "https://files.pythonhosted.org/packages/f4/7d/562641012b8b09872742c3b8adf9629ec479fd78f8d68ae4a0c13da8add6/watchfiles-1.2.0-cp315-cp315-m
anylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:f88af53d6ddaf72179ef613ddc905e6f4785f712b49b80b3bef9f3525e6194b4", size = 453593, uploa
d-time = "2026-05-18T04:31:23.464Z" },
    { url = "https://files.pythonhosted.org/packages/56/fe/cb8ef3d6f929d14158fdaaad9925985b7310abc9384dcd4d82dd0016fb59/watchfiles-1.2.0-cp315-cp315-m
anylinux_2_31_riscv64.whl", hash = "sha256:cee9d5efd929efdac5f7e58f72b3376f676b64050a91c5b99a7094c5b2317488", size = 465096, upload-time = "2026-05-18
T04:31:30.384Z" },
    { url = "https://files.pythonhosted.org/packages/25/91/80908e835e100527a9267147b08c0eee1fa6ab0ffec15edc04d1d44885f7/watchfiles-1.2.0-cp315-cp315-m
usllinux_1_1_aarch64.whl", hash = "sha256:b718bf356bbc15e559bd8ef41782b573b8ae0e3f177ab244b440568d7ea02cfb", size = 630638, upload-time = "2026-05-18T
04:30:49.89Z" },
    { url = "https://files.pythonhosted.org/packages/46/4b/95ab2f256bb4af3cb2eb23b9317bda984ee6e0f11733a5c004a6c95b06e3/watchfiles-1.2.0-cp315-cp315-m
usllinux_1_1_x86_64.whl", hash = "sha256:922c0e019fe68b3ae392965a766b02a71ba1168c932cebc3733cd52c5fe5b377", size = 657684, upload-time = "2026-05-18T0
4:31:32.027Z" },
]

[[package]]
name = "websockets"
version = "15.0.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/21/e6/26d09fab466b7ca9c7737474c52be4f76a40301b08362eb2dbc19dcc16c1/websockets-15.0.1.tar.gz",
 hash = "sha256:82544de02076bafba038ce055ee6412d68da13ab47f0c60cab827346de828dee", size = 177016, upload-time = "2025-03-05T20:03:41.606Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/51/6b/4545a0d843594f5d0771e86463606a3988b5a09ca5123136f8a76580dd63/websockets-15.0.1-cp312-cp312-
macosx_10_13_universal2.whl", hash = "sha256:3e90baa811a5d73f3ca0bcbf32064d663ed81318ab225ee4f427ad4e26e5aff3", size = 175437, upload-time = "2025-03-
05T20:02:16.706Z" },
    { url = "https://files.pythonhosted.org/packages/f4/71/809a0f5f6a06522af902e0f2ea2757f71ead94610010cf570ab5c98e99ed/websockets-15.0.1-cp312-cp312-
macosx_10_13_x86_64.whl", hash = "sha256:592f1a9fe869c778694f0aa806ba0374e97648ab57936f092fd9d87f8bc03665", size = 173096, upload-time = "2025-03-05T2
0:02:18.832Z" },
    { url = "https://files.pythonhosted.org/packages/3d/69/1a681dd6f02180916f116894181eab8b2e25b31e484c5d0eae637ec01f7c/websockets-15.0.1-cp312-cp312-
macosx_11_0_arm64.whl", hash = "sha256:0701bc3cfcb9164d04a14b149fd74be7347a530ad3bbf15ab2c678a2cd3dd9a2", size = 173332, upload-time = "2025-03-05T20:
02:20.187Z" },
    { url = "https://files.pythonhosted.org/packages/a6/02/0073b3952f5bce97eafbb35757f8d0d54812b6174ed8dd952aa08429bcc3/websockets-15.0.1-cp312-cp312-
manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:e8b56bdcdb4505c8078cb6c7157d9811a85790f2f2b3632c7d1462ab5783d215", size = 183152, up
load-time = "2025-03-05T20:02:22.286Z" },
    { url = "https://files.pythonhosted.org/packages/74/45/c205c8480eafd114b428284840da0b1be9ffd0e4f87338dc95dc6ff961a1/websockets-15.0.1-cp312-cp312-
manylinux_2_5_i686.manylinux1_i686.manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:0af68c55afbd5f07986df82831c7bff04846928ea8d1fd7f3005263
8788bc9b5", size = 182096, upload-time = "2025-03-05T20:02:24.368Z" },
    { url = "https://files.pythonhosted.org/packages/14/8f/aa61f528fba38578ec553c145857a181384c72b98156f858ca5c8e82d9d3/websockets-15.0.1-cp312-cp312-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:64dee438fed052b52e4f98f76c5790513235efaa1ef7f3f
2192c392cd7c91b65", size = 182523, upload-time = "2025-03-05T20:02:25.669Z" },
    { url = "https://files.pythonhosted.org/packages/ec/6d/0267396610add5bc0d0d3e77f546d4cd287200804fe02323797de77dbce9/websockets-15.0.1-cp312-cp312-
musllinux_1_2_aarch64.whl", hash = "sha256:d5f6b181bb38171a8ad1d6aa58a67a6aa9d4b38d0f8c5f496b9e42561dfc62fe", size = 182790, upload-time = "2025-03-05
T20:02:26.99Z" },
    { url = "https://files.pythonhosted.org/packages/02/05/c68c5adbf679cf610ae2f74a9b871ae84564462955d991178f95a1ddb7dd/websockets-15.0.1-cp312-cp312-
musllinux_1_2_i686.whl", hash = "sha256:5d54b09eba2bada6011aea5375542a157637b91029687eb4fdb2dab11059c1b4", size = 182165, upload-time = "2025-03-05T20
:02:30.291Z" },
    { url = "https://files.pythonhosted.org/packages/29/93/bb672df7b2f5faac89761cb5fa34f5cec45a4026c383a4b5761c6cea5c16/websockets-15.0.1-cp312-cp312-
musllinux_1_2_x86_64.whl", hash = "sha256:3be571a8b5afed347da347bfcf27ba12b069d9d7f42cb8c7028b5e98bbb12597", size = 182160, upload-time = "2025-03-05T
20:02:31.634Z" },
    { url = "https://files.pythonhosted.org/packages/ff/83/de1f7709376dc3ca9b7eeb4b9a07b4526b14876b6d372a4dc62312bebee0/websockets-15.0.1-cp312-cp312-
win32.whl", hash = "sha256:c338ffa0520bdb12fbc527265235639fb76e7bc7faafbb93f6ba80d9c06578a9", size = 176395, upload-time = "2025-03-05T20:02:33.017Z"
},
    { url = "https://files.pythonhosted.org/packages/7d/71/abf2ebc3bbfa40f391ce1428c7168fb20582d0ff57019b69ea20fa698043/websockets-15.0.1-cp312-cp312-
win_amd64.whl", hash = "sha256:fcd5cf9e305d7b8338754470cf69cf81f420459dbae8a3b40cee57417f4614a7", size = 176841, upload-time = "2025-03-05T20:02:34.49
8Z" },
    { url = "https://files.pythonhosted.org/packages/cb/9f/51f0cf64471a9d2b4d0fc6c534f323b664e7095640c34562f5182e5a7195/websockets-15.0.1-cp313-cp313-
macosx_10_13_universal2.whl", hash = "sha256:ee443ef070bb3b6ed74514f5efaa37a252af57c90eb33b956d35c8e9c10a1931", size = 175440, upload-time = "2025-03-
05T20:02:36.695Z" },
    { url = "https://files.pythonhosted.org/packages/8a/05/aa116ec9943c718905997412c5989f7ed671bc0188ee2ba89520e8765d7b/websockets-15.0.1-cp313-cp313-
macosx_10_13_x86_64.whl", hash = "sha256:5a939de6b7b4e18ca683218320fc67ea886038265fd1ed30173f5ce3f8e85675", size = 173098, upload-time = "2025-03-05T2
0:02:37.985Z" },
    { url = "https://files.pythonhosted.org/packages/ff/0b/33cef55ff24f2d92924923c99926dcce78e7bd922d649467f0eda8368923/websockets-15.0.1-cp313-cp313-
macosx_11_0_arm64.whl", hash = "sha256:746ee8dba912cd6fc889a8147168991d50ed70447bf18bcda7039f7d2e3d9151", size = 173329, upload-time = "2025-03-05T20:
02:39.298Z" },
    { url = "https://files.pythonhosted.org/packages/31/1d/063b25dcc01faa8fada1469bdf769de3768b7044eac9d41f734fd7b6ad6d/websockets-15.0.1-cp313-cp313-
manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:595b6c3969023ecf9041b2936ac3827e4623bfa3ccf007575f04c5a6aa318c22", size = 183111, up
load-time = "2025-03-05T20:02:40.595Z" },
    { url = "https://files.pythonhosted.org/packages/93/53/9a87ee494a51bf63e4ec9241c1ccc4f7c2f45fff85d5bde2ff74fcb68b9e/websockets-15.0.1-cp313-cp313-
manylinux_2_5_i686.manylinux1_i686.manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:3c714d2fc58b5ca3e285461a4cc0c9a66bd0e24c5da9911e3015828
6c9b5be7f", size = 182054, upload-time = "2025-03-05T20:02:41.926Z" },
    { url = "https://files.pythonhosted.org/packages/ff/b2/83a6ddf56cdcbad4e3d841fcc55d6ba7d19aeb89c50f24dd7e859ec0805f/websockets-15.0.1-cp313-cp313-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:0f3c1e2ab208db911594ae5b4f79addeb3501604a165019
dd221c0bdcabe4db8", size = 182496, upload-time = "2025-03-05T20:02:43.304Z" },
    { url = "https://files.pythonhosted.org/packages/98/41/e7038944ed0abf34c45aa4635ba28136f06052e08fc2168520bb8b25149f/websockets-15.0.1-cp313-cp313-
musllinux_1_2_aarch64.whl", hash = "sha256:229cf1d3ca6c1804400b0a9790dc66528e08a6a1feec0d5040e8b9eb14422375", size = 182829, upload-time = "2025-03-05
T20:02:48.812Z" },
    { url = "https://files.pythonhosted.org/packages/e0/17/de15b6158680c7623c6ef0db361da965ab25d813ae54fcfeae2e5b9ef910/websockets-15.0.1-cp313-cp313-
musllinux_1_2_i686.whl", hash = "sha256:756c56e867a90fb00177d530dca4b097dd753cde348448a1012ed6c5131f8b7d", size = 182217, upload-time = "2025-03-05T20
:02:50.14Z" },
    { url = "https://files.pythonhosted.org/packages/33/2b/1f168cb6041853eef0362fb9554c3824367c5560cbdaad89ac40f8c2edfc/websockets-15.0.1-cp313-cp313-
musllinux_1_2_x86_64.whl", hash = "sha256:558d023b3df0bffe50a04e710bc87742de35060580a293c2a984299ed83bc4e4", size = 182195, upload-time = "2025-03-05T
20:02:51.561Z" },
    { url = "https://files.pythonhosted.org/packages/86/eb/20b6cdf273913d0ad05a6a14aed4b9a85591c18a987a3d47f20fa13dcc47/websockets-15.0.1-cp313-cp313-
win32.whl", hash = "sha256:ba9e56e8ceeeedb2e080147ba85ffcd5cd0711b89576b83784d8605a7df455fa", size = 176393, upload-time = "2025-03-05T20:02:53.814Z"
},
    { url = "https://files.pythonhosted.org/packages/1b/6c/c65773d6cab416a64d191d6ee8a8b1c68a09970ea6909d16965d26bfed1e/websockets-15.0.1-cp313-cp313-
win_amd64.whl", hash = "sha256:e09473f095a819042ecb2ab9465aee615bd9c2028e4ef7d933600a8401c79561", size = 176837, upload-time = "2025-03-05T20:02:55.23
7Z" },
    { url = "https://files.pythonhosted.org/packages/fa/a8/5b41e0da817d64113292ab1f8247140aac61cbf6cfd085d6a0fa77f4984f/websockets-15.0.1-py3-none-any
.whl", hash = "sha256:f7a866fbc1e97b5c617ee4116daaa09b722101d4a3c170c787450ba409f9736f", size = 169743, upload-time = "2025-03-05T20:03:39.41Z" },
]

[[package]]
name = "win32-setctime"
version = "1.2.0"
```

```
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/b3/8f/705086c9d734d3b663af0e9bb3d4de6578d08f46b1b101c2442fd9aecaa2/win32_setctime-1.2.0.tar.gz", hash = "sha256:ae1fdf948f5640aae05c511ade119313fb6a30d7eabe25fef9764dca5873c4c0", size = 4867, upload-time = "2024-12-07T15:28:28.314Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e1/07/c6fe3ad3e685340704d314d765b7912993bcb8dc198f0e7a89382d37974b/win32_setctime-1.2.0-py3-none-any.whl", hash = "sha256:95d644c4e708aba81dc3704a116d8cbc974d70b3bdb8be1d150e36be6e9d1390", size = 4083, upload-time = "2024-12-07T15:28:26.465Z" },
]

[[package]]
name = "wrapt"
version = "2.3.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/2b/b0/c1f5a970721f06b85c0cd5142e0ff8fe067708abd779b0c4f4be7d61d09f/wrapt-2.3.0.tar.gz", hash = "sha256:681a2d0eefd721998f90642762b8e75c2159ec531b20ad5e437245ea7b06a107", size = 131509, upload-time = "2026-07-28T06:06:14.895Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/5b/4a/d17a0fad1bf1c5f2c887ff71fef75654141b0880bff71d157d955b5bec3a/wrapt-2.3.0-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:0a45ffae742ce91a16e11cb6c7cd71e7f9994f3cbd283b962ab093f5c6dcf525", size = 82139, upload-time = "2026-07-28T06:04:35.082Z" },
    { url = "https://files.pythonhosted.org/packages/6e/55/51b92daaf6defb57f4dc56bdcce985400f75c6984a03ca5e78ccac717028/wrapt-2.3.0-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:69e477046f2237ef0bc6547544ee73008dc764ca26eff44f09e976d221b34d5d", size = 82723, upload-time = "2026-07-28T06:04:36.502Z" },
    { url = "https://files.pythonhosted.org/packages/28/7f/cfd9bc4b1f5e424eeea83d0493e43f3b1b02707ce8e50c47945873982bd5/wrapt-2.3.0-cp312-cp312-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:5d221a6e6ddd302b8397433184e96b59f259f50024b854db1c411a881586b6b8", size = 172381, upload-time = "2026-07-28T06:04:37.674Z" },
    { url = "https://files.pythonhosted.org/packages/cb/89/ff7814f6eb6856b479946117d1138a2fbb46cdb6b1f379db359056c69743/wrapt-2.3.0-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:392158c9a7f2ab1b8699418bfc0fe6f83548788c418b27d7bf2019ad3405cebb", size = 174120, upload-time = "2026-07-28T06:04:38.987Z" },
    { url = "https://files.pythonhosted.org/packages/12/1e/8eded8615d39e3ce81f626937a3a87b280a2a86239a2bf14a4b4bb345034/wrapt-2.3.0-cp312-cp312-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:e5301c35cf75655eb33498f2bd6ae8703ca19940e3167dc9cdf740c712a39c60", size = 163035, upload-time = "2026-07-28T06:04:40.361Z" },
    { url = "https://files.pythonhosted.org/packages/35/ea/a0af2d9da62897af2a055484920de05dade30d2ba2c0d65cbdea875d3d8b/wrapt-2.3.0-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:418f54bb09d1762db02c7009b4051149893af3153a87f92d70356703c11eea02", size = 171887, upload-time = "2026-07-28T06:04:41.614Z" },
    { url = "https://files.pythonhosted.org/packages/7e/dd/63cd4c864c65ef4906df64bd2d378f4a62b54f28063f282dfb3bf93caead/wrapt-2.3.0-cp312-cp312-musllinux_1_2_riscv64.whl", hash = "sha256:1598becd30f8f2777d18564064eb4f4dbe1ab0e05a8f09786d0ef505ac782bf3", size = 161113, upload-time = "2026-07-28T06:04:42.864Z" },
    { url = "https://files.pythonhosted.org/packages/ca/ee/82f1fc9e431b5c2c5a6d201aa865dbeae3984c311c6d11a185f0c8367cf6/wrapt-2.3.0-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:3da470536bf9645143323dd41b32db55c6f4304ad382094c1a1da8a92061e10d", size = 170530, upload-time = "2026-07-28T06:04:44.212Z" },
    { url = "https://files.pythonhosted.org/packages/37/a5/5dc590e863a419930d988f8b7ca3e75a6befcfb10b6003b3a152f3d5f732/wrapt-2.3.0-cp312-cp312-win32.whl", hash = "sha256:fb8e2e6704a1e0b1b989546c69e2688371ef4a07fa5f61bde3eb6211186f5ac1", size = 78323, upload-time = "2026-07-28T06:04:45.484Z" },
    { url = "https://files.pythonhosted.org/packages/51/f9/4a6925a07951df56394f7e6ebe14f69f1c5ef9d87aa63e0839acf15aa63a/wrapt-2.3.0-cp312-cp312-win_amd64.whl", hash = "sha256:cdc021cb0b62471d6aac7f2bd92f3b4658073775f9ee7fcd325c511129e7bcc8", size = 81180, upload-time = "2026-07-28T06:04:47.021Z" },
    { url = "https://files.pythonhosted.org/packages/a8/4f/8b5de0395b2a72216751d41c9861df6facaeb611b619d8810ed2b3b23eb2/wrapt-2.3.0-cp312-cp312-win_arm64.whl", hash = "sha256:67bfe2485f50368c3fcd2275fc1fd100e350d601e0058921a7c82678a465aeab", size = 80155, upload-time = "2026-07-28T06:04:48.373Z" },
    { url = "https://files.pythonhosted.org/packages/8e/6e/0f88a072483e76b881e3fdcd6b6ffb4a5791002514fe541e72b1b73c859a/wrapt-2.3.0-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:0d3fb71e65b001adfc42684522eeccd9c21d8ba679945abc993439567b66e59f", size = 81960, upload-time = "2026-07-28T06:04:49.622Z" },
    { url = "https://files.pythonhosted.org/packages/d7/ff/b7e2776e7c294075eb712cc9ef573d1b818f393006d09787262b8fc871c4/wrapt-2.3.0-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:51a7a4181c1295774812271fbcd7c909df372bc25579d4ed9eb875caaf0ae86f", size = 82435, upload-time = "2026-07-28T06:04:50.9Z" },
    { url = "https://files.pythonhosted.org/packages/d8/90/343bb5d0f1f9669bc252a6073f085b4abf862511bd5c9c9eaec754341f1d/wrapt-2.3.0-cp313-cp313-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:9045917809c63fdf7abe3a2ceaed3d670b8ee4500ddd9291192d30aeb34467c5", size = 170350, upload-time = "2026-07-28T06:04:52.187Z" },
    { url = "https://files.pythonhosted.org/packages/59/f8/13b79a392930bd0dd6b86cbfbfe1c40944110456e1dc6d809e5c46ece904/wrapt-2.3.0-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:54ca1d5573f69b5fe1d74f1f65799c68015e82f685efec9fd8cfa40a094c44d0", size = 170022, upload-time = "2026-07-28T06:04:53.599Z" },
    { url = "https://files.pythonhosted.org/packages/b2/fc/4f1b6918f5290db959d6e0c07f77385d87cede29c39c9cf8f145e9c82954/wrapt-2.3.0-cp313-cp313-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:242b60c21e30866e6a2fa606c612b47c553fa60c0eaeeeb7797fb842ac0ce609", size = 161043, upload-time = "2026-07-28T06:04:54.936Z" },
    { url = "https://files.pythonhosted.org/packages/01/e1/45d3cf74414780bdff6d0380467e003f6eb0f028b6c9403db868dbc7209c/wrapt-2.3.0-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:e3f3d7ec0a51fbfe00d3aef047641ff2c58b25565b4717fc1f90e050be01cba8", size = 168576, upload-time = "2026-07-28T06:04:56.261Z" },
    { url = "https://files.pythonhosted.org/packages/f3/73/2fa58dd97f191c997755e2c6d569a68f0c433db4e4b36099bdd7227b6cac/wrapt-2.3.0-cp313-cp313-musllinux_1_2_riscv64.whl", hash = "sha256:261f53870cd4fb2bf38f9f972c56c728fd224cb7c65721307de59d9e7e6741ae", size = 159140, upload-time = "2026-07-28T06:04:57.754Z" },
    { url = "https://files.pythonhosted.org/packages/29/a8/08a56e2000a8816d449dcbad8c8b081697acbbd490821ceca0f9d8e8d20c/wrapt-2.3.0-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:8159ec0b0cb7608175eb150de94c19e34f4d47ac655f5ca9baf45df6b688ffd3", size = 169263, upload-time = "2026-07-28T06:04:59.161Z" },
    { url = "https://files.pythonhosted.org/packages/9e/d4/354e1725e35a73b2af4fa70a3e024c7a5d1bf1802dfb862dcb668aae0253/wrapt-2.3.0-cp313-cp313-win32.whl", hash = "sha256:10461884b3014fbfc8eb7d09a93c5f246363e6711d9d881f95eb8c27fdef049f", size = 78241, upload-time = "2026-07-28T06:05:00.507Z" },
    { url = "https://files.pythonhosted.org/packages/6c/7e/34c87fa2174848dfee820322aaa318bab08913998ccecc8d2f57b4ad4639/wrapt-2.3.0-cp313-cp313-win_amd64.whl", hash = "sha256:ac870cc97b73bb00ac353329e9559a4bebc47c4c86792ed9b23b58c15b6ad838", size = 81113, upload-time = "2026-07-28T06:05:01.839Z" },
    { url = "https://files.pythonhosted.org/packages/11/86/fcc9a530579e008c9478bb565a6cdfbfd33536660f069c8b91a6607c5050/wrapt-2.3.0-cp313-cp313-win_arm64.whl", hash = "sha256:a65e8db2b4e90c2e7ade931086351c98ef420bf7a94ee08c95ac8a3cbbc43579", size = 80182, upload-time = "2026-07-28T06:05:03.152Z" },
    { url = "https://files.pythonhosted.org/packages/96/50/3864848b95b28ef73e17551fc8dccbff2628a834f52cf26a57f9c419fb83/wrapt-2.3.0-cp313-cp313t-macosx_10_13_x86_64.whl", hash = "sha256:fd1f2f557dd3491fe75905e578f4db967393d40d1a8f468edc4d40ac7f2d5944", size = 83921, upload-time = "2026-07-28T06:05:04.476Z" },
    { url = "https://files.pythonhosted.org/packages/3b/4c/3d1921a60c3e8c71c540ff136e6a47a1fbccf7f671e818394889f7871d9c/wrapt-2.3.0-cp313-cp313t-macosx_11_0_arm64.whl", hash = "sha256:9f5d2aec29dfc76c37e23897dee92766a3fd4f3bff3ae7fc9c6b4bf37d8c1360", size = 84412, upload-time = "2026-07-28T06:05:05.921Z" },
    { url = "https://files.pythonhosted.org/packages/fa/1a/4a796ff7adb26ada6d4b758c94d47a38320b085e7099afc088efbbcdb006/wrapt-2.3.0-cp313-cp313t-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:646d20d413ffcd1b0a2f700076e2d0252d872dcb7754860a73e45a59ea883614", size = 207168, upload-time = "2026-07-28T06:05:07.256Z" },
    { url = "https://files.pythonhosted.org/packages/1d/3e/d7777776806c579b761bac2f91721dda9f04c7a1b380213c5935cc750ae6/wrapt-2.3.0-cp313-cp313t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:379f670f45b7bb8993edd9f6fc36c6cc65edb81cffa0b504be34acb0303fff0a", size = 214351, upload-time = "2026-07-28T06:05:08.945Z" },
    { url = "https://files.pythonhosted.org/packages/63/27/2d64d394df7bf181955b3bb562bf33c4492fb4be113f53071106d43ad8b5/wrapt-2.3.0-cp313-cp313t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:6208f302f110295d64b22a7ac96500c791bf492dce4366e622e4912b077c9687", size = 199020, upload-time = "2026-07-28T06:05:10.418Z" },
    { url = "https://files.pythonhosted.org/packages/3e/3d/fb31d3db7d9834d265fb1a27a2adf0ddf51557c67458c97b22439ad6ae3d/wrapt-2.3.0-cp313-cp313t-musllinux_1_2_aarch64.whl", hash = "sha256:ed635a9ca4f3a5a2b900c10c69e823373bc00ebc114b459383596d3487da3570", size = 209969, upload-time = "2026-07-28T06:05:11.983Z" },
    { url = "https://files.pythonhosted.org/packages/1f/d1/8724b5da582e62070dc9bf4d8bf1972f317297eefd7ba1f2b5c6393ccf6c/wrapt-2.3.0-cp313-cp313t-musllinux_1_2_riscv64.whl", hash = "sha256:e3b9eaa742ae7a0aaaaad4ca4b69469d757af2d6e6663ef1dadc47adec0aeb41", size = 196324, upload-time = "2026-07-28T06:05:13.557Z" },
    { url = "https://files.pythonhosted.org/packages/0d/5c/3d9ef411149543016ee6bcf3af707f787cebd946527452b94bf122e9b7b4/wrapt-2.3.0-cp313-cp313t-musllinux_1_2_x86_64.whl", hash = "sha256:d0f7284f88f4833705132d06d3b425a43095c2cbd07c58166aac3ab646ba12a4", size = 202610, upload-time = "2026-07-28T06:05:15.048Z" },
    { url = "https://files.pythonhosted.org/packages/13/9b/4fc042ceb757866dd4a5fc057b3b736f2b360d3703ce9f830d83dc9226e0/wrapt-2.3.0-cp313-cp313t-win32.whl", hash = "sha256:7ebb274aba688b043429eb1500ff8a76ce0cb8ac0812ca3e301f06247b8722b3", size = 79178, upload-time = "2026-07-28T06:05:16.469Z" },
    { url = "https://files.pythonhosted.org/packages/6b/ff/b94878f8eed809ca042685276bcea9f24e8c2ca7c9653bb80bbb920a68a5/wrapt-2.3.0-cp313-cp313t-win_amd64.whl", hash = "sha256:c4bded758ad6f03b965830944a2f0bc5b2eb3767fe5a7310134315d1a6610e98", size = 82634, upload-time = "2026-07-28T06:05:18.026Z" },
    { url = "https://files.pythonhosted.org/packages/80/fb/663e1de5332a71685a729754312d327d4cada767c36e1c5a2db4c8de49e6/wrapt-2.3.0-cp313-cp313t-win_arm64.whl", hash = "sha256:d2cc64539da63e39ffb9c7ede849b6e8ddaaf7b3876b5cfb04efd85a5f3f4eb6", size = 81387, upload-time = "2026-07-28T06:05:19.417Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/58/10/b073beaea89bc0d3670a75ff51139430a54b6af7ba7796507730634536dd/wrapt-2.3.0-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:ea52a0d0f08c584943d5764be0e84efa912c8da23c23e1e285ff2f5641c18fcc", size = 81978, upload-time = "2026-07-28T06:05:21.133Z" },
    { url = "https://files.pythonhosted.org/packages/b3/31/0916d9cebf848ed3f1a0c1888faee421747df77331e4db2bc527a9a85988/wrapt-2.3.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:fd85b0aa88efdb189d6ae2f35f4526943a8f091c38599c9c31478241c819e6a1", size = 82518, upload-time = "2026-07-28T06:05:22.562Z" },
    { url = "https://files.pythonhosted.org/packages/f5/73/31c1bf0f3384062751c2094dadb314916d70aa9b6bfd26d994b4a7b393fa/wrapt-2.3.0-cp314-cp314-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:141ed6211286a9660d8d6702de598b43f0934b4f0eda16393f100a80f501d945", size = 170187, upload-time = "2026-07-28T06:05:23.904Z" },
    { url = "https://files.pythonhosted.org/packages/ed/25/fce087d54b79b8905f3c3c9dd5f454bbd8d8acb80b960c4a6aee5b4659b3/wrapt-2.3.0-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:2e49885a62ec4ee854d1b9e6371fda6afd219917225752abf729a3f36d4df9a5", size = 169288, upload-time = "2026-07-28T06:05:25.378Z" },
    { url = "https://files.pythonhosted.org/packages/c7/30/0d09e6dddc6b7a7230ac77f50254b5980ab4fcd22976f72f8cc8a0404458/wrapt-2.3.0-cp314-cp314-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:1d6159c9b2fefec02314e1332dbbbfaf960e369dfd26bcf7f8b258b5732065b3", size = 160932, upload-time = "2026-07-28T06:05:27.022Z" },
    { url = "https://files.pythonhosted.org/packages/2c/ca/0913af0d2ec0c43865d32d615f518fea66c13c5c930e489e9b0de248e9a8/wrapt-2.3.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:24da48596326ef8e448cfa837b454f638713d3531262375f00e5a9681682fc07", size = 169017, upload-time = "2026-07-28T06:05:28.501Z" },
    { url = "https://files.pythonhosted.org/packages/c3/f2/3d1e47ea81b822210f5df1bf942fd90780a75c055243d569b664529dea88/wrapt-2.3.0-cp314-cp314-musllinux_1_2_riscv64.whl", hash = "sha256:cd3a2edf0427013736b8127955cec62608c56e53ea47e82812ea32059cda407f", size = 159065, upload-time = "2026-07-28T06:05:30.01Z" },
    { url = "https://files.pythonhosted.org/packages/43/a5/ef2066ced8e5fca204e2b361e9708e36555b40949c583d997ea3b590817d/wrapt-2.3.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:4fa0df3bff4e7ce45759f33fd39335fe2f60477bb9ecf7b8aa41e7d07ee36a23", size = 168821, upload-time = "2026-07-28T06:05:31.649Z" },
    { url = "https://files.pythonhosted.org/packages/d5/e1/016104650d4e572fa91506eb396b3dd8efbccc9284fdc1c9479c3d21db28/wrapt-2.3.0-cp314-cp314-win32.whl", hash = "sha256:2935d5454b3f179a29b12cf390ee47246740ba2c3a7545b1b46ba31a5f2a4a0b", size = 78700, upload-time = "2026-07-28T06:05:33.391Z" },
    { url = "https://files.pythonhosted.org/packages/3d/97/6fdc20a9f2ca304748b3f0819cbf377d55260562777bf0b615431bc3c181/wrapt-2.3.0-cp314-cp314-win_amd64.whl", hash = "sha256:cc2cea812e5cb179a796b766747e7d3b21088760d8deb95676d482b8c8e6fa7d", size = 81422, upload-time = "2026-07-28T06:05:34.774Z" },
    { url = "https://files.pythonhosted.org/packages/5e/a4/9cbd53bf05746bea2c392af39cb052427a8ec95cbd494d930733d8f44681/wrapt-2.3.0-cp314-cp314-win_arm64.whl", hash = "sha256:22cc5c0a717bd4da87018ae0bffd4c19c6fb679d3ff357216ba566ab26c76cab", size = 80639, upload-time = "2026-07-28T06:05:36.228Z" },
    { url = "https://files.pythonhosted.org/packages/43/bb/6c5e4a0f66ea0d2b2dd267e8dd05a0014eea56840b3c8595d40b0a5d1f91/wrapt-2.3.0-cp314-cp314t-macosx_10_15_x86_64.whl", hash = "sha256:a6b5984cd65dd639546f0eb4b8eacf1c31cb2fe9fb5c27bffe240987cdb2cf84", size = 84030, upload-time = "2026-07-28T06:05:37.714Z" },
    { url = "https://files.pythonhosted.org/packages/6a/eb/a1aedf03283bc9cbf8a1783995ddc54e3c5a86878f19002d2c428494f4c5/wrapt-2.3.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:c88abcf53daef80e01a75c7530e727fa6e2c1888fe83e3dcdba4c96216a1f5c7", size = 84419, upload-time = "2026-07-28T06:05:39.131Z" },
    { url = "https://files.pythonhosted.org/packages/63/61/50d511c0dc5105563849e86daa3e16ac7feef699f79fb05af45ea70107d5/wrapt-2.3.0-cp314-cp314t-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:85de890ff968196e92dd1ae73a9fb8970495e7650a457b1c9ef0ac3dd550bce2", size = 207171, upload-time = "2026-07-28T06:05:40.69Z" },
    { url = "https://files.pythonhosted.org/packages/3f/59/9b538cf7795217e810699d16bc88b96a830d9b5c403eb2ec2db6b5f2ae81/wrapt-2.3.0-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:50f416b74d092bb9f41b424e90dd457f365f7ba4b11de62a23679769a21bd85c", size = 214329, upload-time = "2026-07-28T06:05:42.287Z" },
    { url = "https://files.pythonhosted.org/packages/b3/28/9935d62b1499e5c8b3d191e99ba4eb31ca237a0b699142011a837e9dc7ea/wrapt-2.3.0-cp314-cp314t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:39febbee6d77301d31da6996b152ce52452da7c7ef72aba10c2fa976dff9c295", size = 199079, upload-time = "2026-07-28T06:05:43.958Z" },
    { url = "https://files.pythonhosted.org/packages/2b/01/4446b80fa2ffa47a3449b250d004ba1c1937f07f64a179608fec735df866/wrapt-2.3.0-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:93513bec052c6cd987f9f580c3df068c8bc4ebae6543736be3ca7ec5959cafcd", size = 209992, upload-time = "2026-07-28T06:05:45.677Z" },
    { url = "https://files.pythonhosted.org/packages/d4/07/56f26c9f9979586a021e8148747004aba4498f49458c90b0502969b904e1/wrapt-2.3.0-cp314-cp314t-musllinux_1_2_riscv64.whl", hash = "sha256:729126e667da34d251b8ebf8a45ef0c5ddadc21542b3d6e1abf4259ece6508df", size = 196334, upload-time = "2026-07-28T06:05:47.608Z" },
    { url = "https://files.pythonhosted.org/packages/8b/41/6d7bcc895b0f28b2250e10908f060687b9165429dcd7f22ddb3d4c031b74/wrapt-2.3.0-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:626b69db2021aa01671ec7bbc9740e558522bd44c18cf2ce69bf3d666a014109", size = 202644, upload-time = "2026-07-28T06:05:49.183Z" },
    { url = "https://files.pythonhosted.org/packages/cd/25/7860927edba06b758b8852a6f02e832be715563c67a6795d94350bc81099/wrapt-2.3.0-cp314-cp314t-win32.whl", hash = "sha256:629d73378082c00a8173031f9fb30a3ac6abbc894a5bfdfae71fabc60642d501", size = 79685, upload-time = "2026-07-28T06:05:50.976Z" },
    { url = "https://files.pythonhosted.org/packages/c4/0f/270bafe92fde3b069a39bc01e39ee79340895b335640df861d43d2a51885/wrapt-2.3.0-cp314-cp314t-win_amd64.whl", hash = "sha256:42869085687f0aefd57c0f636c3f9354f8ffb321a8ba9cb52d19beb796e561c5", size = 83104, upload-time = "2026-07-28T06:05:52.405Z" },
    { url = "https://files.pythonhosted.org/packages/55/b3/af176d79a8515a8a720eccdad9a96f6e31a30abf2865430c8c42adf2fd13/wrapt-2.3.0-cp314-cp314t-win_arm64.whl", hash = "sha256:b1e5aa486e269b00ed35e64771c7d0ab8096cfd2643405ca8cd60ebedc099a51", size = 81774, upload-time = "2026-07-28T06:05:53.902Z" },
    { url = "https://files.pythonhosted.org/packages/00/39/3daf9f47be208606586de4568ba6713db53ebc8fd7a575aea1fe57983b69/wrapt-2.3.0-py3-none-any.whl", hash = "sha256:d8c7ed08477429752b8c44991f40ad7838b18332a160698740a6bfbc10d998a2", size = 61866, upload-time = "2026-07-28T06:06:12.9Z" },
]

[[package]]
name = "xxhash"
version = "3.8.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/8e/63/71aa56b151a1b28770037a61bd4e461c2619cfc8866a4fcaf1548605e325/xxhash-3.8.1.tar.gz", hash = "sha256:b0de4bf3aa66363552d52c6a89003c479911f12098cd48a53d44a0f7a25f7c46", size = 86223, upload-time = "2026-07-06T10:49:58.937Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/42/91/f65c34a7aa7b4e7cf4854f8e6ef3f7ee32ceac41d4f008da0780db0612f6/xxhash-3.8.1-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:e6e49370822c1f4d8d90e678b06dbcb08b51a026a7c4b55479e7d467f2e813bc", size = 34680, upload-time = "2026-07-06T10:44:40.932Z" },
    { url = "https://files.pythonhosted.org/packages/57/04/b10a245a4c09a9cfa88f8e9ae755029413ad1ac17047f9a61906e5ae0799/xxhash-3.8.1-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:220d68130f83f7cc86d6edfdeab176adc73d7200bf3a8ec10c629e8cf605c215", size = 32397, upload-time = "2026-07-06T10:44:42.196Z" },
    { url = "https://files.pythonhosted.org/packages/3a/75/45ab795b5945b6388583bd75202106af505537935566c15a1577797a0e08/xxhash-3.8.1-cp312-cp312-manylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:4d365ee1892c1fa803536f8c6ce21d24b29c9718ec75eb856095c07830f8c478", size = 220549, upload-time = "2026-07-06T10:44:43.603Z" },
    { url = "https://files.pythonhosted.org/packages/13/44/5ba2bd0a14ddf4193fc7d8ec29625f659f22c06d60b28f04bf46305d8330/xxhash-3.8.1-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:852bfe059720632e2f16a6a4745e41d20937b2bf2a42a401e2412046bb6971cc", size = 241186, upload-time = "2026-07-06T10:44:45.534Z" },
    { url = "https://files.pythonhosted.org/packages/23/32/c4147def4d1e4538b906f82731e0ba23424377fc50a7cddd03cd284c8f63/xxhash-3.8.1-cp312-cp312-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:2f8c25a7061d952de589bd0ea0eaadee32378ff83dd6a677b267f9cd86f401f8", size = 264852, upload-time = "2026-07-06T10:44:47.199Z" },
    { url = "https://files.pythonhosted.org/packages/6c/bd/71ed14f4f0318bb7fd7b2ec51999413487fa8da8d41208e84d50d1ef0f98/xxhash-3.8.1-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:868a8dcaff1a84ba78038e1cef14fc88ccf84d9b4d12ea604696e0693296aa56", size = 242663, upload-time = "2026-07-06T10:44:48.846Z" },
    { url = "https://files.pythonhosted.org/packages/91/09/70af22c565a8473b3f2ae73f88e7721af281bc4a575236dbd1970c9f76f6/xxhash-3.8.1-cp312-cp312-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:6536d8677d2fff7e64cd0b98b976df9de7aee0e69590044c2af5f51b76b7a170", size = 473510, upload-time = "2026-07-06T10:44:50.695Z" },
    { url = "https://files.pythonhosted.org/packages/18/96/34db781c8f0cf99c544ca1f2bc2e5bf55426e1eb4ca6de8ea5da56a9f352/xxhash-3.8.1-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:82c0cedd280eab2e8291270e6c04894dbc096f8159a39dcf1807429f026ca3cc", size = 220469, upload-time = "2026-07-06T10:44:52.422Z" },
    { url = "https://files.pythonhosted.org/packages/93/5f/9a184f615fa5a4dce30c01534f62946ce5a11ce40f73785cbd356ccabaa9/xxhash-3.8.1-cp312-cp312-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:daa86e4b68221d38e669bb236ba112d0335353829fb627c82e5909e4bbe8694c", size = 310290, upload-time = "2026-07-06T10:44:54.142Z" },
    { url = "https://files.pythonhosted.org/packages/a9/dc/9b9a9789011ee153723a5eb9e7dd7fcbae2ba9b3fe7a729249ca7c252056/xxhash-3.8.1-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:2bc7113e6f2b6b3922dd61796ca9f36af09da3773898e7003038dc992fc83b8d", size = 238173, upload-time = "2026-07-06T10:44:55.693Z" },
    { url = "https://files.pythonhosted.org/packages/ec/4d/71c6005ada9dcb608a4e1902e8475ecadb5f3fbfa04e1e244d276a2d0c43/xxhash-3.8.1-cp312-cp312-musllinux_1_2_armv7l.whl", hash = "sha256:5eed32dad81d6ba8e62dc7b9ffa0500199385d7810a8dd9d4eafaceb8c6e20bb", size = 269026, upload-time = "2026-07-06T10:44:57.424Z" },
    { url = "https://files.pythonhosted.org/packages/2f/87/d6c036ba25dfbd9c8633be5aa86fc9474bbb9e2c68212a841d090abe7344/xxhash-3.8.1-cp312-cp312-musll
```

```
inux_1_2_i686.whl", hash = "sha256:83697b0ea1f10e7f5d8b26a4906fa851393c61546c63839643a2b7fe2d868061", size = 224970, upload-time = "2026-07-06T10:44:5
9.085Z" },
    { url = "https://files.pythonhosted.org/packages/48/62/4c1f035a41c5752aa05e195b6c904c07b94fe9061a16de61e72a6e6b135f/xxhash-3.8.1-cp312-cp312-musll
inux_1_2_ppc64le.whl", hash = "sha256:36fc69160465ae75c6ec4ac9f781bb2aa16ae7ff869e73c26fee85fbb11b9887", size = 240820, upload-time = "2026-07-06T10:4
5:00.746Z" },
    { url = "https://files.pythonhosted.org/packages/da/14/d39d565069b87e86d21a2af2a31d04db79249d25aa8d5b62959056a89857/xxhash-3.8.1-cp312-cp312-musll
inux_1_2_riscv64.whl", hash = "sha256:445e0f5a31f2f3546ae0895d4811e159518cdc9d824c11419898d40cfadb677e", size = 300619, upload-time = "2026-07-06T10:4
5:02.716Z" },
    { url = "https://files.pythonhosted.org/packages/13/22/75467acc887edc8cf71c97ab1708feb3df7a88bda589b9f399765c6387d2/xxhash-3.8.1-cp312-cp312-musll
inux_1_2_s390x.whl", hash = "sha256:dfe0580fbfd5e4af87d0cc52d2044f155d55ebd8c8a93568758a2ea7d8e15975", size = 443267, upload-time = "2026-07-06T10:45:
04.653Z" },
    { url = "https://files.pythonhosted.org/packages/a4/b6/1da3baa5fa6ef705e3425fddd382be7dfc4dfba2686df90a20f16e9c7b1b/xxhash-3.8.1-cp312-cp312-musll
inux_1_2_x86_64.whl", hash = "sha256:095e1323fa108be1292c54c86da3ef3c7a7dc015b105a52133973bc07a6ad11a", size = 217338, upload-time = "2026-07-06T10:45
:06.304Z" },
    { url = "https://files.pythonhosted.org/packages/78/dd/b5295a9f97484e7a1c2b283a742ca45e3104991c55a1ef670dde161829ba/xxhash-3.8.1-cp312-cp312-win32
.whl", hash = "sha256:bf28f55e427e0483acb1f666bd0d869b6d5e5a716680c216ad7befe3d4cfba2e", size = 31970, upload-time = "2026-07-06T10:45:07.823Z" },
    { url = "https://files.pythonhosted.org/packages/ec/31/3fa0b807d7e21515cd975e7fe5c039d52ac3e9401a96d6ad68dae6305215/xxhash-3.8.1-cp312-cp312-win_a
md64.whl", hash = "sha256:2256e80e4960ee282f63428adb349cb7f8bd8efe4db770d88eb815f4b9860724", size = 32741, upload-time = "2026-07-06T10:45:09.42Z" },
    { url = "https://files.pythonhosted.org/packages/b8/05/86feada74e239600e6875aa507afb40482a89b92700aa74a92da83bdcb77/xxhash-3.8.1-cp312-cp312-win_a
rm64.whl", hash = "sha256:9df56e6df96a60590935e22373041cccc91fd55858763dcffb55bf63b3a2b396", size = 29234, upload-time = "2026-07-06T10:45:10.809Z" },
    { url = "https://files.pythonhosted.org/packages/6b/8c/446bb782cd0d27007a917b5569a08dd73219c3e8d6e459014db104b27bdb/xxhash-3.8.1-cp313-cp313-andro
id_21_arm64_v8a.whl", hash = "sha256:3c682fcd96eb4bf64be32a4d95f96107e1588005831bd8a741b324fdda01b913", size = 38562, upload-time = "2026-07-06T10:45:
12.425Z" },
    { url = "https://files.pythonhosted.org/packages/d7/ec/c0c45627eaa6be7a5d6117423adf8f7a15b17ee74b4b17072cca5959a225/xxhash-3.8.1-cp313-cp313-andro
id_21_x86_64.whl", hash = "sha256:036a024d8b9c01f70782e09ed98d532e76fd23f950ae7154bd950fe94e90ebec", size = 36656, upload-time = "2026-07-06T10:45:13.
932Z" },
    { url = "https://files.pythonhosted.org/packages/f6/94/8324c04cc7597154caaeba6c094e01fbd2e7601d01e7a13eea9f5420e77b/xxhash-3.8.1-cp313-cp313-ios_1
3_0_arm64_iphoneos.whl", hash = "sha256:d6a5c0bce213b23b0166fe0d35bcbbe23ce4b968f257cc7eb6fd57cb8e1e6297", size = 31169, upload-time = "2026-07-06T10:
45:15.687Z" },
    { url = "https://files.pythonhosted.org/packages/40/a4/beb6bb26e1184e126dbe7a5682330214ef54dcfbf882078aa9f4b5428d42/xxhash-3.8.1-cp313-cp313-ios_1
3_0_arm64_iphonesimulator.whl", hash = "sha256:5177aa44eddaa97c6ef0cc00c6d540edb64d51781d2f8fb941612ec61a92c9ed", size = 32177, upload-time = "2026-07
-06T10:45:17.035Z" },
    { url = "https://files.pythonhosted.org/packages/56/0f/fc4c92a5a528f839b34b6419b2e53c8597f2a629d5a1f5d721f65bfa1fd6/xxhash-3.8.1-cp313-cp313-ios_1
3_0_x86_64_iphonesimulator.whl", hash = "sha256:7801b7223db017b9c0c9ccf37e44524edb35a1544a1c032add22c061c6af0276", size = 34642, upload-time = "2026-0
7-06T10:45:18.39Z" },
    { url = "https://files.pythonhosted.org/packages/d4/58/edbfb141d4000767ac6a9694f8ac0763e2c2e983e65c9e31620ba56e2667/xxhash-3.8.1-cp313-cp313-macos
x_10_13_x86_64.whl", hash = "sha256:9e80238259655bf69d7bcd08226a970d7f42605f3157786bfa76dd13472d7fa0", size = 34684, upload-time = "2026-07-06T10:45:2
0.033Z" },
    { url = "https://files.pythonhosted.org/packages/07/3f/5072f1f0f5714186f0ac2a0b5a4929ce30d4b845e94886b6c01b6ebda0be/xxhash-3.8.1-cp313-cp313-macos
x_11_0_arm64.whl", hash = "sha256:bcab50a389cc04d87f90092af78a6adba2ab3deca63175a3344ca83514045315", size = 32401, upload-time = "2026-07-06T10:45:21.
414Z" },
    { url = "https://files.pythonhosted.org/packages/49/c7/802ea2f9c2ed59219934d6d65c470d502b1788043eae277a52af8658bda6/xxhash-3.8.1-cp313-cp313-manyl
inux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:a2489d3a776fa380cb8e71f54c7fda268a9baf3de9b1395093fd280f95735907", size = 22061
7, upload-time = "2026-07-06T10:45:23.234Z" },
    { url = "https://files.pythonhosted.org/packages/99/a8/e10488efd31fcb13fcd6acbc6e788f10c6f8e3a0cc4ae3eb89dc19c55a12/xxhash-3.8.1-cp313-cp313-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:32ab1e5432690276e71192be7401b55f96db2d0eedea5d44eb1f164505669cc0",
 size = 241295, upload-time = "2026-07-06T10:45:25.364Z" },
    { url = "https://files.pythonhosted.org/packages/18/cc/14180b17d44892a631f8ae7323c30bfbb1328efc8209e528a480293528ac/xxhash-3.8.1-cp313-cp313-manyl
inux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:b30e01a0b97a4bc3f519a4d7a82da3dc53251fb0de5eeea8660dcd4ff094c0c2", si
ze = 264688, upload-time = "2026-07-06T10:45:27.09Z" },
    { url = "https://files.pythonhosted.org/packages/a9/72/a14019d0c5f6c41ee407a503036ae32787c91325ca218a96a9b5627be651/xxhash-3.8.1-cp313-cp313-manyl
inux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:1f44275ddb0978b67a58a951501903f04d49335a91f7681c9ce122ecb8ccb329",
 size = 242740, upload-time = "2026-07-06T10:45:28.753Z" },
    { url = "https://files.pythonhosted.org/packages/68/08/92550e556c6fcfcb96c6a336945eb53a431ed43120ed749636debb16c5cf/xxhash-3.8.1-cp313-cp313-manyl
inux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:e3b87cbd974512c0c5fc7b469c36b2cdc9ee6d76e4ec78bccb2c7184611c49b0", size
= 473599, upload-time = "2026-07-06T10:45:30.524Z" },
    { url = "https://files.pythonhosted.org/packages/29/83/e361d3c1acd1b21e1d489616de6fa4aaf843365d8179f612e3743eac20a9/xxhash-3.8.1-cp313-cp313-manyl
inux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:98ee81b4b7f3023c9cb04a78cc67610baffcb5812d92f2096cb5a5efc6f19437", si
ze = 220559, upload-time = "2026-07-06T10:45:32.979Z" },
    { url = "https://files.pythonhosted.org/packages/05/01/006a4243c2c2a6831827f9999f6d1c23feeef100eb023c1f886022a00bf3/xxhash-3.8.1-cp313-cp313-manyl
inux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:2666f059a1588a99267e33605365ed89cea92f424b3522806a9f4bd8ad2e3d62", size = 310383, upload
-time = "2026-07-06T10:45:35.875Z" },
    { url = "https://files.pythonhosted.org/packages/d8/20/af388e8bf9f9a0f89eeef7d2a1935d176ee1c20bc6adeda05035879379cf/xxhash-3.8.1-cp313-cp313-musll
inux_1_2_aarch64.whl", hash = "sha256:b0093cf7eeb91b84776e8742113afa4bdf47533d36cf719179aaaf1f56f6f8bf", size = 238228, upload-time = "2026-07-06T10:4
5:38.02Z" },
    { url = "https://files.pythonhosted.org/packages/63/6b/4666579a87eebd1744663c404297355fa0658617b015cedfa58810ee7036/xxhash-3.8.1-cp313-cp313-musll
inux_1_2_armv7l.whl", hash = "sha256:3a800912a2e5e975d4128969d645c4a2a80aa886ccd6c9b1c6f44529e327e8cf", size = 269137, upload-time = "2026-07-06T10:45
:39.954Z" },
    { url = "https://files.pythonhosted.org/packages/de/d3/e963a8a46f900a137d91b02144d8ea07a8f812971b138204a3b2f8b8e55c/xxhash-3.8.1-cp313-cp313-musll
inux_1_2_i686.whl", hash = "sha256:0fe37f72a207223d22a4eddc3149d4298993385aa9daef25c039246ca5a309f3", size = 225068, upload-time = "2026-07-06T10:45:4
1.718Z" },
    { url = "https://files.pythonhosted.org/packages/aa/80/9d181dbcde4b0fe48375f48833a5832d4b8cd2b349b15110c92ee472d874/xxhash-3.8.1-cp313-cp313-musll
inux_1_2_ppc64le.whl", hash = "sha256:5db43f249b4be9f99ef4b967863f37094fb40e67effafb78ba4f0356b6396104", size = 240874, upload-time = "2026-07-06T10:4
5:43.414Z" },
    { url = "https://files.pythonhosted.org/packages/39/15/ce3ab5a1cd27ead25a5196e55a7284220f6ad6e316da494ffd900b2b600f/xxhash-3.8.1-cp313-cp313-musll
inux_1_2_riscv64.whl", hash = "sha256:c4ed42965c2cd9081f011be22f69d0e65d3b6165fe7734072fd0c232840bbd4e", size = 300702, upload-time = "2026-07-06T10:4
5:45.135Z" },
    { url = "https://files.pythonhosted.org/packages/96/c0/2281a8ab5f2a62dbf57a23c58a01ccc1d98abf40f71193c8a81f59e759b5/xxhash-3.8.1-cp313-cp313-musll
inux_1_2_s390x.whl", hash = "sha256:3557bec8fcb11738a8920eeb68974bc76b75262f6947998d3147954ce0a4b893", size = 443351, upload-time = "2026-07-06T10:45:
47.188Z" },
    { url = "https://files.pythonhosted.org/packages/81/2e/071a58c1a53a52d4f7a3aa0987be0c396dffd40da8204805fe1b130a81f4/xxhash-3.8.1-cp313-cp313-musll
inux_1_2_x86_64.whl", hash = "sha256:00de40f3b42240db23a82a5c682b55d7263d84a26a953240c1aee463409660e3", size = 217396, upload-time = "2026-07-06T10:45
:48.925Z" },
    { url = "https://files.pythonhosted.org/packages/68/44/36ab58134badd9d3433fc7b53c4ca8d113d8e807782885628640f8297a4d/xxhash-3.8.1-cp313-cp313-win32
.whl", hash = "sha256:b5196cc2574cfec572a5f3fb7cfa5ade27305ae3d06516a082132441aff4c83a", size = 31974, upload-time = "2026-07-06T10:45:50.591Z" },
    { url = "https://files.pythonhosted.org/packages/96/2a/2a0b84798448e766f7b89ceed073cb0cb5a43fc9ebbacbdea74a38de18e3/xxhash-3.8.1-cp313-cp313-win_a
md64.whl", hash = "sha256:538f5f865df6cd8c32dd63158a0e5b4f5dd08d732a7da8b7228a5a0776c8ce55", size = 32739, upload-time = "2026-07-06T10:45:52.221Z" },
    { url = "https://files.pythonhosted.org/packages/d4/60/bb51dbf7c363ff88a7cbd50b7959718219577ef44d7cf255929ffc4a2194/xxhash-3.8.1-cp313-cp313-win_a
rm64.whl", hash = "sha256:a6617f30641ba0d8baa1635fbefb1dffc5165ec36d26921bd5cee13497cd937a", size = 29239, upload-time = "2026-07-06T10:45:53.714Z" },
    { url = "https://files.pythonhosted.org/packages/56/d3/827ca123c2ee5443a6aaed3c5dd199237dc2f010e2bebd7ec09ef36f3a5f/xxhash-3.8.1-cp313-cp313t-maco
sx_10_13_x86_64.whl", hash = "sha256:bfcd82852c62a60e314670a9602de354c4460f8adad916e2e42a20860c7870bc", size = 34964, upload-time = "2026-07-06T10:45:
55.535Z" },
    { url = "https://files.pythonhosted.org/packages/05/67/67ae2a3ccdeb8b8ef025d35aee9edd1d26c3abe5051d47da9286232afbf8/xxhash-3.8.1-cp313-cp313t-maco
sx_11_0_arm64.whl", hash = "sha256:08ea2081f5e88615fec8622a9f87fbe21b8ea58d88cfc02163ca11026ee62a92", size = 32697, upload-time = "2026-07-06T10:45:57
.288Z" },
    { url = "https://files.pythonhosted.org/packages/38/5a/3d3994346e1f45493679cb5c1ffc2bf454e410e9d1e8a662d253becee91e/xxhash-3.8.1-cp313-cp313t-many
linux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:2e32855b6f9e5b18f449e59d45e3d5778bdeb660632ef2693cca267a11246c75", size = 2259
54, upload-time = "2026-07-06T10:45:58.897Z" },
    { url = "https://files.pythonhosted.org/packages/3f/2c/53169270309b7cd8e05504e07fe123bac053b89d00ac63617faacf0a2ec0/xxhash-3.8.1-cp313-cp313t-many
linux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:a6e088bd7870775624256a0d84c2a6714afd223b2eeb56b0ca58398e52a32fda"
, size = 249776, upload-time = "2026-07-06T10:46:00.977Z" },
    { url = "https://files.pythonhosted.org/packages/70/e0/5c551d8d592f944506f7c5185e210255c15e672a3c6008c156a1bd9b775e/xxhash-3.8.1-cp313-cp313t-many
linux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:72eb5ae575cc7ae2b23f6f8064a8b10f638c7149819ae9cc6d20ebd4d37a1629", s
ize = 274776, upload-time = "2026-07-06T10:46:02.869Z" },
    { url = "https://files.pythonhosted.org/packages/a0/2a/d3a762270cee2d7bcd0e25e28c623e5f3f5c0dc637b66e3e47dd5b0bb3f0/xxhash-3.8.1-cp313-cp313t-many
```

```
linux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:d0b48cdf690a64cedf7258c3dc9506cc41fc86edd7739c40e3098952265dc068"
, size = 252056, upload-time = "2026-07-06T10:46:04.688Z" },
    { url = "https://files.pythonhosted.org/packages/c1/8f/b78e4373b2cb6d1c42af60ea2d7e9146ad0710b239ac7f706d5d31d5bb98/xxhash-3.8.1-cp313-cp313t-many
linux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:fb9e256a357dfcede7818c6d34e70db2d6b664394803d1de4b6984d2de76c0f1", size
 = 482108, upload-time = "2026-07-06T10:46:06.498Z" },
    { url = "https://files.pythonhosted.org/packages/e6/0d/642d923336ea61a15f8ce64fc7e078729e6e06c3a026e517fa79b2c23b7a/xxhash-3.8.1-cp313-cp313t-many
linux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:51f71a6e2ad071e70c937e41fcb6c19f82c3f9f49831eba850ed4a106ffbb647", s
ize = 226739, upload-time = "2026-07-06T10:46:08.598Z" },
    { url = "https://files.pythonhosted.org/packages/a6/0a/a37d6da6427d45a8d23e3ee3a0ca9c9d4a90364849c6637fe2963a755f9b/xxhash-3.8.1-cp313-cp313t-many
linux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:e4a6443968c4e8dc69967e12776776a5952c119cc1bd94168ad1c5ad667c2be1", size = 319658, uploa
d-time = "2026-07-06T10:46:10.504Z" },
    { url = "https://files.pythonhosted.org/packages/4a/51/ebbd40da8a3f1bc53b4b7a9a87f8e28bd95c5f21bc14b8a57860cf367d1b/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_aarch64.whl", hash = "sha256:714503083a1f2065c9ad15340dd49ac8a8e948a505a705ffa1750cb951519113", size = 246059, upload-time = "2026-07-06T10:
46:12.634Z" },
    { url = "https://files.pythonhosted.org/packages/24/4c/d9014030147e1f0bb26e7da47aa240dd9ec61c763c573e558111d869f8e1/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_armv7l.whl", hash = "sha256:77f74e45a1e5574bbbf80181c8027b3a4c65c2248fffbd557bd596fff13102f9", size = 275535, upload-time = "2026-07-06T10:4
6:14.614Z" },
    { url = "https://files.pythonhosted.org/packages/84/86/caee2db41fadcd5a25aa4323213f9afec5a8586d4e419241e3d659362bd7/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_i686.whl", hash = "sha256:4e0e1b0fb0259c1b75d1251ac0bb4d7ab675d36f7a6bf4ba6aa630dae94f9ffa", size = 231292, upload-time = "2026-07-06T10:46:
16.452Z" },
    { url = "https://files.pythonhosted.org/packages/0b/60/f52f08bcdc904c4514ea5c25caa19e9f3214144434a6ff96dc82dc1cbddd/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_ppc64le.whl", hash = "sha256:10e4393ec33633c2f05ad01869e546ad080b1a18f2650503731f153774608b31", size = 250490, upload-time = "2026-07-06T10:
46:18.318Z" },
    { url = "https://files.pythonhosted.org/packages/24/a0/94dc7ae310838f250669c6ad7168e6d6fca17d49dac1053f06dc232c4a56/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_riscv64.whl", hash = "sha256:b3ba794c3d885803db6c3116686923f1ec13bc86e621e169a375282b63ea1cc6", size = 309861, upload-time = "2026-07-06T10:
46:20.503Z" },
    { url = "https://files.pythonhosted.org/packages/8b/f9/adeead7d0eb28cdfc2832544ea639ffbc6749ccde47a8e228d667459182e/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_s390x.whl", hash = "sha256:57189a69c0891e4818853feaa521c972d22c880a001453addea015f48e3c3398", size = 448739, upload-time = "2026-07-06T10:46
:22.79Z" },
    { url = "https://files.pythonhosted.org/packages/04/a4/22ec0e07db57d901c9298ae98aa3cf2be45bafded6f07c13131e85b89032/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_x86_64.whl", hash = "sha256:d59e71153fe9ff85648d00e18649b07e9b22c797291abb7e27274fa06df8b838", size = 223657, upload-time = "2026-07-06T10:4
6:24.831Z" },
    { url = "https://files.pythonhosted.org/packages/94/32/8a9531f37b59e5a013003db7cb7414baf4ce7e0e1268e0d5947cd3d6a2df/xxhash-3.8.1-cp313-cp313t-win3
2.whl", hash = "sha256:5b96f0024e9840f449bd91b2d005c921a4b666055a0d1b6492463799f32aae22", size = 32377, upload-time = "2026-07-06T10:46:26.86Z" },
    { url = "https://files.pythonhosted.org/packages/e7/ab/2ca45fd7f671de5f81fc297ef1c95080b40c86ec6be0cc6034b8f7707ac8/xxhash-3.8.1-cp313-cp313t-win_
amd64.whl", hash = "sha256:37d5a56c36dcc0b9a87b814cd992598d33863ff683749de6c86081f278d5e629", size = 33274, upload-time = "2026-07-06T10:46:28.39Z" },
    { url = "https://files.pythonhosted.org/packages/5a/54/20d7163463ddb6438b73a427d1655a77a502cf9b9b0c3ada3599629d9c0a/xxhash-3.8.1-cp313-cp313t-win_
arm64.whl", hash = "sha256:6696c8752aded28ff3b16f33ef28ce28fb5d209b80c206746f943199fcf5fd65", size = 29375, upload-time = "2026-07-06T10:46:29.962Z" }
,
    { url = "https://files.pythonhosted.org/packages/c2/8b/df2ba04f22a6cd6b39f96a6577329a8471a55c90ef8d8e2f7c102363613f/xxhash-3.8.1-cp314-cp314-andro
id_24_arm64_v8a.whl", hash = "sha256:9db455cb649dcfe4504d6d68a6d83a7315a99a3ca59871dc3ff840671f99adba", size = 38430, upload-time = "2026-07-06T10:46:
31.496Z" },
    { url = "https://files.pythonhosted.org/packages/b2/4f/6a059e8ad3ca8deedc91dfe335b21120490089515221 2c03ebbe721de68b/xxhash-3.8.1-cp314-cp314-andro
id_24_x86_64.whl", hash = "sha256:affb37f152e55b5e4494bb9d0107f7bb08515c6704fbed82d9f61214d74adc17", size = 36558, upload-time = "2026-07-06T10:46:33.
078Z" },
    { url = "https://files.pythonhosted.org/packages/cb/95/40be178205acce092ae418feb20ac737b32a02c7b864926ed0717354c9f8/xxhash-3.8.1-cp314-cp314-ios_1
3_0_arm64_iphoneos.whl", hash = "sha256:460261045936975193bfd20549a0de1cd52a33b405cbb972f0d80940c42266cd", size = 31181, upload-time = "2026-07-06T10:
46:34.793Z" },
    { url = "https://files.pythonhosted.org/packages/3f/89/2da4dbf051bafa156c0e3f12012db2b0ac3b84ff37ca1f021f6bfffcdfbb/xxhash-3.8.1-cp314-cp314-ios_1
3_0_arm64_iphonesimulator.whl", hash = "sha256:38c887aedb696ef8bca19983206d270848558cfae4a91afa6a2fb05dde58ffc5", size = 32192, upload-time = "2026-07
-06T10:46:36.393Z" },
    { url = "https://files.pythonhosted.org/packages/7c/4e/e000bbae3566bc8e0be771a8a0f294aa99075e3f0bc4ef43922ebffdebc8/xxhash-3.8.1-cp314-cp314-ios_1
3_0_x86_64_iphonesimulator.whl", hash = "sha256:594131ce1aad18db3689781f806db1b065cdaa04f4df36b4c038d2013aefd0bf", size = 34691, upload-time = "2026-0
7-06T10:46:38.1Z" },
    { url = "https://files.pythonhosted.org/packages/b4/4a/ea954aacc7d1c8711880ac2b55da94429a9b4296b151c4fc0966549ca1ee/xxhash-3.8.1-cp314-cp314-macos
x_10_15_x86_64.whl", hash = "sha256:78c794b643d214f1522e7a288bcf5a2de120d26cd170516749a4009dc92722c9", size = 34807, upload-time = "2026-07-06T10:46:3
9.647Z" },
    { url = "https://files.pythonhosted.org/packages/ca/29/df598e738ff37558ac627264deb2e560902d9bf7f46d3bd5175c9eee593e/xxhash-3.8.1-cp314-cp314-macos
x_11_0_arm64.whl", hash = "sha256:af0c9fedc4a2c24e8664953882fe8185f3790b8338c9c700f76f5ad660817711", size = 32410, upload-time = "2026-07-06T10:46:41.
359Z" },
    { url = "https://files.pythonhosted.org/packages/59/9c/81ab40e7d33ada0b3df5d1bc884894d15dbf4f805cd645b685e4606bb8e0/xxhash-3.8.1-cp314-cp314-manyl
inux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:115772daeb71b2f3b9381177017f53e6cf3f3439c840737fdabd21aba6e54920", size = 22056
4, upload-time = "2026-07-06T10:46:43.463Z" },
    { url = "https://files.pythonhosted.org/packages/fd/6f/62ae6f5c8606320a0e2a41c2dc8c6d91cc5d63d0f84dd9582e9543779dd8/xxhash-3.8.1-cp314-cp314-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:000435984a0469b0f822fe76f35bddea0f96a4d6521b3339a60a6428cdee1edc",
 size = 241462, upload-time = "2026-07-06T10:46:45.509Z" },
    { url = "https://files.pythonhosted.org/packages/15/a1/9c3a0ec6cb524396f551eddd102a76690a795494eb9784fc67542b0daa37/xxhash-3.8.1-cp314-cp314-manyl
inux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:2f1c68394818e0595569c2ff3cbc1e6d5a36a434e796f5c526b987b80c8a8c62", si
ze = 264491, upload-time = "2026-07-06T10:46:47.655Z" },
    { url = "https://files.pythonhosted.org/packages/64/f2/700a4674e4308eb59d2fdb973977e82eae231bea5044753fee5c9eec0e0c/xxhash-3.8.1-cp314-cp314-manyl
inux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:46b39976d008e2a845758650f0ff7136bca004f40da0c8798bd37ac37860154f",
 size = 242905, upload-time = "2026-07-06T10:46:49.857Z" },
    { url = "https://files.pythonhosted.org/packages/f3/8a/72d9874375c8d4cbc64a8cd1d659d5695a8765c3db82efa82dc5bd9f14d0/xxhash-3.8.1-cp314-cp314-manyl
inux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:d5006c65ec507a333479e76e00e2c368781f16c24ededa764763956b32a0e93e", size
= 473873, upload-time = "2026-07-06T10:46:51.953Z" },
    { url = "https://files.pythonhosted.org/packages/03/f0/6db07590ed7e0a77f186ef0bcea8d52553bf1ba57833e09467a2411f0f2d/xxhash-3.8.1-cp314-cp314-manyl
inux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:c31a2649bcf1fe97cf11c79848d761df33ac46b3896942d31b640557b486ff6b", si
ze = 220765, upload-time = "2026-07-06T10:46:55.41Z" },
    { url = "https://files.pythonhosted.org/packages/8f/10/00d12d8b8beabbf49a8bbc626fb9f40445145a8887eb41a6acfb69149ac4/xxhash-3.8.1-cp314-cp314-manyl
inux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:8f759eed402448c2bdbb492e4fba1f20668ffe29688605ea61f0f67f9e4e386d", size = 310478, upload
-time = "2026-07-06T10:46:57.729Z" },
    { url = "https://files.pythonhosted.org/packages/1f/f9/12a82394eefb0f185d15a7f7b9f627c61c475a72dd83718436a5b84b42ac/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_aarch64.whl", hash = "sha256:7b5f97ecfede10d5b2870383620e2d25c8561e217c7bf9081073802b54248d2b", size = 238393, upload-time = "2026-07-06T10:4
6:59.87Z" },
    { url = "https://files.pythonhosted.org/packages/20/f3/53f963e320b9ce678337aa7273f39ce692ded8b99e3d22a866ec722159ab/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_armv7l.whl", hash = "sha256:1da930bbcac3e8fbe2191850e2abb57977a99348c12c4b385e1058ac1b0a9ecc", size = 268704, upload-time = "2026-07-06T10:47
:01.806Z" },
    { url = "https://files.pythonhosted.org/packages/0a/50/5b5badbd87c82d9f9b5f58ac74a3f29ef08f6fc387b324b8fd482450b862/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_i686.whl", hash = "sha256:747476436f6891b9773374ce8d48edcc8b12cb5b61b67c6fb6289633747d088f", size = 225015, upload-time = "2026-07-06T10:47:0
3.784Z" },
    { url = "https://files.pythonhosted.org/packages/30/93/3ca68265afe7b4e69435e08a7b6a1d9d0f2a071e889da1f8041ed00fe878/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_ppc64le.whl", hash = "sha256:4ef09bbc2519a93cd0f95f2ceb5f7b85919dffea643278e02362bf40e3c4bed1", size = 240951, upload-time = "2026-07-06T10:4
7:05.816Z" },
    { url = "https://files.pythonhosted.org/packages/dd/a6/27e19670c40f46b5e76e11f2f4713d210548045684 25d870670e757172ad/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_riscv64.whl", hash = "sha256:a5eed9d41995a83f3332b4e3396abb7f433cac584222bd7e305b606d8353861e", size = 300751, upload-time = "2026-07-06T10:4
7:07.95Z" },
    { url = "https://files.pythonhosted.org/packages/bc/fb/b33e27689959fe7ed2ae0b830af41560d65213943983afa9db3a8d481bce/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_s390x.whl", hash = "sha256:53f3ed9118397074ff63a79b66b7fec1c84c782eecde35c5bc94e420a971c231", size = 443480, upload-time = "2026-07-06T10:47:
10Z" },
    { url = "https://files.pythonhosted.org/packages/26/60/0e0d973be5fe280753ef02fbc89349492ad6e903bf1dcb870b668f94b662/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_x86_64.whl", hash = "sha256:d247b34bf433c92b41689318fd25d246313cab2275a6a47e2efac178b80d6efe", size = 217657, upload-time = "2026-07-06T10:47
:12.196Z" },
    { url = "https://files.pythonhosted.org/packages/ad/68/c9e3ecef4a9a417d464cb5bd200aa12f73192dee677901b9e08e0ad0d1bb/xxhash-3.8.1-cp314-cp314-win32
.whl", hash = "sha256:d58ce8b6cfa9c4d2f230557f69caf7c06369e318015d0b19485095bc2c5963ab", size = 32690, upload-time = "2026-07-06T10:47:14.204Z" },
    { url = "https://files.pythonhosted.org/packages/d7/99/e9e44588c0b62837bbec5ba7927816de0afa03406b1a0b6c7a7e1d1a30a0/xxhash-3.8.1-cp314-cp314-win_a
```

```
md64.whl", hash = "sha256:6cee733fe4ccb1737e0997135283c82341e5cfa9cf214b165f9087fb663aaf4f", size = 33460, upload-time = "2026-07-06T10:47:16.021Z" },
    { url = "https://files.pythonhosted.org/packages/45/2b/64f36d86380b3657ad9031967ab814f3ef31307174650853f69c18932ebc/xxhash-3.8.1-cp314-cp314-win_a
rm64.whl", hash = "sha256:58346024d47e84f7d8b3e7f5d6faa1d58acbbe49a8771497872059f58c1d8ea5", size = 30092, upload-time = "2026-07-06T10:47:17.81Z" },
    { url = "https://files.pythonhosted.org/packages/92/cb/18b64bff88c58a0ca209dc533e63cf02d7ae5aa6b1b9a9fd14e81b5dbd60/xxhash-3.8.1-cp314-cp314t-maco
sx_10_15_x86_64.whl", hash = "sha256:01cab782f8a0a05ecad2c63d7ef10f7ab475f660e0d6419d069418c14d88de7c", size = 35024, upload-time = "2026-07-06T10:47:
19.821Z" },
    { url = "https://files.pythonhosted.org/packages/af/1d/72d8a70520e5dcddb472ea0486d299da3240745a10658290cd7b5690ede2/xxhash-3.8.1-cp314-cp314t-maco
sx_11_0_arm64.whl", hash = "sha256:717b12fdc51819833704e85e6926d76981ffa3f780ef92e33ebb8b26d46bb230", size = 32697, upload-time = "2026-07-06T10:47:21
.649Z" },
    { url = "https://files.pythonhosted.org/packages/9c/b8/e041f555903c56db3d0a731b3d72a6575d75e0ed868b1bd2e5176111ca44/xxhash-3.8.1-cp314-cp314t-many
linux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:ec55d80e9b8a519d742669e0b49e8ce9e6747be42bf3c138158b6543a9c8e489", size = 2260
44, upload-time = "2026-07-06T10:47:23.612Z" },
    { url = "https://files.pythonhosted.org/packages/3a/7e/5cdcf06bf6ec4b5d2ac073feb23432ec1d603fd438864cbd2c09c7cb45e1/xxhash-3.8.1-cp314-cp314t-many
linux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:98d8ac1129b4dd39098cffed94d1284aceb61c3aa396757ccc736ac392e4cee5"
, size = 249899, upload-time = "2026-07-06T10:47:25.812Z" },
    { url = "https://files.pythonhosted.org/packages/c0/c0/eb7e059cb5e1dba11fd30d2fdf882f56e5a417a3eaa43669d43623767f45/xxhash-3.8.1-cp314-cp314t-many
linux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:3bc0fa90830df1e1277f33cc6e55de9990b83c0319fd8c7412866cfde38b025e", s
ize = 274892, upload-time = "2026-07-06T10:47:27.931Z" },
    { url = "https://files.pythonhosted.org/packages/66/74/a600aaf7cd39957fd1510adeedb1749c1e7eb82bd632a1153d9c664c3135/xxhash-3.8.1-cp314-cp314t-many
linux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:c73b6f652f0745425aa6378319c331293b5341756262e9408ed3d45f183375e6"
, size = 252243, upload-time = "2026-07-06T10:47:30.288Z" },
    { url = "https://files.pythonhosted.org/packages/ad/04/78d88fa75a6763e5d09bf1b947a392a27988903381b219006f92f3c68fc8/xxhash-3.8.1-cp314-cp314t-many
linux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:f6114692261eff4266386cdec0f7d87eee24e317ab397c218b7ae6a76b4c6339", size
 = 482191, upload-time = "2026-07-06T10:47:32.45Z" },
    { url = "https://files.pythonhosted.org/packages/7f/06/07a8aea1108d682de8791ce608cdf367d75ff4e7e57cd3c154bdc6f47b23/xxhash-3.8.1-cp314-cp314t-many
linux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:4df57c0b161ec1b3ed0526a67b0db0914b557e86ee8aae51887aec941b261542", s
ize = 226877, upload-time = "2026-07-06T10:47:34.705Z" },
    { url = "https://files.pythonhosted.org/packages/ed/b5/86bade5618a524d2c06c4041aa2fe8e5749ce16e88afba60d67c1684a21f/xxhash-3.8.1-cp314-cp314t-many
linux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:9043877a917be88ccf230aa5667c1bd059bce80f4c2727e4defa1b29b7f48b08", size = 319794, uploa
d-time = "2026-07-06T10:47:37.08Z" },
    { url = "https://files.pythonhosted.org/packages/23/69/9b1a2b89b1621bb740fbcb7beb512f60f99480c1bdc680c0c90e1f56ff75/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_aarch64.whl", hash = "sha256:559e3cabe522231909f9de98ef06929edbd53782046bd21aae0c72db6f2a0775", size = 246202, upload-time = "2026-07-06T10:
47:39.676Z" },
    { url = "https://files.pythonhosted.org/packages/08/ea/662ed6cb49f1d34078b6a3a3e0f3d29ff93fd7b5a03c0bc9ecfd9b2159c3/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_armv7l.whl", hash = "sha256:264710bd335016f303763ce1275c6486df30bb57c2245c91b224c983d7ac39b8", size = 275628, upload-time = "2026-07-06T10:4
7:41.99Z" },
    { url = "https://files.pythonhosted.org/packages/13/f5/49fc9e4c6728a5a3bd8fe639199d2fa67609b3a84f938aff6e8568dd3e4f/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_i686.whl", hash = "sha256:e14800b9b10bb39d7a60ad4a310e403164d7b8988a27ae933d4e40618a44088e", size = 231390, upload-time = "2026-07-06T10:47:
44.233Z" },
    { url = "https://files.pythonhosted.org/packages/64/9d/3acaf8f599c0e0b30e910a3a11ba32929da53c86dc73c7c55fe6a010b4e9/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_ppc64le.whl", hash = "sha256:ea6a3e734b0fd41b82784a400be946821900daebe610c050a5e0760838a34f99", size = 250600, upload-time = "2026-07-06T10:
47:47.611Z" },
    { url = "https://files.pythonhosted.org/packages/23/64/8acab4c5ec60dbe664b5b9858fd44c2413b07e535b09556a0a5022e78aa6/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_riscv64.whl", hash = "sha256:cf399fac542a1c7a4734a435b93df2c55e858c7d31abf6c1bdf46f9ae67fbfd0", size = 310032, upload-time = "2026-07-06T10:
47:49.88Z" },
    { url = "https://files.pythonhosted.org/packages/56/47/a0288d7329b1fe63e2734a32d19d444a96ae2b4810f545bc61e561224917/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_s390x.whl", hash = "sha256:44c89d915a75c11d2547eaee9098fcd80398987c4bff2974a0497a925bf92c07", size = 448882, upload-time = "2026-07-06T10:47
:52.631Z" },
    { url = "https://files.pythonhosted.org/packages/01/e7/3071dfd3beb5c38204ce1cf56bf7749fce08de900fa92714b81d1d8ca1f2/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_x86_64.whl", hash = "sha256:358650d5bda9c635da699c53adf4e8134af492ecc79c960f917eebf088bb6799", size = 223728, upload-time = "2026-07-06T10:4
7:55.093Z" },
    { url = "https://files.pythonhosted.org/packages/12/11/b99949f0ba2b07e9f9ffe83b9c86faa685f9080725dc21a916a607313be5/xxhash-3.8.1-cp314-cp314t-win3
2.whl", hash = "sha256:c240939e963653054fc7e4a17c382829cda4aa88a7daf0af841715dbded1b497", size = 33150, upload-time = "2026-07-06T10:47:57.274Z" },
    { url = "https://files.pythonhosted.org/packages/54/1c/09703eb341f8416e74e58d6c6732d4b5c46de59c942363203cb237cc95b0/xxhash-3.8.1-cp314-cp314t-win_
amd64.whl", hash = "sha256:7258ee276e8772599bc19e14b36f6260306e21b637190cd7cb489a2449d48684", size = 34005, upload-time = "2026-07-06T10:47:59.434Z" }
,
    { url = "https://files.pythonhosted.org/packages/d6/f9/6ed7251bb6a8af10ac73b1821c60583d2826e5b2064e45a979c935287c98/xxhash-3.8.1-cp314-cp314t-win_
arm64.whl", hash = "sha256:8f454166c2ffed45636c8d501741e649851ba2f346c4eb73a64c07ac00428f20", size = 30239, upload-time = "2026-07-06T10:48:01.874Z" }
,
]

[[package]]
name = "yarl"
version = "1.24.5"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "idna" },
    { name = "multidict" },
    { name = "propcache" },
]
sdist = { url = "https://files.pythonhosted.org/packages/31/33/ebe9e3d1f86c7a0b51094c0a146392045ca1631d2664889539dec8088a33/yarl-1.24.5.tar.gz", hash
= "sha256:e81b83143bee16329c23db3c1b2d82b29892fcbcb849186d2f6e98a5abe9a57f", size = 228679, upload-time = "2026-07-20T02:07:45.435Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/1b/84/71d051c850b5af41d168c679d9eb67eb7c55283ac4ee131673edf134bc4e/yarl-1.24.5-cp312-cp312-macosx
_10_13_universal2.whl", hash = "sha256:d693396e5aea78db03decd60aec9ece16c9b40ba00a587f089615ff4e718a81d", size = 136035, upload-time = "2026-07-20T02:
05:25.489Z" },
    { url = "https://files.pythonhosted.org/packages/03/4d/8ad27f9a1b7e69313cca5d695b925b48efe51208d3490e0844bae97cabc0/yarl-1.24.5-cp312-cp312-macosx
_10_13_x86_64.whl", hash = "sha256:3363fcc96e665878946ad7a106b9a13eac0541766a690ef287c0232ac768b6ec", size = 97642, upload-time = "2026-07-20T02:05:27
.429Z" },
    { url = "https://files.pythonhosted.org/packages/ea/b4/05b4131c407006cd1e410e9c6539f16a0945724677e5364447313c15ea3e/yarl-1.24.5-cp312-cp312-macosx
_11_0_arm64.whl", hash = "sha256:9d399bdcfb4a0f659b9b3788bbc89babe63d9a6a65aacdf4d4e7065ff2e6316c", size = 97323, upload-time = "2026-07-20T02:05:29.4
41Z" },
    { url = "https://files.pythonhosted.org/packages/20/16/e618c875c73e0e39611f20a581b3d5e8d59b8857bf001bee3263044c6deb/yarl-1.24.5-cp312-cp312-manyli
nux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:90333fd89b43c0d08ac85f3f1447593fc2c66de18c3d6378d7125ea118dc7a54",
size = 107741, upload-time = "2026-07-20T02:05:31.367Z" },
    { url = "https://files.pythonhosted.org/packages/d9/9a/c4defeaf3ed33fcb346aacf9c6e971a8d4e2bde04a0310e79abb208e7965/yarl-1.24.5-cp312-cp312-manyli
nux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:665b0a2c463cc9423dd647e0bfd9f4ccc9b50f768c55304d5e9f80b177c1de12", siz
e = 103570, upload-time = "2026-07-20T02:05:33.303Z" },
    { url = "https://files.pythonhosted.org/packages/5f/e7/0e0e0de5865ebd5914537ef486f36c727a59865c3ac0cf5ff1b32aececbf/yarl-1.24.5-cp312-cp312-manyli
nux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:e006d3a974c4ee19512e5f058abedb6eef36a5e553c14812bdeba1758d812e6d",
size = 115815, upload-time = "2026-07-20T02:05:35.292Z" },
    { url = "https://files.pythonhosted.org/packages/2b/27/ca56b700cb170aba25a3893b75355b213935657dc5714d2383354a270e62/yarl-1.24.5-cp312-cp312-manyli
nux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:e7d42c531243450ef0d4d9c172e7ed6ef052640f195629065041b5add4e058d1", size =
 116025, upload-time = "2026-07-20T02:05:37.503Z" },
    { url = "https://files.pythonhosted.org/packages/d6/d0/d56c859b8222116f5d68459199f48359e0bf121b6f65a69bf329b3602ba0/yarl-1.24.5-cp312-cp312-manyli
nux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:f08c7513ecef5aad65687bfdf6bc601ae9fccd04a42904501f8f7141abad9eb9", siz
e = 109835, upload-time = "2026-07-20T02:05:39.506Z" },
    { url = "https://files.pythonhosted.org/packages/70/a2/3a35557e4d1a79425040eba202ccaf08bdc8717680fc77e2498a1ad2e0a5/yarl-1.24.5-cp312-cp312-manyli
nux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:6c95b17fe34ed802f17e205112e6e10db92275c34fee290aa9bdc55a9c724027", size = 108884, upload-
time = "2026-07-20T02:05:41.584Z" },
    { url = "https://files.pythonhosted.org/packages/e4/35/ef4c26356b7913c68983bac2d72a4212b3347af551cb8d250b99b5ed7b7f/yarl-1.24.5-cp312-cp312-muslli
nux_1_2_aarch64.whl", hash = "sha256:56b149b22de33b23b0c6077ab9518c6dcb538ad462e1830e68d06591ccf6e38b", size = 107308, upload-time = "2026-07-20T02:05
:43.697Z" },
    { url = "https://files.pythonhosted.org/packages/d5/91/ff0dc66c2ccf3e0153ab97ff61eabab4400e6a5264af427ab30cd69f1857/yarl-1.24.5-cp312-cp312-muslli
nux_1_2_armv7l.whl", hash = "sha256:a8fe66b8f300da93798025a785a5b90b42f3810dc2b72283ff84a41aaaebc293", size = 103646, upload-time = "2026-07-20T02:05:
```

45.895Z" },
    { url = "https://files.pythonhosted.org/packages/74/f0/33b9271c7f881766359d58266fa0811d2e5210ed860e28da7dc6d7786344/yarl-1.24.5-cp312-cp312-muslinux_1_2_ppc64le.whl", hash = "sha256:377fe3732edbaf78ee74efdf2c9f49f6e99f20e7f9d2649fda3eb4badd77d76e", size = 115305, upload-time = "2026-07-20T02:05:47.832Z" },
    { url = "https://files.pythonhosted.org/packages/ef/65/fd79fb1868c4a80db8661091de525bf430f63c3bea1b20e8b6a84fc7d359/yarl-1.24.5-cp312-cp312-muslinux_1_2_riscv64.whl", hash = "sha256:e8ffa78582120024f476a611d7befc123cee59e47e8309d470cf667d806e613b", size = 108404, upload-time = "2026-07-20T02:05:49.604Z" },
    { url = "https://files.pythonhosted.org/packages/ff/ba/dbabe6b262f17a816c70cfc09558dbf03ece3ec76684d02f911a3d3a189c/yarl-1.24.5-cp312-cp312-muslinux_1_2_s390x.whl", hash = "sha256:daba5e594f06114e37db186efd2dd916609071e59daca901a0a2e71f02b142ce", size = 115940, upload-time = "2026-07-20T02:05:51.741Z" },
    { url = "https://files.pythonhosted.org/packages/a5/43/fab2d1dad9d340a268cdde63756a123d069723efff6a372d123fa74a9517/yarl-1.24.5-cp312-cp312-muslinux_1_2_x86_64.whl", hash = "sha256:65be18ec59496c13908f02a2472751d9ef840b4f3fb5726f129306bf6a2a7bba", size = 110006, upload-time = "2026-07-20T02:05:53.554Z" },
    { url = "https://files.pythonhosted.org/packages/c4/27/41eb51bbd1b8d89546b83897cfb0164f1e109304fd408dbb151b639eec0f/yarl-1.24.5-cp312-cp312-win_amd64.whl", hash = "sha256:a929d878fec099030c292803b31e5d5540a7b6a31e6a3cc76cb4685fc2a2f51b", size = 97618, upload-time = "2026-07-20T02:05:55.57Z" },
    { url = "https://files.pythonhosted.org/packages/3c/25/b2553764b3d65db711d8f45416351ec4f420847558eb669edcbcaadf5780/yarl-1.24.5-cp312-cp312-win_amd64.whl", hash = "sha256:7ce27823052e2013b597e0c738b13e7e36b8ccb9400df8959417b052ab0fd92c", size = 93018, upload-time = "2026-07-20T02:05:57.554Z" },
    { url = "https://files.pythonhosted.org/packages/e1/63/64ef361967cc983573149dc1515d531db5da8a4c92d22bb833d59e01b313/yarl-1.24.5-cp313-cp313-macosx_10_13_universal2.whl", hash = "sha256:79af890482fc94648e8cde4c68620378f7fef60932710fa17a66abc039244da2", size = 135075, upload-time = "2026-07-20T02:05:59.671Z" },
    { url = "https://files.pythonhosted.org/packages/bb/89/55920fd853ce43e608adbc3962456f0d649d6bb15250dc2988321da0fe1c/yarl-1.24.5-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:46c2f213e23a04b93a392942d782eb9e413e6ef6bf7c8c53884e599a5c174dcb", size = 97225, upload-time = "2026-07-20T02:06:01.769Z" },
    { url = "https://files.pythonhosted.org/packages/15/f0/7688d3f2cfff7590df2af38ec46d969f4281a4dddb08a9ad2eafbcdddf98/yarl-1.24.5-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:92ab3e11448f2ff7bf53c5a26eff0edc086898ec8b21fb154b85839ce1d88075", size = 96751, upload-time = "2026-07-20T02:06:03.676Z" },
    { url = "https://files.pythonhosted.org/packages/05/1a/a851a0f94aaaf379dd4f901bfc80f634280bec51eb260b47363e2a4cd62e/yarl-1.24.5-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:ebb0ec7f17803063d5aeb982f3b1bd2b2f4e4fae6751226cbd6ba1fcfe9e63ff", size = 107960, upload-time = "2026-07-20T02:06:05.699Z" },
    { url = "https://files.pythonhosted.org/packages/6c/a8/faea066c12f9c77ca0de90641f1655f9dd7b412477bf28c76d692f3aecff/yarl-1.24.5-cp313-cp313-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:82632daed195dcc8ea664e8556dc9bdbd671960fb3776bd92806ce05792c2448", size = 103500, upload-time = "2026-07-20T02:06:07.556Z" },
    { url = "https://files.pythonhosted.org/packages/fb/9c/1e67084c2a6e2f2db0e3be798328cb3be42c0119b621d25461479a224d21/yarl-1.24.5-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:53e549287ef628fecba270045c9701b0c564563a9b0577d24a4ec75b8ab8040f", size = 115780, upload-time = "2026-07-20T02:06:09.599Z" },
    { url = "https://files.pythonhosted.org/packages/58/86/1f94664e147474337e3359f52012cf3d02f825f694317b178bfba1078c62/yarl-1.24.5-cp313-cp313-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:fcd3b77e2f17bbe4ca56ec7bcb07992647d19d0b9c05d84886dcd6f9eb810afd", size = 115308, upload-time = "2026-07-20T02:06:11.352Z" },
    { url = "https://files.pythonhosted.org/packages/0a/43/8e55ae7538ba5f28ccb3c845c6dd4549cf7016d5992e5326512519107cdd/yarl-1.24.5-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:d46b86567dd4e248c6c159fcbcdcce01e0a5c8a7cd2334a0fff759d0fa075b16", size = 110574, upload-time = "2026-07-20T02:06:13.129Z" },
    { url = "https://files.pythonhosted.org/packages/ce/ba/a889ec8765cedcf2ac44dcb02d6a21e4861399b243b263c5f2dde27ee740/yarl-1.24.5-cp313-cp313-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:7f72c74aa99359e27a2ee8d6613fefa28b5f76a983c083074dfc2aaa4ab46213", size = 109914, upload-time = "2026-07-20T02:06:15.243Z" },
    { url = "https://files.pythonhosted.org/packages/9c/c3/e45f821af67b791c2dbbe4a9f4137a1d33f8d386654a05a0c3f47bdfa25d/yarl-1.24.5-cp313-cp313-muslinux_1_2_aarch64.whl", hash = "sha256:3f45789ce415a7ec0820dc4f82925f9b5f7732070be1dec1f5f23ec381435a24", size = 107712, upload-time = "2026-07-20T02:06:17.443Z" },
    { url = "https://files.pythonhosted.org/packages/02/00/2ab0f42c9857fcb490bfaa6647b14540b53d241ab209f23220b958cc5832/yarl-1.24.5-cp313-cp313-muslinux_1_2_armv7l.whl", hash = "sha256:6e73e7fe93f17a7b191f52ec9da9dd8c06a8fe735a1ecbd13b97d1c723bff385", size = 104251, upload-time = "2026-07-20T02:06:19.259Z" },
    { url = "https://files.pythonhosted.org/packages/7a/70/709d9a286e98af2c7fd8e4e6cada658b5c0e30d87dd7e2a63c2fb5767217/yarl-1.24.5-cp313-cp313-muslinux_1_2_ppc64le.whl", hash = "sha256:4a36f9becdd4c5c52a20c3e9484128b070b1dcfc8944c006f3a528295a359a9c", size = 115319, upload-time = "2026-07-20T02:06:21.207Z" },
    { url = "https://files.pythonhosted.org/packages/5c/6c/3eaa515142991fe84cfc483ff986492211f1978f90161ccefdbec919d09b/yarl-1.24.5-cp313-cp313-muslinux_1_2_riscv64.whl", hash = "sha256:7bcbe0fcf850eae67b6b01749815a4f7161c560a844c769ad7b48fcd99f791c4", size = 109163, upload-time = "2026-07-20T02:06:23.006Z" },
    { url = "https://files.pythonhosted.org/packages/bb/64/711dafce66c323a3144d470547a71c5384c57623308ac8bb5e4b903ac148/yarl-1.24.5-cp313-cp313-muslinux_1_2_s390x.whl", hash = "sha256:24e861e9630e0daddcb9191fb187f60f034e17a4426f8101279f0c475cd74144", size = 115435, upload-time = "2026-07-20T02:06:24.923Z" },
    { url = "https://files.pythonhosted.org/packages/cf/f3/9b9d0e6d84bea851eb1ba99e4bdc755b86fd813e49ec86dfe42f26befdef/yarl-1.24.5-cp313-cp313-muslinux_1_2_x86_64.whl", hash = "sha256:9335a099ad87287c37fe5d1a982ff392fa5efe5d14b40a730b1ec1d6a41382b4", size = 110691, upload-time = "2026-07-20T02:06:26.973Z" },
    { url = "https://files.pythonhosted.org/packages/86/e4/62a06b7e87c4246ac76b7c2da136f972eb4a3a1fc94abb07e7022d6fdb0a/yarl-1.24.5-cp313-cp313-win_amd64.whl", hash = "sha256:2dbe06fc16bc91502bca713704022182e5729861ae00277c3a23354b40929740", size = 97454, upload-time = "2026-07-20T02:06:29.163Z" },
    { url = "https://files.pythonhosted.org/packages/9e/c9/5fc8025b318ab10db413b61056bd0d95c557a70e8df4210c7511f866329c/yarl-1.24.5-cp313-cp313-win_arm64.whl", hash = "sha256:6b8536851f9f65e7f00c7a1d49ba7f2be0ffe2c11555367fc9f50d9f842410a1", size = 92813, upload-time = "2026-07-20T02:06:31.113Z" },
    { url = "https://files.pythonhosted.org/packages/a9/08/5f3085fef9564217074db9dd8573de1795bc82cde61a7ad10b6a7234a569/yarl-1.24.5-cp314-cp314-macosx_10_15_universal2.whl", hash = "sha256:2729fcfc4f6a596fb0c50f32090400aa9367774ac296a00387e65098c0befa76", size = 135680, upload-time = "2026-07-20T02:06:33.273Z" },
    { url = "https://files.pythonhosted.org/packages/98/35/ba9436e579bd48a8801f2021d842d9ab4994c26e4c7dd3a4c1f1bcb57a9e/yarl-1.24.5-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:ff330d3c30db4eb6b01d79e29d2d0b407a7ecad39cfd9ec993ece57396a2ec0d", size = 97395, upload-time = "2026-07-20T02:06:35.259Z" },
    { url = "https://files.pythonhosted.org/packages/18/a9/a07f76f3c44e02b25cc743af5ef93eef27f7013eadca770451b6a6ccb5db/yarl-1.24.5-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:e42d75862735da90e7fc5a7b23db0c976f737113a54b3c9777a9b665e9cbff75", size = 97223, upload-time = "2026-07-20T02:06:37.216Z" },
    { url = "https://files.pythonhosted.org/packages/77/f7/a9a1d6fa7dd9e388f95b30f6ad3ec4e285f6c8f61f44ce16070c3fcfe414/yarl-1.24.5-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:a3732e66413163e72508da9eff9ce9d2846fde51fae45d3605393d3e6cd303e9", size = 108777, upload-time = "2026-07-20T02:06:39.292Z" },
    { url = "https://files.pythonhosted.org/packages/2f/44/e0b86c302471fabd6f02808ecf2ac52b8412b624787849d4bf2cdb466f6f/yarl-1.24.5-cp314-cp314-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:5b8ee53be440a0cffc991a27be3057e0530122548dbe7c0892df08822fce5ede", size = 103119, upload-time = "2026-07-20T02:06:41.456Z" },
    { url = "https://files.pythonhosted.org/packages/d1/16/9c16d180bf8faaf223225eb50e1245870ff1ae0e302a27153988e65c51fd/yarl-1.24.5-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:af3aefa655adb5869491fa907e652290386800ae99cc50095cba71e2c6aefdca", size = 116471, upload-time = "2026-07-20T02:06:43.696Z" },
    { url = "https://files.pythonhosted.org/packages/d2/8d/b219b9df28a02ce95cfbdd41d2f7caa5669d0ff979c1c9975697145e33c5/yarl-1.24.5-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:2120b96872df4a117cde97d270bac96aea7cc52205d305cf4611df694a487027", size = 115974, upload-time = "2026-07-20T02:06:45.874Z" },
    { url = "https://files.pythonhosted.org/packages/9b/e8/f20557aca240d88e69850ad1ee91756821d094bb1310565c04d25c6682a2/yarl-1.24.5-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:66410eb6345d467151934b49bfa70fb32f5b35a6140baa40ad97d6436abea2e9", size = 110830, upload-time = "2026-07-20T02:06:47.852Z" },
    { url = "https://files.pythonhosted.org/packages/db/18/199b85109a53eeca64ee19c9cca228287e8e4ab0cc1a09b28f530e65cce0/yarl-1.24.5-cp314-cp314-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:4af7b7e1be0a69bee8210735fe6dcfc38879adfac6d62e789d53ba432d1ffa41", size = 110054, upload-time = "2026-07-20T02:06:49.84Z" },
    { url = "https://files.pythonhosted.org/packages/aa/2f/ed28147f8cd7f48c49367c90713b30a555284b6105a6a56f3a05568da795/yarl-1.24.5-cp314-cp314-muslinux_1_2_aarch64.whl", hash = "sha256:fa139875ff98ab97da323cfadfaff08900d1ad42f1b5087b0b812a55c5a06373", size = 108312, upload-time = "2026-07-20T02:06:51.835Z" },
    { url = "https://files.pythonhosted.org/packages/c5/c5/55e16ae0a5c227cea8df1c6871ba57d614a34243146c05729caf2a1bd9c5/yarl-1.24.5-cp314-cp314-muslinux_1_2_armv7l.whl", hash = "sha256:0055afc45e864b92729ac7600e2d102c17bef060647e74bca75fa84d66b9ff36", size = 103662, upload-time = "2026-07-20T02:06:54.061Z" },
    { url = "https://files.pythonhosted.org/packages/8d/ea/dbd7c2caec459c9a426f18b02688ecbfb58620d0f6a3422d24769fbaf8ab/yarl-1.24.5-cp314-cp314-muslinux_1_2_ppc64le.whl", hash = "sha256:f0e466ed7511fe9d459a819edbc6c2585c0b6eabde9fa8a8947552468a7a6ef0", size = 116090, upload-time = "2026-07-20T02:06:56.015Z" },
    { url = "https://files.pythonhosted.org/packages/06/84/39ce4ce3059e07fece5fbdbee8c4053406af9aca911ce9fa5f8548aab6af/yarl-1.24.5-cp314-cp314-muslinux_1_2_riscv64.whl", hash = "sha256:f141474e85b7e54998ec5180530a7cda99ab29e282fa50e0756d89981a9b43c5", size = 109523, upload-time = "2026-07-20T02:06:57.926Z" },

```
    { url = "https://files.pythonhosted.org/packages/a9/8b/71ff44137b405c64a7788075669c24010019f57a7464b78c3a6cbee539d9/yarl-1.24.5-cp314-cp314-musllinux_1_2_s390x.whl", hash = "sha256:e2935f8c39e3b03e83519292d78f075189978f3f4adc15a78144c7c8e2a1cba5", size = 116084, upload-time = "2026-07-20T02:06:59.868Z" },
    { url = "https://files.pythonhosted.org/packages/62/c0/423078fdd4042e1862c11f0ffd977a0ffa393783c12bee94685923bc189e/yarl-1.24.5-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:9d1216a7f6f77836617dba35687c5b78a4170afc3c3f18fc788f785ba26565c4", size = 111006, upload-time = "2026-07-20T02:07:01.907Z" },
    { url = "https://files.pythonhosted.org/packages/cf/52/6daa2ee9d95e5c98b8128f8df91eb692eb423ab274b8cf08db52152fad26/yarl-1.24.5-cp314-cp314-win_amd64.whl", hash = "sha256:5ba4f78df2bcc19f764a4b26a8a4f5049c110090ad5825993aacb052bf8003ad", size = 99215, upload-time = "2026-07-20T02:07:03.852Z" },
    { url = "https://files.pythonhosted.org/packages/ec/0e/464a847d7359e0da75dd9fc5c1d1aa35d0159ea31e5f8e66a3c1c29ff3d0/yarl-1.24.5-cp314-cp314-win_arm64.whl", hash = "sha256:9e4e16c73d717c5cf27626c524d0a2e261ad20e46932b2670f64ad5dde23e26f", size = 94566, upload-time = "2026-07-20T02:07:06.074Z" },
    { url = "https://files.pythonhosted.org/packages/e2/55/e03acc4446772660bc335e86e41ef31e4d0d838fd641531a11a5ee33b493/yarl-1.24.5-cp314-cp314t-macosx_10_15_universal2.whl", hash = "sha256:e1ae548a9d901adca07899a4147a7c826bbcc06239d3ce9a59f57886a28a4c88", size = 142533, upload-time = "2026-07-20T02:07:08.284Z" },
    { url = "https://files.pythonhosted.org/packages/ae/71/4acd3a1fc7cf14345cdb302665ecd2097f62c365b4f14ca17d4f37775cf9/yarl-1.24.5-cp314-cp314t-macosx_10_15_x86_64.whl", hash = "sha256:ff405d91509d88e8d44129cd87b18d70acd1f0c1aeabd7bc3c46792b1fe2acba", size = 100776, upload-time = "2026-07-20T02:07:10.197Z" },
    { url = "https://files.pythonhosted.org/packages/ff/0b/cfb76b7fe99686db264bff829779a539d923e7564ffd7ef18da6c54c3774/yarl-1.24.5-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:47e98aab9d8d82ff682e7b0b5dded33bf138a32b817fcf7fa3b27b2d7c412928", size = 100913, upload-time = "2026-07-20T02:07:12.357Z" },
    { url = "https://files.pythonhosted.org/packages/8b/3f/7116e782992abbd4fb6948488aec72078895e929a23078290739e8396fce/yarl-1.24.5-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:f0a658a6d3fafee5c6f63c58f3e785c8c43c93fbc02bf9f2b6663f8185e0971f", size = 106507, upload-time = "2026-07-20T02:07:14.173Z" },
    { url = "https://files.pythonhosted.org/packages/33/90/d4d2d73ee78229cc889872eb8e085d8f5c6f51abdb178409fd9b23cf74fd/yarl-1.24.5-cp314-cp314t-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:4377407001ca3c057773f44d8ddd6358fa5f691407c1ba92210bd3cf8d9e4c95", size = 99219, upload-time = "2026-07-20T02:07:16.019Z" },
    { url = "https://files.pythonhosted.org/packages/3e/fa/a6df1a9bccd644eec00abee0dff4277416222cec435330fd1f2858523ec1/yarl-1.24.5-cp314-cp314t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:7c0494a31a1ac5461a226e7947a9c9b78c44e1dc7185164fa7e9651557a5d9bc", size = 111804, upload-time = "2026-07-20T02:07:18.141Z" },
    { url = "https://files.pythonhosted.org/packages/8a/9e/7b2a1f4bcc20e9447156dd2b1c4d01f70d9df0759025ee7d09a84ffae134/yarl-1.24.5-cp314-cp314t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:a7cff474ab7cd149765bb784cf6d78b32e18e20473fb7bda860bce98ab58e9da", size = 110943, upload-time = "2026-07-20T02:07:20.06Z" },
    { url = "https://files.pythonhosted.org/packages/08/ff/22c92affb0f9b623ca753d27d968b5625b868f12c6378d049d55ae247643/yarl-1.24.5-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:cbb833ccacdb5519eff9b8b71ee618cc2801c878e77e288775d77c3a2ced858a", size = 108251, upload-time = "2026-07-20T02:07:22.217Z" },
    { url = "https://files.pythonhosted.org/packages/45/44/5769b96298c1e195fb412997b6090af2a84105cf59c17613558a2d011d1f/yarl-1.24.5-cp314-cp314t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:82f75e05912e84b7a0fe57075d9c59de3cb352b928330f2eb69b2e1f54c3e1f0", size = 106025, upload-time = "2026-07-20T02:07:24.083Z" },
    { url = "https://files.pythonhosted.org/packages/4c/40/009e8e791fd9762c0e1567e69248acb4f49064597e1680874c16dd8bb798/yarl-1.24.5-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:16a2f5010280020e90f5330257e6944bc33e73593b136cc5a241e6c1dc292498", size = 106573, upload-time = "2026-07-20T02:07:26.248Z" },
    { url = "https://files.pythonhosted.org/packages/20/c6/b7480578f8a0a80946f36ad6df547ecec704f9ba69d2de60f8aa6f1c1cbf/yarl-1.24.5-cp314-cp314t-musllinux_1_2_armv7l.whl", hash = "sha256:ffcd54362564dc1a30fb74d8b8a6e5a6b11ebd5e27266adc3b7427a21a6c9104", size = 100751, upload-time = "2026-07-20T02:07:28.098Z" },
    { url = "https://files.pythonhosted.org/packages/d4/27/4476f3360b91a48c5cf125e91f59a3bd35299d84a431a258d57f5977bb11/yarl-1.24.5-cp314-cp314t-musllinux_1_2_ppc64le.whl", hash = "sha256:0465ec8cedc2349b97a6b595ace64084a50c6e839eca40aa0626f38b8350e331", size = 111643, upload-time = "2026-07-20T02:07:30.88Z" },
    { url = "https://files.pythonhosted.org/packages/4c/4b/5cdd3e5ee944e8af31e52f6cd3d3af5fd7b937e036ccbbba2c9ffebede95/yarl-1.24.5-cp314-cp314t-musllinux_1_2_riscv64.whl", hash = "sha256:4db9aecb141cb7a5447171b57aa1ed3a8fee06af40b992ffc31206c0b0121550", size = 106312, upload-time = "2026-07-20T02:07:33.06Z" },
    { url = "https://files.pythonhosted.org/packages/18/86/f406b0c2a6f99575de2da671ef47aa06f89a5be83a27a46971c3b86cecdb/yarl-1.24.5-cp314-cp314t-musllinux_1_2_s390x.whl", hash = "sha256:f540c013589084679a6c7fac07096b10159737918174f5dfc5e11bf5bca4dfe6", size = 110379, upload-time = "2026-07-20T02:07:35.155Z" },
    { url = "https://files.pythonhosted.org/packages/f0/6c/9f3adfbd3b30b4fa0f7ccb3a83eba2c1152d3fff554d535e640ba0f7ba2b/yarl-1.24.5-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:a61834fb15d81322d872eaafd333838ae7c9cea84067f232656f75965933d047", size = 108497, upload-time = "2026-07-20T02:07:37.35Z" },
    { url = "https://files.pythonhosted.org/packages/dd/37/91eb2e5ca883a529c1b390348a74cd9fc0512171727f547ce70bfe02be5c/yarl-1.24.5-cp314-cp314t-win_amd64.whl", hash = "sha256:5c88e5815a49d289e599f3513aa7fde0bc2092ff188f99c940f007f90f53d104", size = 102450, upload-time = "2026-07-20T02:07:39.578Z" },
    { url = "https://files.pythonhosted.org/packages/bf/f4/ed5c402ac8fde4403ed3366c2716bfddc8a6677ebd59f3d62772cc7fe468/yarl-1.24.5-cp314-cp314t-win_arm64.whl", hash = "sha256:cf139c02f5f23ef6532040a30ff662c00a318c952334f211046b8e60b7f17688", size = 97222, upload-time = "2026-07-20T02:07:41.55Z" },
    { url = "https://files.pythonhosted.org/packages/61/02/962c1cbfc401a30c1d034dc67ff395f64b52302c6d62de556c1fca99acc0/yarl-1.24.5-py3-none-any.whl", hash = "sha256:a33700d13d9b7d84fd10947b09ff69fb9a792e519c8cb9764a3ca70baa6c23a7", size = 58612, upload-time = "2026-07-20T02:07:43.461Z" },
]

[[package]]
name = "zstandard"
version = "0.25.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/fd/aa/3e0508d5a5dd96529cdc5a97011299056e14c6505b678fd58938792794b1/zstandard-0.25.0.tar.gz", hash = "sha256:7713e1179d162cf5c7906da876ec2ccb9c3a9dcbdffef0cc7f70c3667a205f0b", size = 711513, upload-time = "2025-09-14T22:15:54.002Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/82/fc/f26eb6ef91ae723a03e16eddb198abcfce2bc5a42e224d44cc8b6765e57e/zstandard-0.25.0-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:7b3c3a3ab9daa3eed242d6ecceead93aebbb8f5f84318d82cee643e019c4b73b", size = 795738, upload-time = "2025-09-14T22:16:56.237Z" },
    { url = "https://files.pythonhosted.org/packages/aa/1c/d920d64b22f8dd028a8b90e2d756e431a5d86194caa78e3819c7bf53b4b3/zstandard-0.25.0-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:913cbd31a400febff93b564a23e17c3ed2d56c064006f54efec210d586171c00", size = 640436, upload-time = "2025-09-14T22:16:57.774Z" },
    { url = "https://files.pythonhosted.org/packages/53/6c/288c3f0bd9fcfe9ca41e2c2fbfd17b2097f6af57b62a81161941f09afa76/zstandard-0.25.0-cp312-cp312-manylinux2010_i686.manylinux2014_i686.manylinux_2_12_i686.manylinux_2_17_i686.whl", hash = "sha256:011d388c76b11a0c165374ce660ce2c8efa8e5d87f34996aa80f9c0816698b64", size = 5343019, upload-time = "2025-09-14T22:16:59.302Z" },
    { url = "https://files.pythonhosted.org/packages/1e/15/efef5a2f204a64bdb5571e6161d49f7ef0fffdbca953a615efbec045f60f/zstandard-0.25.0-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:6dffecc361d079bb48d7caef5d673c88c8988d3d33fb74ab95b7ee6da42652ea", size = 5063012, upload-time = "2025-09-14T22:17:01.156Z" },
    { url = "https://files.pythonhosted.org/packages/b7/37/a6ce629ffdb43959e92e87ebdaeebb5ac81c944b6a75c9c47e300f85abdf/zstandard-0.25.0-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:7149623bba7fdf7e7f24312953bcf73cae103db8cae49f8154dd1eadc8a29ecb", size = 5394148, upload-time = "2025-09-14T22:17:03.091Z" },
    { url = "https://files.pythonhosted.org/packages/e3/79/2bf870b3abeb5c070fe2d670a5a8d1057a8270f125ef7676d29ea900f496/zstandard-0.25.0-cp312-cp312-manylinux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:6a573a35693e03cf1d67799fd01b50ff578515a8aeadd4595d2a7fa9f3ec002a", size = 5451652, upload-time = "2025-09-14T22:17:04.979Z" },
    { url = "https://files.pythonhosted.org/packages/53/60/7be26e610767316c028a2cbedb9a3beabdbe33e2182c373f71a1c0b88f36/zstandard-0.25.0-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:5a56ba0db2d244117ed744dfa8f6f5b366e14148e00de44723413b2f3938a902", size = 5546993, upload-time = "2025-09-14T22:17:06.781Z" },
    { url = "https://files.pythonhosted.org/packages/85/c7/3483ad9ff0662623f3648479b0380d2de5510abf00990468c286c6b04017/zstandard-0.25.0-cp312-cp312-musllinux_1_1_aarch64.whl", hash = "sha256:10ef2a79ab8e2974e2075fb984e5b9806c64134810fac21576f0668e7ea19f8f", size = 5046806, upload-time = "2025-09-14T22:17:08.415Z" },
    { url = "https://files.pythonhosted.org/packages/08/b3/206883dd25b8d1591a1caa44b54c2aad84badccf2f1de9e2d60a446f9a25/zstandard-0.25.0-cp312-cp312-musllinux_1_1_x86_64.whl", hash = "sha256:aaf21ba8fb76d102b696781bddaa0954b782536446083ae3fdaa6f16b25a1c4b", size = 5576659, upload-time = "2025-09-14T22:17:10.164Z" },
    { url = "https://files.pythonhosted.org/packages/9d/31/76c0779101453e6c117b0ff22565865c54f48f8bd807df2b00c2c404b8e0/zstandard-0.25.0-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:1869da9571d5e94a85a5e8d57e4e8807b175c9e4a6294e3b66fa4efb074d90f6", size = 4953933, upload-time = "2025-09-14T22:17:11.857Z" },
    { url = "https://files.pythonhosted.org/packages/18/e1/97680c664a1bf9a247a280a053d98e251424af51f1b196c6d52f117c9720/zstandard-0.25.0-cp312-cp312-musllinux_1_2_i686.whl", hash = "sha256:809c5bcb2c67cd0ed81e9229d227d4ca28f82d0f778fc5fea624a9def3963f91", size = 5268008, upload-time = "2025-09-14T22:17:13.627Z" },
    { url = "https://files.pythonhosted.org/packages/1e/73/316e4010de585ac798e154e88fd81bb16afc5c5cb1a72eeb16dd37e8024a/zstandard-0.25.0-cp312-cp312-m
```

usllinux_1_2_ppc64le.whl", hash = "sha256:f27662e4f7dbf9f9c12391cb37b4c4c3cb90ffbd3b1fb9284dadbbb8935fa708", size = 5433517, upload-time = "2025-09-14T22:17:16.103Z" },
    { url = "https://files.pythonhosted.org/packages/5b/60/dd0f8cfa8129c5a0ce3ea6b7f70be5b33d2618013a161e1ff26c2b39787c/zstandard-0.25.0-cp312-cp312-musllinux_1_2_s390x.whl", hash = "sha256:99c0c846e6e61718715a3c9437ccc625de26593fea60189567f0118dc9db7512", size = 5814292, upload-time = "2025-09-14T22:17:17.827Z" },
    { url = "https://files.pythonhosted.org/packages/fc/5f/75aafd4b9d11b5407b641b8e41a57864097663699f23e9ad4dbb91dc6bfe/zstandard-0.25.0-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:474d2596a2dbc241a556e965fb76002c1ce655445e4e3bf38e5477d413165ffa", size = 5360237, upload-time = "2025-09-14T22:17:19.954Z" },
    { url = "https://files.pythonhosted.org/packages/ff/8d/0309daffea4fcac7981021dbf21cdb2e3427a9e76bafbcdbdf5392ff99a4/zstandard-0.25.0-cp312-cp312-win32.whl", hash = "sha256:23ebc8f17a03133b4426bcc04aabd68f8236eb78c3760f12783385171b0fd8bd", size = 436922, upload-time = "2025-09-14T22:17:24.398Z" },
    { url = "https://files.pythonhosted.org/packages/79/3b/fa54d9015f945330510cb5d0b0501e8253c127cca7ebe8ba46a965df18c5/zstandard-0.25.0-cp312-cp312-win_amd64.whl", hash = "sha256:ffef5a74088f1e09947aecf91011136665152e0b4b359c42be3373897fb39b01", size = 506276, upload-time = "2025-09-14T22:17:21.429Z" },
    { url = "https://files.pythonhosted.org/packages/ea/6b/8b51697e5319b1f9ac71087b0af9a40d8a6288ff8025c36486e0c12abcc4/zstandard-0.25.0-cp312-cp312-win_arm64.whl", hash = "sha256:181eb40e0b6a29b3cd2849f825e0fa34397f649170673d385f3598ae17cca2e9", size = 462679, upload-time = "2025-09-14T22:17:23.147Z" },
    { url = "https://files.pythonhosted.org/packages/35/0b/8df9c4ad06af91d39e94fa96cc010a24ac4ef1378d3efab9223cc8593d40/zstandard-0.25.0-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:ec996f12524f88e151c339688c3897194821d7f03081ab35d31d1e12ec975e94", size = 795735, upload-time = "2025-09-14T22:17:26.042Z" },
    { url = "https://files.pythonhosted.org/packages/3f/06/9ae96a3e5dcfd119377ba33d4c42a7d89da1efabd5cb3e366b156c45ff4d/zstandard-0.25.0-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:a1a4ae2dec3993a32247995bdfe367fc3266da832d82f8438c8570f989753de1", size = 640440, upload-time = "2025-09-14T22:17:27.366Z" },
    { url = "https://files.pythonhosted.org/packages/d9/14/933d27204c2bd404229c69f445862454dcc101cd69ef8c6068f15aaec12c/zstandard-0.25.0-cp313-cp313-manylinux2010_i686.manylinux2014_i686.manylinux_2_12_i686.manylinux_2_17_i686.whl", hash = "sha256:e96594a5537722fdfb79951672a2a63aec5ebfb823e7560586f7484819f2a08f", size = 5343070, upload-time = "2025-09-14T22:17:28.896Z" },
    { url = "https://files.pythonhosted.org/packages/6d/db/ddb11011826ed7db9d0e485d13df79b58586bfdec56e5c84a928a9a78c1c/zstandard-0.25.0-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:bfc4e20784722098822e3eee42b8e576b379ed72cca4a7cb856ae733e62192ea", size = 5063001, upload-time = "2025-09-14T22:17:31.044Z" },
    { url = "https://files.pythonhosted.org/packages/db/00/87466ea3f99599d02a5238498b87bf84a6348290c19571051839ca943777/zstandard-0.25.0-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:457ed498fc58cdc12fc48f7950e02740d4f7ae9493dd4ab2168a47c93c31298e", size = 5394120, upload-time = "2025-09-14T22:17:32.711Z" },
    { url = "https://files.pythonhosted.org/packages/2b/95/fc5531d9c618a679a20ff6c29e2b3ef1d1f4ad66c5e161ae6ff847d102a9/zstandard-0.25.0-cp313-cp313-manylinux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:fd7a5004eb1980d3cefe26b2685bcb0b17989901a70a1040d1ac86f1d898c551", size = 5451230, upload-time = "2025-09-14T22:17:34.41Z" },
    { url = "https://files.pythonhosted.org/packages/63/4b/e3678b4e776db00f9f7b2fe58e547e8928ef32727d7a1ff01dea010f3f13/zstandard-0.25.0-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:8e735494da3db08694d26480f1493ad2cf86e99bdd53e8e9771b2752a5c0246a", size = 5547173, upload-time = "2025-09-14T22:17:36.084Z" },
    { url = "https://files.pythonhosted.org/packages/4e/d5/ba05ed95c6b8ec30bd468dfeab20589f2cf709b5c940483e31d991f2ca58/zstandard-0.25.0-cp313-cp313-musllinux_1_1_aarch64.whl", hash = "sha256:3a39c94ad7866160a4a46d772e43311a743c316942037671beb264e395bdd611", size = 5046736, upload-time = "2025-09-14T22:17:37.891Z" },
    { url = "https://files.pythonhosted.org/packages/50/d5/870aa06b3a76c73eced65c044b92286a3c4e00554005ff51962deef28e28/zstandard-0.25.0-cp313-cp313-musllinux_1_1_x86_64.whl", hash = "sha256:172de1f06947577d3a3005416977cce6168f2261284c02080e7ad0185faeced3", size = 5576368, upload-time = "2025-09-14T22:17:40.206Z" },
    { url = "https://files.pythonhosted.org/packages/5d/35/398dc2ffc89d304d59bc12f0fdd931b4ce455bddf7038a0a67733a25f550/zstandard-0.25.0-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:3c83b0188c852a47cd13ef3bf9209fb0a77fa5374958b8c53aaa699398c6bd7b", size = 4954022, upload-time = "2025-09-14T22:17:41.879Z" },
    { url = "https://files.pythonhosted.org/packages/9a/5c/36ba1e5507d56d2213202ec2b05e8541734af5f2ce378c5d1ceaf4d88dc4/zstandard-0.25.0-cp313-cp313-musllinux_1_2_i686.whl", hash = "sha256:1673b7199bbe763365b81a4f3252b8e80f44c9e323fc42940dc8843bfeaf9851", size = 5267889, upload-time = "2025-09-14T22:17:43.577Z" },
    { url = "https://files.pythonhosted.org/packages/70/e8/2ec6b6fb7358b2ec0113ae202647ca7c0e9d15b61c005ae5225ad0995df5/zstandard-0.25.0-cp313-cp313-musllinux_1_2_ppc64le.whl", hash = "sha256:0be7622c37c183406f3dbf0cba104118eb16a4ea7359eeb5752f0794882fc250", size = 5433952, upload-time = "2025-09-14T22:17:45.271Z" },
    { url = "https://files.pythonhosted.org/packages/7b/01/b5f4d4dbc59ef193e870495c6f1275f5b2928e01ff5a81fecb22a06e22fb/zstandard-0.25.0-cp313-cp313-musllinux_1_2_s390x.whl", hash = "sha256:5f5e4c2a23ca271c218ac025bd7d635597048b366d6f31f420aaeb715239fc98", size = 5814054, upload-time = "2025-09-14T22:17:47.08Z" },
    { url = "https://files.pythonhosted.org/packages/b2/e5/fbd822d5c6f427cf158316d012c5a12f233473c2f9c5fe5ab1ae5d21f3d8/zstandard-0.25.0-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:4f187a0bb61b35119d1926aee039524d1f93aaf38a9916b8c4b78ac8514a0aaf", size = 5360113, upload-time = "2025-09-14T22:17:48.893Z" },
    { url = "https://files.pythonhosted.org/packages/8e/e0/69a553d2047f9a2c7347caa225bb3a63b6d7704ad74610cb7823baa08ed7/zstandard-0.25.0-cp313-cp313-win32.whl", hash = "sha256:7030defa83eef3e51ff26f0b7bfb229f0204b66fe18e04359ce3474ac33cbc09", size = 436936, upload-time = "2025-09-14T22:17:52.658Z" },
    { url = "https://files.pythonhosted.org/packages/d9/82/b9c06c870f3bd8767c201f1edbdf9e8dc34be5b0fbc5682c4f80fe948475/zstandard-0.25.0-cp313-cp313-win_amd64.whl", hash = "sha256:1f830a0dac88719af0ae43b8b2d6aef487d437036468ef3c2ea59c51f9d55fd5", size = 506232, upload-time = "2025-09-14T22:17:50.402Z" },
    { url = "https://files.pythonhosted.org/packages/d4/57/60c3c01243bb81d381c9916e2a6d9e149ab8627c0c7d7abb2d73384b3c0c/zstandard-0.25.0-cp313-cp313-win_arm64.whl", hash = "sha256:85304a43f4d513f5464ceb938aa02c1e78c2943b29f44a750b48b25ac999a049", size = 462671, upload-time = "2025-09-14T22:17:51.533Z" },
    { url = "https://files.pythonhosted.org/packages/3d/5c/f8923b595b55fe49e30612987ad8bf053aef555c14f05bb659dd5dbe3e8a/zstandard-0.25.0-cp314-cp314-macosx_10_13_x86_64.whl", hash = "sha256:e29f0cf06974c899b2c188ef7f783607dbef36da4c242eb6c82dcd8b512855e3", size = 795887, upload-time = "2025-09-14T22:17:54.198Z" },
    { url = "https://files.pythonhosted.org/packages/8d/09/d0a2a14fc3439c5f874042dca72a79c70a532090b7ba0003be73fee37ae2/zstandard-0.25.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:05df5136bc5a011f33cd25bc9f506e7426c0c9b3f9954f056831ce68f3b6689f", size = 640658, upload-time = "2025-09-14T22:17:55.423Z" },
    { url = "https://files.pythonhosted.org/packages/5d/7c/8b6b71b1ddd517f68ffb55e10834388d4f793c49c6b83effaaa05785b0b4/zstandard-0.25.0-cp314-cp314-manylinux2010_i686.manylinux_2_12_i686.manylinux_2_28_i686.whl", hash = "sha256:f604efd28f239cc21b3adb53eb061e2a205dc164be408e553b41ba2ffe0ca15c", size = 5379849, upload-time = "2025-09-14T22:17:57.372Z" },
    { url = "https://files.pythonhosted.org/packages/a4/86/a48e56320d0a17189ab7a42645387334fba2200e904ee47fc5a26c1fd8ca/zstandard-0.25.0-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:223415140608d0f0da010499eaa8ccdb9af210a543fac54bce15babbcfc78439", size = 5058095, upload-time = "2025-09-14T22:17:59.498Z" },
    { url = "https://files.pythonhosted.org/packages/f8/ad/eb659984ee2c0a779f9d06dbfe45e2dc39d99ff40a319895df2d3d9a48e5/zstandard-0.25.0-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:2e54296a283f3ab5a26fc9b8b5d4978ea0532f37b231644f367aa588930aa043", size = 5551751, upload-time = "2025-09-14T22:18:01.618Z" },
    { url = "https://files.pythonhosted.org/packages/61/b3/b637faea43677eb7bd42ab204dfb7053bd5c4582bfe6b1baefa80ac0c47b/zstandard-0.25.0-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:ca54090275939dc8ec5dea2d2afb400e0f83444b2fc24e07df7fdef677110859", size = 6364818, upload-time = "2025-09-14T22:18:03.769Z" },
    { url = "https://files.pythonhosted.org/packages/31/dc/cc50210e11e465c975462439a492516a73300ab8caa8f5e0902544fd748b/zstandard-0.25.0-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:e09bb6252b6476d8d56100e8147b803befa9a12cea144bbe629dd508800d1ad0", size = 5560402, upload-time = "2025-09-14T22:18:05.954Z" },
    { url = "https://files.pythonhosted.org/packages/c9/ae/56523ae9c142f0c08efd5e868a6da613ae76614eca1305259c3bf6a0ed43/zstandard-0.25.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:a9ec8c642d1ec73287ae3e726792dd86c96f5681eb8df274a757bf62b750eae7", size = 4955108, upload-time = "2025-09-14T22:18:07.68Z" },
    { url = "https://files.pythonhosted.org/packages/98/cf/c899f2d6df0840d5e384cf4c4121458c72802e8bda19691f3b16619f51e9/zstandard-0.25.0-cp314-cp314-musllinux_1_2_i686.whl", hash = "sha256:a4089a10e598eae6393756b036e0f419e8c1d60f44a831520f9af41c14216cf2", size = 5269248, upload-time = "2025-09-14T22:18:09.753Z" },
    { url = "https://files.pythonhosted.org/packages/1b/c0/59e912a531d91e1c192d3085fc0f6fb2852753c301a812d856d857ea03c6/zstandard-0.25.0-cp314-cp314-musllinux_1_2_ppc64le.whl", hash = "sha256:f67e8f1a324a900e75b5e28ffb152bcac9fbed1cc7b43f99cd90f395c4375344", size = 5430330, upload-time = "2025-09-14T22:18:11.966Z" },
    { url = "https://files.pythonhosted.org/packages/a0/1d/7e31db1240de2df22a58e2ea9a93fc6e38cc29353e660c0272b6735d6669/zstandard-0.25.0-cp314-cp314-musllinux_1_2_s390x.whl", hash = "sha256:9654dbc012d8b06fc3d19cc825af3f7bf8ae242226df5f83936cb39f5fdc846c", size = 5811123, upload-time = "2025-09-14T22:18:13.907Z" },
    { url = "https://files.pythonhosted.org/packages/f6/49/fac46df5ad353d50535e118d6983069df68ca5908d4d65b8c466150a4ff1/zstandard-0.25.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:4203ce3b31aec23012d3a4cf4a2ed64d12fea5269c49aed5e4c3611b938e4088", size = 5359591, upload-time = "2025-09-14T22:18:16.465Z" },
    { url = "https://files.pythonhosted.org/packages/c2/38/f249a2050ad1eea0bb364046153942e34abba95dd5520af199aed86fbb49/zstandard-0.25.0-cp314-cp314-w

```
in32.whl", hash = "sha256:da469dc041701583e34de852d8634703550348d5822e66a0c827d39b05365b12", size = 444513, upload-time = "2025-09-14T22:18:20.61Z" },
    { url = "https://files.pythonhosted.org/packages/3a/43/241f9615bcf8ba8903b3f0432da069e857fc4fd1783bd26183db53c4804b/zstandard-0.25.0-cp314-cp314-w
in_amd64.whl", hash = "sha256:c19bcdd826e95671065f8692b5a4aa95c52dc7a02a4c5a0cac46deb879a017a2", size = 516118, upload-time = "2025-09-14T22:18:17.849
Z" },
    { url = "https://files.pythonhosted.org/packages/f0/ef/da163ce2450ed4febf6467d77ccb4cd52c4c30ab45624bad26ca0a27260c/zstandard-0.25.0-cp314-cp314-w
in_arm64.whl", hash = "sha256:d7541afd73985c630bafcd6338d2518ae96060075f9463d7dc14cfb33514383d", size = 476940, upload-time = "2025-09-14T22:18:19.088
Z" },
]
```

CLAUDE.md

379 lines · 17.1 KB

Generative BI Platform — Project Context

> **Owner:** Iven Kwan | **Stack Tier:** Enterprise | **Mode:** Agentic Coding
> This file is the persistent memory layer for Claude Code. Read it fully before touching any code.

---

1. Project Overview

A **Generative BI (GenBI) platform** that enables business users to query enterprise data warehouses using natural language, auto-generates visualizat
ions and narrative insights, and delivers governed, explainable analytics at scale.

Core value proposition:
- Natural language → SQL → chart + narrative (single interaction cycle)
- Semantic layer as the single source of truth for metrics
- Multi-tenant, role-aware data access with full audit trail
- Deployable on-premise or cloud for regulated industries (FSI, AML/KYC contexts)

Primary users: Data analysts, business stakeholders, compliance officers, C-suite
Persona: Users who know *what* they want to see but not *how* to query it

---

2. Tech Stack

```
Backend:       Python 3.12 · FastAPI · Uvicorn / Gunicorn
LLM Layer:     Anthropic Claude API (claude-opus-4 for reasoning, claude-haiku-4 for speed)
               LangChain / LangGraph for orchestration and agent loops
               LlamaIndex for semantic indexing and RAG
Semantic Layer: dbt (MetricFlow) + Cube.dev (headless BI / metrics API)
Databases:     PostgreSQL 16 (primary store, metadata, audit)
               Apache AGE (graph queries for lineage & relationship analysis)
               pgvector (schema embeddings, semantic search)
               Redis (query cache, session state)
Vector Store:  pgvector (default) / Weaviate (scale-out option)
Charts:        Vega-Lite (declarative) + Plotly.js (interactive frontend)
Frontend:      Next.js 15 (App Router) · TypeScript 5.9 · Tailwind CSS v4 · shadcn/ui
Auth/RBAC:     FastAPI + JWT + row-level security (PostgreSQL RLS)
               DID/VC-compatible identity hooks (future)
Containerization: Docker + Docker Compose · GitHub Actions CI
Observability:  OpenTelemetry · Langfuse (LLM tracing) · Prometheus + Grafana
```

Package managers: `uv` for Python (NOT pip/poetry), `pnpm` for Node (NOT npm/yarn)
Python version: 3.12 — use match/case, walrus operator, typed dicts freely
NEVER use: raw `pip install`, `npm install`, `yarn`

---

3. Architecture Layers

```
┌──────────────────────────────────────────┐
│  Layer 1: Interface (Next.js, REST, Slack)  │
├──────────────────────────────────────────┤
│  Layer 2: Orchestration (LangGraph Agents)  │
│   ┌─────────┐ ┌─────────┐ ┌──────────┐  │
│   │ NL2SQL  │ │ ChartGen │ │ Narrative │  │
│   │ Agent   │ │ Agent   │ │ Agent    │  │
│   └─────────┘ └─────────┘ └──────────┘  │
├──────────────────────────────────────────┤
│  Layer 3: Semantic Layer (dbt + Cube.dev)   │
│  Metrics store · Schema catalog · RLS rules │
├──────────────────────────────────────────┤
│  Layer 4: Data Retrieval                    │
│  PostgreSQL · Snowflake · Redshift (via     │
│  SQLAlchemy connectors)                     │
├──────────────────────────────────────────┤
│  Layer 5: Governance & Audit                │
│  Prompt logs · Query lineage · Feedback     │
└──────────────────────────────────────────┘
```

Agent Responsibilities
Agent	Input	Output	Model
`NL2SQLAgent`	User prompt + schema context	SQL query + explanation	claude-opus-4
`ChartGenAgent`	SQL result set + prompt intent	Vega-Lite spec JSON	claude-haiku-4
`NarrativeAgent`	Data + chart + business context	Insight paragraph	claude-haiku-4
`RouterAgent`	Raw user query	Agent dispatch plan	claude-haiku-4
`ValidationAgent`	Generated SQL	Safety check + dry-run	deterministic

---

4. Project File Structure

```
genbi/
├── CLAUDE.md                  ← you are here
├── .claude/
│   ├── rules/
│   │   ├── sql-safety.md      ← SQL injection & destructive query rules
│   │   ├── llm-patterns.md    ← LLM prompt engineering conventions
│   │   └── api-design.md      ← FastAPI route design standards
│   └── skills/
│       ├── add-data-source/   ← how to onboard a new database connector
│       ├── add-agent/         ← how to register a new agent
│       └── deploy/            ← deployment runbook
├── backend/
```

```
│   ├── app/
│   │   ├── api/              ← FastAPI routers (versioned: /v1/*)
│   │   ├── agents/           ← LangGraph agent definitions
│   │   ├── semantic/         ← Cube.dev + dbt schema loaders
│   │   ├── connectors/       ← SQLAlchemy DB connectors
│   │   ├── services/         ← business logic (query service, chart service)
│   │   ├── models/           ← Pydantic schemas (NOT SQLAlchemy here)
│   │   ├── db/               ← SQLAlchemy ORM models + Alembic migrations
│   │   └── core/             ← config, auth, logging, middleware
│   ├── tests/
│   ├── pyproject.toml
│   └── Dockerfile
├── frontend/
│   ├── src/
│   │   ├── app/              ← Next.js App Router pages
│   │   ├── components/       ← shared UI (shadcn/ui base)
│   │   ├── lib/              ← API clients, utils
│   │   └── types/            ← TypeScript interfaces
│   ├── package.json
│   └── Dockerfile
├── semantic/
│   ├── dbt/                  ← dbt project (models, seeds, metrics)
│   └── cube/                 ← Cube.dev schema files
├── infra/
│   ├── docker-compose.yml
│   ├── docker-compose.dev.yml
│   └── postgres/init.sql
└── docs/
    ├── adr/                  ← Architecture Decision Records
    └── api/                  ← OpenAPI specs
```

---

5. Key Commands

```bash
# Backend
cd backend
uv run uvicorn app.main:app --reload --port 8000   # dev server
uv run pytest tests/ -v --tb=short                 # run tests
uv run alembic upgrade head                        # apply migrations
uv run alembic revision --autogenerate -m "msg"    # new migration
uv run ruff check . && uv run ruff format --check . # lint + format

# Frontend
cd frontend
pnpm dev                    # Next.js dev (port 3000)
pnpm build && pnpm start    # production build
pnpm lint                   # ESLint
pnpm typecheck              # tsc --noEmit

# Full stack
docker compose -f infra/docker-compose.dev.yml up -d   # start all services
docker compose down -v                                  # tear down (keeps volumes)
docker compose logs -f backend                          # tail backend logs

# Semantic layer
cd semantic/dbt
dbt run --select +metrics.*   # run dbt models
dbt test                      # run dbt tests
```

---

6. Hard Rules — NEVER Violate

Security & Data Safety
- **NEVER generate destructive SQL** (DROP, DELETE, TRUNCATE) — all DB access is READ-ONLY through the query engine. Insert/update only via explicit service methods.
- **NEVER expose raw connection strings or secrets** in responses, logs, or LLM prompts. Use env vars; reference `settings.py` which loads from `.env`.
- **NEVER pass user-submitted strings directly into SQL**. Always use SQLAlchemy parameterized queries or the query validator.
- **NEVER bypass `ValidationAgent`** before executing generated SQL in production paths.

LLM & Agent Rules
- **NEVER hardcode model names** in business logic — always reference `settings.LLM_REASONING_MODEL` and `settings.LLM_FAST_MODEL`.
- **NEVER call LLM APIs without timeout + retry logic**. Use the `llm_client.py` wrapper.
- **NEVER store raw user prompts in plaintext DB** without first stripping PII via the sanitizer middleware.
- **Every agent must log**: input tokens, output tokens, latency, model version, and user_id to the `audit_log` table.

Code Conventions
- **NEVER use `print()` for logging** — use `loguru` logger (`from app.core.logging import logger`).
- **NEVER use `Any` type in Pydantic models** — define explicit schemas.
- **NEVER commit `.env` files** — use `.env.example` with placeholder values only.
- **ALL API routes are versioned** under `/api/v1/` — no bare routes.

---

7. Coding Conventions

Python / Backend
- **Async everywhere**: all FastAPI routes and DB calls are `async def`; use `asyncpg` driver
- **Pydantic v2**: use `model_validator`, `field_validator`, `model_config` — not v1 syntax
- **SQLAlchemy 2.0**: use `select()`, `Session.execute()`, mapped classes — NOT legacy Query API
- **Error handling**: raise `HTTPException` with typed detail dicts `{"code": "...", "message": "..."}`
- **Dependency injection**: auth/session/db via FastAPI `Depends()` — never instantiate in route body
- **File naming**: `snake_case` for Python files, `kebab-case` for frontend files

TypeScript / Frontend
- **Server components by default** — add `"use client"` only for interactive components
- **All data fetching via** `src/lib/api-client.ts` — never `fetch()` directly in components
- **Zod schemas** for all API response validation in frontend
- **shadcn/ui components** as base — extend with Tailwind, never override base component internals

Agent & LLM Patterns
- **Prompt templates are versioned files** in `.claude/prompts/` — never inline long prompts in code
- **Few-shot examples** live in `backend/app/agents/examples/` — load dynamically, not hardcoded
- **Schema context compression**: embed only relevant table/column metadata (top-k by cosine similarity from pgvector) — never send full DB schema to LLM
- **Chain of thought**: reasoning models (`claude-opus-4`) always use `thinking` blocks for SQL generation; fast models (`claude-haiku-4`) use direct generation

---

8. Database Rules

```sql
-- Every table MUST have:
id          UUID PRIMARY KEY DEFAULT gen_random_uuid()
created_at  TIMESTAMPTZ NOT NULL DEFAULT NOW()
updated_at  TIMESTAMPTZ NOT NULL DEFAULT NOW()
tenant_id   UUID NOT NULL REFERENCES tenants(id)   -- for multi-tenancy

-- RLS policy must exist for every user-data table
ALTER TABLE <table> ENABLE ROW LEVEL SECURITY;
```

- **NEVER drop columns** — mark as `deprecated_at TIMESTAMPTZ` first, remove after 2 sprint cycles
- **Migrations**: Alembic only — no manual `ALTER TABLE` in psql
- **Indexes**: all foreign keys and `tenant_id` columns must be indexed
- **Query timeout**: all analytical queries are capped at 30s via `SET statement_timeout = '30s'`
- **Apache AGE**: graph queries for data lineage use the `genbi_graph` graph — schema in `backend/app/db/graph_schema.py`

---

9. Testing Approach

```
Unit tests:          Pure functions, agent logic, prompt builders, validators
Integration tests:   API routes with test DB, semantic layer queries
E2E tests:           Full NL → SQL → chart → narrative pipeline (slow, marked @pytest.mark.e2e)
LLM eval tests:      Golden dataset of NL/SQL pairs in tests/evals/nl2sql_golden.json
```

- **Always mock LLM calls in unit/integration tests** using `tests/fixtures/llm_mock.py`
- **Golden NL2SQL test suite** must pass before any merge to `main` (20 queries, 90% accuracy threshold)
- **Test DB** is a separate PostgreSQL instance with synthetic data (see `infra/postgres/test_seed.sql`)
- Run: `uv run pytest tests/ -v -m "not e2e"` for fast local feedback

---

10. Semantic Layer Rules

- **Metrics are defined ONCE in dbt MetricFlow** — `semantic/dbt/models/metrics/`
- **Cube.dev** exposes those metrics via REST + GraphQL API to agents and frontend
- **NEVER re-implement metric logic** in agent prompts or service code — always query via Cube API
- **Schema catalog** is synchronized to pgvector embeddings nightly via `scripts/embed_schema.py`
- **Column descriptions** are mandatory for all semantic model fields — they are injected into LLM context

---

11. Multi-Tenancy & Access Control

- All queries are scoped by `tenant_id` injected from JWT claims
- Row-level security (PostgreSQL RLS) enforces data isolation at DB level
- Column-level masking for PII fields (defined in `backend/app/core/masking.py`)
- Agent prompts include: `"You are operating for tenant {tenant_id} with role {user_role}. Only access data within your authorized scope."`
- Audit log records: `user_id`, `tenant_id`, `query_text`, `generated_sql`, `model_version`, `latency_ms`, `token_count`

---

12. Observability & Governance

```python
# Every LLM call must produce an AuditEntry
AuditEntry(
    session_id=...,
    user_id=...,
    tenant_id=...,
    input_prompt_hash=...,   # SHA-256, not raw text
    generated_sql=...,
    model_name=...,
    model_version=...,
    input_tokens=...,
    output_tokens=...,
    latency_ms=...,
    feedback_score=None,     # filled post-hoc
    hallucination_flag=False
)
```

- **Langfuse** is the LLM tracing backend — all traces tagged with `project=genbi`, `env=prod|dev`
- **OpenTelemetry** spans wrap every agent call and DB query
- **Hallucination detection**: if SQL result row count is 0 or diverges >50% from prior similar queries, flag for human review

---

13. DO NOT MODIFY (Stable / Protected)

Path	Reason
`backend/app/core/auth.py`	JWT + RBAC logic — changes need security review
`backend/app/db/migrations/`	Alembic history — never edit existing migration files
`backend/app/agents/validation/validation_agent.py`	SQL safety gate — any relaxation needs explicit approval
`semantic/cube/schema/`	Metric definitions — changes affect downstream Cube API
`infra/postgres/init.sql`	DB initialization — destructive if changed incorrectly

---

14. Deployment

```
Dev:     docker compose -f infra/docker-compose.dev.yml up
Staging: GitHub Actions → build Docker images → push to registry → deploy to K8s (staging NS)
Prod:    Manual approval gate in GitHub Actions → K8s (prod NS) rolling update

Required env vars (set in CI secrets / K8s secrets):
  ANTHROPIC_API_KEY
  DATABASE_URL           # asyncpg format: postgresql+asyncpg://...
  REDIS_URL
  CUBE_API_URL
  CUBE_API_SECRET
  LANGFUSE_SECRET_KEY
  LANGFUSE_PUBLIC_KEY
  JWT_SECRET_KEY
  TENANT_ENCRYPTION_KEY

Frontend build: `pnpm build` — outputs to .next/
Backend build: Docker image from backend/Dockerfile
```

---

15. Context Pointers

- `@docs/adr/001-semantic-layer-choice.md` — why dbt + Cube over standalone solutions
- `@docs/adr/002-agent-architecture.md` — LangGraph multi-agent topology decision
- `@docs/adr/003-nl2sql-approach.md` — schema embedding strategy and prompt design
- `@backend/app/agents/README.md` — agent registry and inter-agent messaging protocol
- `@semantic/dbt/README.md` — metric taxonomy and naming conventions

---

16. Common Tasks

"Add a new data source connector":
1. Create `backend/app/connectors/{name}_connector.py` implementing `BaseConnector`
2. Register in `backend/app/connectors/registry.py`
3. Add schema embedding job in `scripts/embed_schema.py`
4. Add connector integration test in `tests/connectors/`
5. Update `.env.example` with new connection string variable

"Add a new metric":
1. Define in `semantic/dbt/models/metrics/{domain}.yml` using MetricFlow syntax
2. Run `dbt run --select +metrics.{domain}` + `dbt test`
3. Cube.dev picks up on next sync (or `cd semantic/cube && cube build`)
4. Embed updated schema: `uv run python scripts/embed_schema.py --domain {domain}`

"Add a new agent":
1. Create `backend/app/agents/{name}_agent.py` extending `BaseAgent`
2. Register in `backend/app/agents/registry.py` with trigger conditions
3. Add unit tests in `tests/agents/test_{name}_agent.py` with mocked LLM
4. Add golden eval cases to `tests/evals/` if the agent generates SQL or charts
5. Update `RouterAgent` dispatch table in `backend/app/agents/router_agent.py`

"Debug a hallucinated SQL query":
1. Pull audit log: `SELECT * FROM audit_log WHERE session_id = '...'`
2. Check Langfuse trace for the session — inspect retrieved schema context
3. Check pgvector similarity scores in `schema_embeddings` table
4. If schema context was wrong, re-run `scripts/embed_schema.py` for affected tables
5. Add the failing NL query to `tests/evals/nl2sql_golden.json` as a regression case

---

17. Session Notes

When compacting, always preserve:
- The current migration state (last applied Alembic revision)
- Any failing golden eval queries and their root cause analysis
- Active agent architecture decisions in progress
- Any LLM prompt changes and their eval impact

---

docs/adr/001-dual-chart-stack.md

43 lines · 2.4 KB

```
# ADR 001: Dual Chart Stack — Flint MCP + AntV G2

**Status:** Accepted
**Date:** 2026-07-04
**Author:** Iven Kwan

## Context

GenBI needs to render charts from natural language queries. We evaluated two approaches:

1. **AntV G2 only** (DB-GPT's default): Interactive, web-native charts using G2's Grammar of Graphics. Used by DB-GPT's GPT-Vis protocol.
2. **Flint MCP only** (Microsoft's new chart framework): Unified `ChartAssemblyInput` spec compiled to Vega-Lite/ECharts/Chart.js. MCP-native, local rendering, PNG+SVG output.

## Decision

**We will use a dual-stack approach:**

- **Flint MCP** for static, export-quality charts (PNG/SVG for reports, emails, documents, dashboards)
- **AntV G2** (via GPT-Vis protocol) for interactive dashboard widgets (hover, zoom, brush, drill-down)

These are complementary, not competing. The agent pipeline selects the appropriate renderer based on context:

| Scenario | Renderer |
|---|---|
| Chat response with chart | Flint (SVG embedded inline) |
| Report generation | Flint (PNG at 2x scale) |
| Interactive dashboard tile | G2 (interactive canvas) |
| Email attachment | Flint (PNG) |
| Data exploration with brushing | G2 (interactive) |

## Rationale

**Why not Flint-only:** Flint is v0.1.0, stdio-only transport, no interactive features. It can't handle live dashboard widgets with hover/zoom/brush. Maturity risk for critical-path production use.

**Why not G2-only:** G2 produces only canvas/web output — no clean PNG/SVG export for documents and emails. The LLM has to navigate G2's complex configuration directly. Flint's unified schema is simpler for AI agents to author.

**Why both:** Each solves the other's weakness. Flint's unified schema is the AI-friendly authoring format. G2 is the interactive rendering engine. The `vis-flint` protocol extension routes static charts to Flint and interactive ones to G2.

## Consequences

- **Positive:** Best-in-class output for each use case. AI agent gets simple schema (Flint), users get interactive exploration (G2).
- **Negative:** Two chart stacks to maintain, deploy, and debug. Vendor risk on Flint (Microsoft, v0.1.0). Need `flint-chart-mcp` Node.js dependency alongside Python backend.
- **Mitigation:** FlintChartBridge has a Vega-Lite/Altair fallback path that doesn't require Flint MCP at all — the system runs without Flint if needed. Flint can be added later when it stabilizes.
```

docs/adr/002-agent-architecture.md

44 lines · 1.9 KB

```
# ADR 002: Multi-Agent Architecture with LangGraph

**Status:** Accepted
**Date:** 2026-07-04
**Author:** Iven Kwan

## Context

The GenBI pipeline (NL → SQL → Chart → Narrative) can be implemented as a single monolithic LLM call or as a chain of specialized agents. We need to decide the agent topology.

## Decision

**Multi-agent pipeline orchestrated by LangGraph, inspired by DB-GPT's AWEL patterns.**

```
User Query
  → RouterAgent (intent classification)
    → NL2SQLAgent (SQL generation with schema context)
      → ValidationAgent (SQL safety check + dry-run EXPLAIN)
        → Query Executor (read-only execution via SQLAlchemy)
          → ChartGenAgent (Flint ChartAssemblyInput spec + render)
            → NarrativeAgent (insight paragraph)
```

Each agent is a LangGraph node with a typed interface (`BaseAgent` → `AgentResult`). The `RouterAgent` classifies intent and selects which pipeline to run — not all queries need all agents.

Rationale

Why multi-agent over monolithic:
- Specialized prompts produce better results (SQL agent focuses on SQL, chart agent on visual encoding, etc.)
- Each agent can use a different model (Opus-4 for SQL reasoning, Haiku-4 for chart specs)
- Independent testing and evaluation (golden NL2SQL test suite independent from chart quality)
- Audit trail at agent granularity (which agent hallucinated?)

Why LangGraph over raw LangChain chains:
- Conditional routing (skip chart if no data, add validation only for SQL paths)
- Checkpointing for multi-turn conversations
- Streaming per-agent progress to the frontend

Consequences

- **Positive:** Modular, testable, observable. Model selection per agent. Independent iteration.
- **Negative:** More LLM calls per query (cost and latency). State management between agents. Need shared audit/observability layer.
- **Mitigation:** RouterAgent skips unnecessary agents. Fast model (Haiku-4) for non-SQL agents. Shared `AuditLog` model across all agents.
```

docs/adr/003-vis-flint-protocol.md

43 lines · 1.8 KB

```
# ADR 003: vis-flint Protocol Extension

**Status:** Accepted
**Date:** 2026-07-04
**Author:** Iven Kwan

## Context

DB-GPT's GPT-Vis protocol uses markdown code blocks (```vis-dashboard, ```vis_chart) to embed chart specifications in LLM responses. The frontend parses these blocks and renders AntV G2 charts. We need to extend this protocol for Flint Chart rendering.

## Decision

**Add a new `vis-flint` code block type that routes to Flint MCP for rendering.**

Extend the GPT-Vis protocol:
- ''''vis-dashboard` → AntV G2 (existing, for interactive multi-chart dashboards)
- ''''vis_chart` → AntV G2 (existing, for single interactive charts)
- ''''vis-flint` → Flint MCP (new, for static PNG/SVG charts)

The vis-flint block contains a raw `ChartAssemblyInput` JSON spec:

''''
```vis-flint
{
  "chartType": "Bar Chart",
  "encodings": {"x": {"field": "region"}, "y": {"field": "revenue"}},
  "data": {"values": [...]}
}
```
''''

Rationale

- **Backward compatible:** Existing vis-dashboard and vis_chart blocks continue to work with G2
- **Self-documenting:** The block type signals which renderer to use — no heuristic detection needed
- **AI-friendly:** The LLM emits the same `ChartAssemblyInput` schema it learned from Flint's agent skill
- **Frontend dispatch:** The markdown parser routes by block type — adding a new renderer is a registry entry

Consequences

- **Positive:** Clean separation of static (Flint) and interactive (G2) chart paths. The LLM chooses the right format by emitting the right code block.
- **Negative:** Two rendering paths in the chat pipeline. Frontend needs to handle both SVG (Flint) and canvas (G2) embeds.
- **Implementation:** Backend `vis_flint_protocol.py` module parses vis-flint blocks, renders via FlintChartBridge, and replaces blocks with inline SVG or base64 PNG.
```

docs/adr/004-flint-bridge-fallback.md

39 lines · 1.9 KB

```
# ADR 004: FlintChartBridge with Vega-Lite Fallback

**Status:** Accepted
**Date:** 2026-07-04
**Author:** Iven Kwan

## Context

Flint Chart (`flint-chart-mcp`) is v0.1.0 — very early stage, stdio-only transport, no HTTP/SSE, and subject to breaking changes. We need a chart rendering strategy that works today but can upgrade to Flint MCP when it stabilizes.

## Decision

**Implement a `FlintChartBridge` with a Vega-Lite/Altair fallback path.**

The bridge's architecture:

```
FlintChartBridge
  ├── Primary path: Flint MCP (via npx flint-chart-mcp stdio)
  │     ↓ compiles ChartAssemblyInput → Vega-Lite/ECharts/Chart.js → PNG/SVG
  │
  └── Fallback path: Altair (Python Vega-Lite wrapper)
        ↓ translates ChartAssemblyInput → Altair chart → SVG → PNG (via resvg)
```

The bridge auto-detects Flint availability and falls back silently. The `ChartAssemblyInput` schema is the unified API regardless of which path is active — callers never know whether Flint or Altair rendered the chart.

Rationale

- **Ship without Flint dependency:** The system works with just Altair installed. Flint is a future upgrade.
- **Same schema, same output:** Both paths accept `ChartAssemblyInput` and produce PNG/SVG. Callers are unchanged.
- **Graceful degradation:** If Flint MCP isn't available (CI, dev machines without Node, production without npm), Altair handles everything.
- **Migration path:** When Flint reaches v1.0 with SSE transport, we flip the primary path and keep Altair as the fallback.

Consequences

- **Positive:** Zero-risk Flint adoption. System runs today. Upgradable tomorrow.
- **Negative:** Altair has fewer chart types than Flint's full catalog. The fallback covers ~15 types vs. Flint's ~40. ECharts chart types (Sankey, radar, treemap) are Flint-only.
- **Mitigation:** The bridge returns warnings when a chart type falls outside Altair's supported set. The LLM is prompted to prefer types in the intersection.
```

docs/adr/005-age-and-pgvector-image.md

68 lines · 3.1 KB

```
# ADR 005: PostgreSQL image with both pgvector and Apache AGE

- **Status:** Accepted
- **Date:** 2026-08-05
- **Supersedes:** the implicit decision in the initial commit to use
  `pgvector/pgvector:pg16` while running `CREATE EXTENSION age` in `init.sql`.

## Context

The platform relies on two PostgreSQL extensions simultaneously:

- **pgvector** — stores schema/example embeddings for NL2SQL semantic search
  (`schema_embeddings.embedding VECTOR(1536)`, `agent_examples.embedding`).
- **Apache AGE** — property graph for data lineage and downstream-impact
  analysis (`backend/app/db/graph_schema.py`, graph `genbi_graph`).

The initial `docker-compose.*.yml` used `pgvector/pgvector:pg16`, which bundles
pgvector **only**. As a result `init.sql:11` (`CREATE EXTENSION age;`) and
`LOAD 'age';` failed on first container boot, the lineage graph could never be
created, and `init_age_graph()` crashed any code path that touched lineage.

## Decision

Build a **custom Postgres image** that layers Apache AGE onto the pgvector
image, rather than choosing one extension's stock image over the other.

`infra/postgres/Dockerfile`:

```dockerfile
FROM pgvector/pgvector:pg16
RUN apk add --no-cache --virtual .age-build build-base clang llvm-dev git coreutils \
 && git clone --depth 1 --branch PG16 https://github.com/apache/age /age-src \
 && cd /age-src && make PG_CONFIG=/usr/local/bin/pg_config install \
 && cd / && rm -rf /age-src && apk del .age-build
```

Both `infra/docker-compose.dev.yml` and `infra/docker-compose.yml` now
`build: ./postgres` with image tag `genbi/postgres-pgvector-age:pg16`.

Defense-in-depth: AGE is optional at boot

`init.sql` guards the extension so a missing AGE degrades gracefully instead of
crashing DB init — controlled by `GENBI_ENABLE_AGE` (default `true`). This
keeps dev fast (AGE build is the slowest image layer) and lets a contributor
without AGE still bring up the rest of the stack.

Alternatives considered

1. **`apache/age:PG16` + build pgvector into it** — symmetric to the chosen
   approach, but AGE's upstream image is less frequently maintained and the
   pgvector build path is less battle-tested. Rejected.
2. **Drop AGE, use recursive CTEs for lineage** — would lose the property-graph
   query model (multi-hop impact analysis) that the graph layer is built around.
   Rejected; lineage is a documented capability.
3. **Run AGE in a separate Postgres instance** — doubles operational surface
   (two DBs, two backup strategies, cross-DB joins impossible). Rejected.

Consequences

- **Build time:** the first `docker compose build` compiles AGE from source
  (~2–3 min). Cached thereafter.
- **Maintenance:** the AGE git branch (`PG16`) must track AGE releases; bumping
  Postgres major versions requires rebuilding AGE against the new headers.
- **Roles:** `init.sql` grants `ag_catalog` usage to `PUBLIC` (dev convenience).
  Production should restrict this to a dedicated `age_admin` role.
- **Verification:** `infra/postgres/Dockerfile` ends with a
  `RUN test -f $(pg_config --pkglibdir)/age.so` so a broken AGE release fails
  the image build, not a runtime boot.
```

docs/adr/006-enforced-rls-roles.md

69 lines · 3.7 KB

```
# ADR 006: Enforced tenant isolation via dedicated database roles

- **Status:** Accepted (2026-08-15)
- **Context:** Phase 8b — Multi-tenant RLS enforcement
- **Related:** `infra/postgres/init.sql`, Alembic revisions `0002_app_roles` and
  `0003_analytics`, `backend/app/db/session.py`, `docs/data-layer.md`

## Context

Since Phase 6 the schema has had `ENABLE`/`FORCE ROW LEVEL SECURITY` plus a
`tenant_isolation` policy on every tenant-scoped table, keyed on the
`app.current_tenant_id` session GUC that `PostgreSQLConnector` sets per
transaction. The fine print: **RLS never constrains superusers**, and the
backend connected as `POSTGRES_USER` (`genbi`) — a superuser in the official
Postgres image. Tenant isolation was configured but never enforced; any code
path (or SQL injection) that skipped the GUC would read cross-tenant data.

A second gap: the 10 analytics tables that chat queries actually read
(`sales`, `orders`, …) were not RLS-enrolled at all — only the 5 metadata
tables were — so the connector's GUC had no effect on query results.

The login path added in Phase 8a complicated the fix: login must find a user
by email *across* tenants (to support per-tenant credentials and
disambiguation), which a per-tenant RLS policy forbids by design.

## Decision

1. **Two dedicated non-superuser LOGIN roles** (Alembic `0002`, mirrored in
   `init.sql` for fresh volumes; fixed dev passwords, rotate in production):
   - `genbi_app` — the runtime role for the ORM engine and the query
     connector (`DATABASE_URL`). No superuser, no `BYPASSRLS`, no `CREATE`;
     granted exactly the DML it needs on the tenant tables and `SELECT` on
     analytics tables. Every RLS policy applies to it.
   - `genbi_auth` — the login endpoint's role (`DATABASE_URL_AUTH`, derived
     from `DATABASE_URL` when unset). `SELECT` on `users` only, plus a
     permissive RLS policy scoped `TO genbi_auth`
     (`users_login_lookup ... USING (true)`) — the single, auditable
     carve-out that a credential lookup requires. It can read nothing else
     and write nothing.
2. **Analytics tables join the RLS regime** (Alembic `0003`): the 10 seed
   tables (renamed `users` → `web_users` to stop shadowing the app-login
   table) get the same FORCE RLS + `tenant_isolation` policy. The connector
   already set the GUC, so chat queries keep working — but a missing GUC now
   yields zero rows instead of cross-tenant data.
3. **Owner credentials are exiled to `DATABASE_URL_SYNC`**, used only by
   Alembic and admin scripts (`scripts/db_admin.py`). Those scripts connect
   as the owner *and* set the tenant GUC, because FORCE RLS binds the owner
   too.
4. **Enforcement is tested, not assumed**: `tests/api/test_tenant_isolation.py`
   connects with the real roles and asserts cross-tenant invisibility,
   forged-tenant INSERT rejection, no-GUC blindness, and the genbi_auth
   carve-out's exact boundaries. `scripts/verify.sh` checks the same in the
   running stack.

## Consequences

- A compromised or buggy backend can no longer read another tenant's rows by
  forgetting the GUC; it can only see rows the policy allows.
- `genbi_auth` remains a cross-tenant read oracle on `users` by necessity
  (login); the endpoint returns indistinguishable 401s and is the natural
  place for future rate limiting.
- Writing to any tenant table through the ORM now requires setting the GUC
  on the session (no ORM writers exist yet; the isolation tests and
  `db_admin.set_tenant_guc` are the pattern to copy).
- Alembic stays on `DATABASE_URL_SYNC`; migrations that create roles or
  policies must keep running as the owner.
- AGE lineage code (`graph_schema.py`) remains dead code with
  f-string-interpolated Cypher; parameterizing it is a tracked follow-up
  before it can be wired to the runtime role.
```

docs/adr/007-cube-native-semantic-layer.md

70 lines · 3.6 KB

```
# ADR 007: Cube-native semantic layer (dropping the dbt tier)

- **Status:** Accepted (2026-08-15)
- **Context:** Phase 9 — Real metric catalog
- **Supersedes:** the dbt MetricFlow design described in earlier revisions of
  `docs/semantic-layer.md` and `semantic/README.md`
- **Related:** ADR 006 (enforced RLS), `semantic/cube/model/`,
  `backend/app/semantic/cube_client.py`

## Context

The original design was a three-tier pipeline: dbt MetricFlow YAML → dbt
manifest → Cube (via `semantic_layer_sync`) → `CubeClient` → agents. Phase 9
planning found every link of that pipeline non-functional:

1. **dbt was never installed** — no dependency in `pyproject.toml`/`uv.lock`,
   no Dockerfile entry, no CI job, no `profiles.yml`; `dbt parse/run` could
   not execute anywhere.
2. **The bridge was fictional** — `semantic_layer_sync` in `cube.js` is a
   Cube Cloud feature, not Cube Core; the manifest it pointed at could never
   be generated.
3. **The toy metric YAML was invalid** — `revenue.yml` referenced measures
   (`user_id`, `conversions`, `total_users`) that no semantic model defined.
4. **The Cube container was unwired** — no volume mounts at all: it could not
   read `cube.js`, any schema, or any manifest.

Meanwhile the consumer side was real and tested: `CubeClient` (719 lines,
393 test lines locking the `/meta` and `/load` contracts), the ChatService →
NL2SQL metric-context injection, and the per-tenant metric cache.

A second constraint: Phase 8b enforced RLS on all analytics tables.
Cube's `cube_reader` role sees **zero rows** without a per-tenant
`app.current_tenant_id` GUC — so Cube *data* queries need driver-level
tenant work regardless of where metric definitions live.

## Decision

1. **Cube-native data models are the single source of truth.** Ten cubes in
   `semantic/cube/model/` over the seeded analytics tables (23 measures:
   revenue, orders, customers, transactions, web users, activity, deals,
   reps, products, regions). Joins only where seed foreign keys are real.
   Every cube carries a hidden `tenant_id` dimension as the Phase-10 hook.
2. **Delete the dbt scaffold** (`semantic/dbt/`) rather than maintain a
   second, unrunnable definition format.
3. **Wire the Cube runtime properly**: standard `CUBEJS_DB_*` env (read-only
   `cube_reader` role), project mounted at `/cube/conf`, `node`-based
   healthcheck.
4. **Scope data queries out**: `/metrics/list` (metadata) ships now;
   `/metrics/query` waits for per-tenant GUC driver work
   (`contextToOrchestratorId` + connection-init SQL) in Phase 10, together
   with the Explore UI.
5. **Harden the client while here**: real JWT auth (was raw secret —
   dev-mode only), cache the raw `/meta` payload and re-parse on hit (the
   old serialization silently dropped parsed metrics), and rank metrics
   against the user query when the catalog exceeds 20 entries.

## Consequences

- One definition format, one runtime, one test surface; the catalog is
  validated by structural tests in CI without Cube or a database.
- "dbt as source of truth" is gone — accepted because nothing consumed dbt
  artifacts and the repo runs no dbt pipeline. If dbt arrives later (tests,
  freshness checks, staging models), Cube can import from it as an addition,
  not a replacement.
- Metric metadata and metric data have different availability: /meta works
  whenever Cube is up; /load additionally requires the Phase-10 tenant
  driver. `CubeClient.query()` callers must not assume tenant-correct
  results until then.
- The NL2SQL agent now receives a real catalog via the same fail-open path —
  Cube being down degrades prompt quality, never availability.
```

docs/adr/008-per-tenant-cube-data-path.md

63 lines · 3.1 KB

```
# ADR 008: Per-tenant Cube data queries via JWT claims and driver GUC

- **Status:** Accepted (2026-08-15)
- **Context:** Phase 10 — Metrics + Explore
- **Related:** ADR 006 (enforced RLS), ADR 007 (Cube-native semantic layer),
  `semantic/cube/cube.js`, `backend/app/semantic/cube_client.py`

## Context

ADR 007 shipped the metric catalog but deliberately scoped out data
queries: the analytics tables are FORCE-RLS tenant-scoped (ADR 006), and
Cube's `cube_reader` role sets no `app.current_tenant_id` GUC — so `/load`
returned **zero rows** for every tenant. Metadata (`/meta`) was unaffected.
Phase 10 had to make `/api/v1/metrics/query` tenant-correct without
weakening the RLS guarantees.

## Decision

Four links in a chain, each simple:

1. **Backend mints tenant-scoped JWTs.** `CubeClient._auth_token(tenant_id)`
   adds a `tenantId` claim to the standard HS256 token (cached per tenant).
   Cube surfaces JWT claims as `securityContext`. Metadata requests still
   send the anonymous token.
2. **Cube isolates orchestrators by tenant.** `contextToOrchestratorId`
   keys on `securityContext.tenantId`, so each tenant gets its own driver
   pool — connections are never shared across tenants.
3. **The driver sets the RLS GUC.** `driverFactory` builds
   `@cubejs-backend/postgres-driver` (bundled in the cubejs/cube image) from
   `CUBEJS_DB_*` env plus the pg `options` connection parameter:
   `-c app.current_tenant_id=<uuid>` — applied on every pooled connection.
   This is the exact GUC the `tenant_isolation` policies consume.
4. **Postgres RLS remains the enforcement layer.** No tenant filter is
   injected into queries (`queryRewrite` unused): hidden dimensions can't be
   filtered externally, and — decisively — the database, not the query
   builder, decides visibility.

Query results are cached per tenant (`genbi:{tenant}:cube_query:{hash}`,
TTL 300s) in the existing two-tier cache; the cache key covers every
result-affecting parameter.

### Why this is safe to ship without a live Cube to test against

**The failure mode is closed.** If any link in the chain is misconfigured —
claim missing, orchestrator shared, GUC not applied — Postgres RLS returns
zero rows for the query. It can never return another tenant's data, because
the GUC (or its absence) is the only thing that opens the policy, and a
wrong GUC opens nothing. A misconfiguration is *visible* (empty results,
caught by `make verify`'s tenant-scoped query check) rather than *silent*.

## Consequences

- `/api/v1/metrics/query` and the Explore page deliver tenant-correct data
  with isolation enforced at the same layer as every other query path.
- The backend never trusts Cube for authorization — only Postgres.
- Token handling grows a per-tenant cache (bounded: one token per active
  tenant, 1h expiry, refreshed 5 min early).
- `CubeClient.query()` now requires `tenant_id` for meaningful results under
  RLS; the parameter is optional only so metadata-only environments keep
  working.
- The pg `options` mechanism is standard libpq behavior (session GUCs at
  connect), not a Cube-specific extension — if Cube's driver ever changes
  config pass-through, the failure is the closed one above.
```

docs/agent-system.md

289 lines · 12.6 KB

```
# Agent System

> See also: [ADR 002: Multi-Agent Architecture](adr/002-agent-architecture.md) | [Architecture Overview](architecture-overview.md)

## Overview

The GenBI agent system is a **six-agent pipeline** orchestrated by **LangGraph**, inspired by DB-GPT's AWEL (Agentic Workflow Expression Language) patterns. Each agent is a LangGraph node with a typed interface (`BaseAgent` → `AgentResult`). The `RouterAgent` classifies intent and selects which pipeline to run — not all queries need all agents.

**Two-tier LLM usage:**
- **Reasoning tasks** (NL2SQL) → `claude-opus-4` with extended thinking
- **Fast tasks** (classification, chart spec, narrative) → `claude-haiku-4`
- **Safety tasks** (SQL validation, chart validation) → deterministic, zero LLM calls

---

## Base Framework

### AgentConfig

```python
@dataclass
class AgentConfig:
    model_name: str
    temperature: float = 0.0
    max_tokens: int = 4096
    thinking: bool = False  # Enables Claude extended thinking
```

AgentResult

Every agent returns this envelope. The `success` flag is the primary pass/fail indicator.

```python
@dataclass
class AgentResult:
    agent_name: str
    success: bool
    output: dict
    input_tokens: int = 0
    output_tokens: int = 0
    latency_ms: float = 0
    model_version: str = ""
    warnings: list[str] = field(default_factory=list)
    errors: list[str] = field(default_factory=list)

    @property
    def total_tokens(self) -> int:
        return self.input_tokens + self.output_tokens
```

BaseAgent (Abstract Class)

```python
class BaseAgent(ABC):
    name: str = "base_agent"
    description: str = "Base agent class"

    def __init__(self, config: AgentConfig):
        self.config = config
        self._run_id = str(uuid4())

    @abstractmethod
    async def execute(self, **kwargs) -> AgentResult:
        """Execute the agent's core logic. Must be overridden by subclasses."""

    def _timed_result(self, result: AgentResult, start_time: float) -> AgentResult:
        result.latency_ms = (datetime.now(timezone.utc).timestamp() - start_time) * 1000
        return result
```

Contract: All agents accept typed `**kwargs` in `execute()`, return `AgentResult` (never raw exceptions), and call `_timed_result()` to attach latency.

Registry

```python
_registry: dict[str, type[BaseAgent]] = {}

def register_agent(name: str):
    """Decorator to register an agent class in the global registry."""
    def decorator(cls: type[BaseAgent]) -> type[BaseAgent]:
        cls.name = name
        _registry[name] = cls
        return cls
    return decorator

def get_agent(name: str) -> type[BaseAgent] | None: ...
def list_agents() -> list[str]: ...
```

Agents self-register via the `@register_agent("name")` decorator on their class. Import triggers registration — no central file needed.

Agent Roster

1. RouterAgent

```

```
Attribute	Value
Registered as	`"router"`
Purpose	Classify user query intent and return a dispatch plan
LLM	`settings.LLM_FAST_MODEL` (Haiku-4), temp=0, max_tokens=256
Key method	`async def execute(self, query: str, conversation_history: list[dict] | None = None, **kwargs) -> AgentResult`

Intent classification:
Intent	Pipeline
`chat_data`	nl2sql → validation → chart_gen → narrative
`chat_visualize`	chart_gen
`chat_report`	nl2sql → validation → chart_gen → narrative
`chat_knowledge`	narrative (mode: rag)
`chat_explore`	nl2sql (mode: explore)
fallback	nl2sql

The router returns the plan as output — it does NOT invoke downstream agents. The orchestrator (`ChatService`) reads `output["dispatch_plan"]` and executes steps sequentially.

---

2. NL2SQLAgent

Attribute	Value
Registered as	`"nl2sql"`
Purpose	Generate read-only SQL from natural language
LLM	`settings.LLM_REASONING_MODEL` (Opus-4), thinking=True, temp=0, max_tokens=4096
Prompt	`.claude/prompts/nl2sql-system.md` (loaded via `load_prompt`)
Output format	JSON (structured output extraction)

Input kwargs:
Parameter	Type	Required	Description
`query`	`str`	Yes	User's natural language question
`schema_context`	`list[dict] | None`	No	Top-k relevant tables from pgvector
`few_shot_examples`	`list[dict] | None`	No	Similar validated NL/SQL pairs
`metric_definitions`	`list[dict] | str | None`	No	Business metrics from Cube.dev
`tenant_id`	`str`	No	Default `"default"`
`user_id`	`str`	No	For audit
`session_id`	`str`	No	For audit

Output: `AgentResult.output` contains `{"sql": "...", "explanation": "...", "tables_used": [...], "assumptions": [...], "run_id": "..."}`

Safety: Runs `_check_destructive_patterns(sql)` — regex-based detection of 11 destructive SQL patterns (DROP, DELETE, TRUNCATE, ALTER, UPDATE, INSERT, CREATE USER, GRANT, REVOKE, COPY FROM). This is defense-in-depth; `ValidationAgent` runs a more comprehensive check downstream.

---

3. ValidationAgent

Attribute	Value
Registered as	`"validation"`
Purpose	Deterministic SQL safety gate — zero LLM calls
Protection	Marked protected in CLAUDE.md — any relaxation requires security review

Input kwargs: `sql: str | None`, `tenant_id: str = "default"`, `user_roles: list[str] | None`, `connector=None` (optional read connector with `explain(sql)` — ChatService passes one so the EXPLAIN dry-run runs under the RLS tenant GUC)

Seven safety checks (sequential):
1. **Multi-statement injection** — detects `;`-delimited statements outside string literals
2. **Destructive patterns** — 17 hard-blocked (DROP, DELETE, TRUNCATE, ALTER, UPDATE, INSERT, CREATE, GRANT, REVOKE, COPY FROM, ...) + 4 warned (UPDATE/INSERT/CREATE TABLE/CREATE INDEX shapes)
3. **Read-only enforcement** — only `SELECT`, `WITH`, `EXPLAIN`, `SHOW`, `DESCRIBE` pass
4. **Timeout policy** — metadata only (`statement_timeout: "30s"` in output); enforced by the connector via `SET LOCAL` at execution time. The validator never mutates the SQL (a regression test pins this)
5. **Basic sanity** — requires FROM/WHERE (currently always returns True)
6. **EXPLAIN dry-run** — runs `EXPLAIN (FORMAT JSON)` via the connector when the SQL is otherwise valid; fail-open (EXPLAIN problems never block)
7. **Row count estimate** — >1M estimated rows sets `requires_confirmation: true` + an advisory warning; the pipeline then stops before execution until the client re-sends with `confirm_large_query: true` (Phase 13)

Output: `AgentResult.success` is the pass/fail verdict. `output` carries `validated_sql` (clean, unmutated), `statement_timeout`, `explain_plan`, `row_estimate`, `requires_confirmation`. The SSE `validation` event surfaces `valid`, `validated_sql`, `requires_confirmation`, `row_estimate`, `warnings`.

---

4. ChartGenAgent

Attribute	Value
Registered as	`"chart_gen"`
Purpose	Generate chart specifications from data and render via Flint MCP
LLM	`settings.LLM_FAST_MODEL` (Haiku-4), thinking=False, temp=0, max_tokens=2048

Input kwargs: `data: list[dict] | None`, `query: str | None`, `intent: str | None`, `preferred_chart_type: str | None`, `tenant_id: str = "default"`

Pipeline:
1. `_infer_semantic_types(data)` — samples first 20 rows, classifies columns as Category/Quantity/Temporal
2. `_suggest_chart_type(data, semantic_types)` — rule-based: Line if temporal+quantity, Bar if single category, Scatter if 2+ quantities, Pie if few rows
3. `_suggest_encodings(data, semantic_types, chart_type)` — assigns x (temporal > category), y (first quantity), color
4. LLM invocation — produces `ChartAssemblyInput` spec (JSON output, extraction fallbacks)
5. `FlintChartBridge.render(spec)` — renders via Flint MCP or Vega-Lite/Altair fallback

Dependencies: `FlintChartBridge` (bridge to Flint/Vega-Lite), `ChartValidator` (post-render hallucination check)

Chart Subsystem

The chart subsystem has three files:

| File | Purpose |
```

|---|---|
`chart/flint_bridge.py`	`FlintChartBridge` — renders specs via Flint MCP or Altair/Vega-Lite fallback. 31 supported chart types. Auto-detects Flint availability.
`chart/vis_flint_protocol.py`	Extends DB-GPT's GPT-Vis with `vis-flint` markdown code blocks. Parses, renders, replaces with inline SVG/base64 PNG.
`chart/flint_operator.py`	AWEL-compatible operator for DB-GPT Flow Canvas. Wraps ChartGenAgent + FlintChartBridge.

See: [ADR 001](adr/001-dual-chart-stack.md) | [ADR 003](adr/003-vis-flint-protocol.md) | [ADR 004](adr/004-flint-bridge-fallback.md)

---

5. NarrativeAgent

Attribute	Value
Registered as	`"narrative"`
Purpose	Data storytelling — 3-5 sentence insight paragraphs for business stakeholders
LLM	`settings.LLM_FAST_MODEL` (Haiku-4), thinking=False, temp=0.3, max_tokens=512
Prompt	`.claude/prompts/narrative-system.md`

Input kwargs: `query: str`, `data_summary: dict | None`, `chart_context: dict | None`, `sql: str`, `tenant_id: str`, `user_id: str`, `session_id: str`

Quality checks (`_quality_checks(narrative) -> list[str]`):
1. **Length:** 2-8 sentences ideal; warns if shorter or longer
2. **Vague language:** flags "many", "large", "significant", "substantial", "most", "some", "few", "various", "generally", "typically"

---

6. Chart Validator (Deterministic)

Attribute	Value
Purpose	Catch and correct LLM hallucinations in chart specifications — zero LLM calls
Philosophy	Correction over rejection — prefers fuzzy-match fix over failing the request

Validation categories:

Function	What it checks
`validate_spec_structure`	Missing required keys, empty data, no encodings
`validate_data_size`	>10K rows (warn), 0 rows (error)
`validate_field_existence`	Encoding fields match actual data columns; fuzzy-match correction
`validate_chart_type_compatibility`	Type compatibility per chart-type requirements
`validate_data_ranges`	Negative business metrics, axis bounds vs data range
`validate_null_handling`	>20% null values in an encoding field

Correction fallback chain: Pie → Bar, Scatter → Line → Bar, Line → Bar, Heatmap → Table, universal → Bar.

Pipeline Data Flow

```
User Query
  │
  ├ RouterAgent._classify_intent(query)    [Haiku, no thinking]
  │    └ Returns: "chat_data", dispatch=["nl2sql","validation","chart_gen","narrative"]
  │
  ├ CubeClient.get_agent_context(query)    [Semantic layer — cached metric definitions]
  │
  ├ NL2SQLAgent.execute(query, ...)        [Opus, thinking, structured JSON]
  │    └ Output: {sql, explanation, tables_used, assumptions}
  │
  ├ ValidationAgent.execute(sql)           [Deterministic — 7 checks]
  │    └ Output: {validated_sql, is_valid}
  │
  ├ PostgreSQLConnector.execute(sql)       [Read-only, 30s timeout]
  │    └ Output: rows (list[dict])
  │
  ├ ChartGenAgent.execute(data=rows)       [Haiku, no thinking]
  │    ├ _generate_chart_spec() → spec
  │    ├ FlintChartBridge.render(spec) → SVG/PNG
  │    └ ChartValidator.validate_chart_spec(spec, data) → corrected spec
  │    └ Output: {chart_spec, image_base64/svg}
  │
  └ NarrativeAgent.execute(...)            [Haiku, temp=0.3, max_tokens=512]
       └ Output: {narrative}
```

LLM Model Usage Summary

Agent	Model	Thinking	Temperature	Max Tokens	Prompt
RouterAgent	Haiku-4	No	0.0	256	Inline
NL2SQLAgent	Opus-4	**Yes**	0.0	4096	`.claude/prompts/nl2sql-system.md`
ValidationAgent	*None*	N/A	N/A	N/A	N/A
ChartGenAgent	Haiku-4	No	0.0	2048	Inline
NarrativeAgent	Haiku-4	No	0.3	512	`.claude/prompts/narrative-system.md`
ChartValidator	*None*	N/A	N/A	N/A	N/A

Design Principles

1. **Two-tier LLM usage** — reasoning tasks use Opus with thinking; fast tasks use Haiku
2. **Deterministic safety gates** — SQL and chart validation run zero LLM calls
3. **Defense in depth** — NL2SQLAgent has its own destructive-pattern check; ValidationAgent runs independent checks downstream
4. **Registry pattern** — agents self-register via decorator; no central registration file
5. **Prompt-as-code** — system prompts live in `.claude/prompts/*.md`, versioned in git
6. **Structured output** — NL2SQL and Chart Gen both request JSON output with extraction fallbacks
7. **Correction over rejection** — Chart Validator prefers fixing hallucinations over failing

docs/api-reference.md

421 lines · 13.8 KB

```
# API Reference

> **Base URL:** `http://localhost:8000/api/v1` | **Auth:** JWT Bearer token | **Content-Type:** `application/json`

## Authentication

All endpoints except health checks require a JWT Bearer token in the `Authorization` header:

```
Authorization: Bearer <token>
```

The token is obtained via `POST /auth/login` (below).

Token payload:
```json
{
  "sub": "<user_id>",
  "tenant_id": "<tenant_id>",
  "roles": ["user"],
  "exp": 1234567890,
  "iat": 1234567890
}
```

On failure (missing, expired, or invalid token): HTTP 401 with:
```json
{"code": "INVALID_TOKEN", "message": "Missing, expired, or invalid authentication token"}
```

Implementation status key: = fully implemented,   = stub

Auth

`POST /auth/login`
  Authenticate a user by email + password and receive a JWT access token.

Request: `LoginRequest`
```json
{
  "email": "admin@genbi.local",
  "password": "admin123"
}
```

Field	Type	Constraints
`email`	`string`	Required, 3–255 chars (normalized to lowercase)
`password`	`string`	Required, 6–128 chars
`tenant_id`	`string | null`	Optional UUID — required only when the same email exists in multiple tenants

Response: `LoginResponse`
```json
{
  "access_token": "eyJhbGciOiJIUzI1NiIs...",
  "token_type": "bearer",
  "user": {
    "id": "00000000-0000-0000-0000-000000000101",
    "email": "admin@genbi.local",
    "name": "admin",
    "tenant_id": "00000000-0000-0000-0000-000000000001",
    "roles": ["admin", "user"]
  }
}
```

Field	Type	Notes
`access_token`	`string`	JWT, signed with `JWT_SECRET_KEY`; accepted by every protected endpoint
`user.name`	`string`	Derived from the email local-part (the `users` table has no name column)

Errors: `401` with `{"code": "INVALID_CREDENTIALS", "message": "Invalid email or password"}` for wrong credentials, unknown email, ambiguous multi-tenant email, or an invalid `tenant_id`. `429` with `{"code": "TOO_MANY_ATTEMPTS", ...}` after 5 failed attempts for the same email within 15 minutes (counter resets on success).

Health

> No auth required.

`GET /health`
  Liveness probe. Returns 200 if the server is running.

Response:
```json
{"status": "ok", "version": "0.1.0"}
```

`GET /health/ready`
  Readiness probe. Pings Postgres and Redis for real; returns 200 only when both are reachable, 503 otherwise (the probe used by Docker healthchecks and `make verify`).

Response:
```json
{"status": "ready", "database": "connected", "redis": "connected", "mcp_flint": "configured"}
```

```
```
```

Chat

> `Authorization: Bearer <token>` required.

`POST /chat`
  Main NL-to-SQL endpoint. Returns SQL, chart spec, and narrative in a single response.

Request: `ChatRequest`
```json
{
  "query": "Show total revenue by region for Q3",
  "conversation_id": null
}
```

Field	Type	Constraints
`query`	`string`	Required, 1–5000 chars
`conversation_id`	`string | null`	Optional UUID for multi-turn
`confirm_large_query`	`bool`	Default false — set true to proceed when the EXPLAIN row estimate exceeds 1M (see below)

Response: `ChatResponse`
```json
{
  "conversation_id": "550e8400-e29b-41d4-a716-446655440000",
  "query": "Show total revenue by region for Q3",
  "sql": "SELECT region, SUM(revenue_amount) as total_revenue FROM transactions WHERE ...",
  "sql_explanation": "This query groups revenue by region using the transactions table...",
  "chart_spec": {"chartType": "Bar Chart", "encodings": {...}},
  "narrative": "Revenue in Q3 was highest in the Northeast region at $4.2M...",
  "chart_image_base64": null,
  "warnings": [],
  "requires_confirmation": false,
  "row_estimate": 1200
}
```

Large-query confirmation: when the validated SQL's EXPLAIN estimate exceeds 1M rows, the response returns early with `requires_confirmation: true`, `row_estimate`, the SQL, and warnings — no data, chart, or narrative. Re-send the same query with `confirm_large_query: true` to execute.

Field	Type	Notes
`conversation_id`	`string`	UUID generated per session
`query`	`string`	Echoes the input query
`sql`	`string | null`	Generated read-only SQL
`sql_explanation`	`string | null`	Plain-English explanation
`chart_spec`	`dict | null`	Flint `ChartAssemblyInput` schema
`narrative`	`string | null`	3–5 sentence insight paragraph
`chart_image_base64`	`string | null`	Rendered chart (PNG)
`warnings`	`string[]`	Advisory messages

`POST /chat/feedback`
  Record thumbs-up/down feedback on a chat response (Phase 15).

Request: `{"session_id": "<uuid>", "score": 1}` — score: `1` up, `-1` down, `0` clear. The `session_id` is the one from the SSE `start` event / sync response (NOT the conversation id).

The score lands on the audit rows of the session (`audit_log.feedback_score`), completing the governance loop.

Errors: `503` `FEEDBACK_UNAVAILABLE` (no matching audit rows or store unreachable), `422` validation.

`POST /chat/stream`
  Streaming variant. Returns SSE events as the agent pipeline progresses.

Request: Same as `POST /chat`.

Response: `text/event-stream` — each event is `data: {json}\n\n`.

SSE Event Types:

Event	Fields	Description
`start`	`conversation_id`, `query`	Pipeline started
`intent`	`intent`, `dispatch_plan`	Router classified the query
`sql`	`sql`, `warnings`	NL2SQL agent generated SQL
`validation`	`valid`, `validated_sql`, `requires_confirmation`, `row_estimate`, `warnings`	Validation gate (EXPLAIN-backed)
`data`	`row_count`, `warnings`	Query executed, data returned
`chart`	`chart_spec`, `image_base64`, `svg`	Chart generated + rendered
`narrative`	`narrative`, `warnings`	Narrative agent wrote insight
`done`	`status`, `warnings`	Pipeline complete

Example SSE stream:
```
data: {"event":"start","conversation_id":"...","query":"Show revenue by region"}

data: {"event":"intent","intent":"chat_data","dispatch_plan":["nl2sql","validation","chart_gen","narrative"]}

data: {"event":"sql","sql":"SELECT region, SUM(revenue_amount) AS total FROM ...","warnings":[]}

data: {"event":"validation","valid":true,"validated_sql":"SELECT region, SUM(revenue_amount) AS total FROM ...","requires_confirmation":false,"row_estimate":1200,"warnings":[]}

data: {"event":"done","status":"complete","warnings":[]}
```

[see more...]

```

**Status values on `done`:** `complete`, `no_sql`, `validation_failed`, `confirmation_required` (>1M-row estimate — re-send with `confirm_large_query: true`), `no_data`, `error`

---
```

Charts

> `Authorization: Bearer <token>` required.

`POST /charts/render`
  Render a chart from a `ChartAssemblyInput` spec. Supports Vega-Lite, ECharts, and Chart.js backends.

Request: `ChartRenderRequest`
```json
{
  "spec": {
    "chartType": "Bar Chart",
    "encodings": {
      "x": {"field": "region"},
      "y": {"field": "revenue_total"}
    },
    "baseSize": {"width": 600, "height": 400},
    "semantic_types": {"region": "Category", "revenue_total": "Quantity"},
    "data": {
      "values": [
        {"region": "Northeast", "revenue_total": 4200000},
        {"region": "West", "revenue_total": 3100000}
      ]
    }
  },
  "backend": "vegalite",
  "format": "png"
}
```

Field	Type	Default	Notes
`spec.chartType`	`string`	—	Bar, Line, Scatter, Pie, Area, etc. (31 supported)
`spec.encodings`	`dict`	—	Field → channel mappings (`x`, `y`, `color`, `size`, etc.)
`spec.baseSize`	`{width, height}`	`{600, 400}`	Output dimensions
`spec.semantic_types`	`dict`	—	Field types: `Category`, `Quantity`, `Temporal`
`spec.data.values`	`object[]`	—	Inline data rows (≤ 100 rows; larger → temp file)
`spec.data.url`	`string`	—	Alternative: file URL for large datasets
`backend`	`string`	`"vegalite"`	`"vegalite"`, `"echarts"`, or `"chartjs"`
`format`	`string`	`"png"`	`"png"` or `"svg"`

Response: `ChartRenderResponse`
```json
{
  "success": true,
  "format": "png",
  "image_base64": "iVBORw0KGgoAAAANSUhEUgA...",
  "svg": null,
  "warnings": [],
  "errors": []
}
```

Conversations

> Multi-turn chat history (Phase 14). Requires `Authorization: Bearer <token>`.

`GET /conversations`
  List the current user's conversations, most recently active first (RLS tenant-scoped + per-user).

Response: `{conversations: [{id, title, created_at, updated_at}], count}`

Errors: `503` `PERSISTENCE_UNAVAILABLE` when the store is unreachable.

`GET /conversations/{conversation_id}/messages`
  A conversation's turns in chronological order (most recent 50 by default). Other tenants' ids return zero messages (RLS).

Response: `{messages: [{role, content, generated_sql?, created_at}], count}`

Errors: `400 INVALID_CONVERSATION` (malformed id), `503 PERSISTENCE_UNAVAILABLE`.

Multi-turn flow: send chat requests with the same `conversation_id`; the NL2SQL agent receives the prior turns as a "Conversation History" prompt section (follow-ups like "now break that down by region" work). The SSE `start` event carries the resolved `conversation_id`; turns persist automatically (fail-open).

Datasources

`GET /datasources`
  List the semantic layer's cubes for the current tenant (from Cube `/meta`).

Response:
```json
{
  "datasources": [
    {"name": "Sales", "title": "Sales", "measures": 4, "dimensions": 4}
  ],
  "count": 1
}
```

Errors: `503` `{"detail": {"code": "CUBE_UNAVAILABLE", ...}}` when Cube is unreachable.

`POST /datasources/test`
   Stub. Targets a different feature (admin-configured external warehouses), not the built-in semantic-layer source.

Metrics

`GET /metrics/list`

  List all metrics from the semantic layer (catalog of 10 cubes / 23 measures).

Response: `{metrics: [{name, title, description, metric_type, cube_name, measure_name, dimensions, time_dimensions}], count}`

`POST /metrics/query`
  Execute a **tenant-scoped** metric query against Cube.dev (ADR 008: the JWT's tenant claim drives Cube's per-tenant driver and its RLS GUC — results are isolated at the database layer).

Request:
```json
{
  "measures": ["Sales.revenue_total"],
  "dimensions": ["Sales.region"],
  "time_dimensions": [{"dimension": "Sales.transaction_date", "granularity": "month"}],
  "limit": 100
}
```

Field	Type	Constraints
`measures`	`string[]`	1–5, must exist in the catalog (`400 INVALID_METRIC` otherwise)
`dimensions`	`string[]?`	≤ 5 group-bys
`time_dimensions`	`[{dimension, granularity, date_range?}]?`	≤ 3; granularity: day/week/month/quarter/year/hour
`filters`	`[{member, operator, values?}]?`	Cube filter syntax
`order`	`[[field, dir]]?`	e.g. `[["Sales.revenue_total", "desc"]]`
`limit` / `offset`	`int`	1–1000 / ≥ 0
`timezone`	`string`	default `UTC`

Response: `{data: [...flattened rows...], annotation, total, query, latency_ms, cached}` — row keys are stripped of cube prefixes. Results are cached per tenant for 300s (same query → `cached: true`).

Errors: `400 INVALID_METRIC`, `503 CUBE_UNAVAILABLE`, `422` validation.

Reports

> Multi-chart reports (Phase 16). Requires `Authorization: Bearer <token>`.

`POST /reports/generate`
  Generate a multi-chart report from a natural language prompt.

LLM-planned (one fast-model call picks 2–4 metrics from the semantic-layer catalog) with deterministic execution: per-section tenant-scoped Cube query → chart render → one overall narrative. Sections that return no data are skipped with a warning; the report persists best-effort.

Request:
```json
{"prompt": "Q3 performance: revenue, pipeline, and active users", "max_sections": 3}
```

Field	Type	Constraints
`prompt`	`string`	Required, 1–2000 chars
`max_sections`	`int`	2–4, default 3

Response: `ReportOut` — `{report_id, title, prompt, summary?, status, created_at, sections: [{position, metric_name, section_title, chart_spec, chart_svg?, data_total?, row_count, narrative?}], warnings}`

Errors: `503` `REPORT_GENERATION_FAILED` (planning/exec failure), `422` validation.

`GET /reports`
  List the current user's reports, newest first.

Response: `{reports: [{id, title, created_at, section_count}], count}`

Errors: `503` `PERSISTENCE_UNAVAILABLE`.

`GET /reports/{report_id}`
  Retrieve a persisted report (RLS tenant-scoped).

Response: same `ReportOut` shape as generate.

Errors: `400` `INVALID_REPORT` (malformed id), `404` `REPORT_NOT_FOUND`, `503` `PERSISTENCE_UNAVAILABLE`.

Error Format

All error responses follow this shape:

```json
{
  "detail": {
    "code": "ERROR_CODE",
    "message": "Human-readable description of what went wrong"
  }
}
```

Common error codes:

Code	HTTP Status	Context
`INVALID_TOKEN`	401	Missing, expired, or invalid JWT
`VALIDATION_FAILED`	422	Request body fails Pydantic validation
`NOT_FOUND`	404	Resource doesn't exist
`INTERNAL_ERROR`	500	Unexpected server error

Pydantic Models (reference)

`ChatRequest`
```python
class ChatRequest(BaseModel):
    query: str = Field(..., min_length=1, max_length=5000)
    conversation_id: str | None = Field(None)
```

```
403 ```
404
405 ### `ChartAssemblyInput`
406 ```python
407 class ChartAssemblyInput(BaseModel):
408     chartType: str
409     encodings: dict  # Record[str, FieldSpec]
410     baseSize: dict   # {"width": int, "height": int}
411     semantic_types: dict[str, str] | None
412     data: dict        # {"values": list[dict]} or {"url": str}
413 ```
414
415 ### `ChartRenderRequest`
416 ```python
417 class ChartRenderRequest(BaseModel):
418     spec: ChartAssemblyInput
419     backend: str = Field(default="vegalite")
420     format: str = Field(default="png")
421 ```
```

docs/api/openapi.yaml

180 lines · 4.0 KB

```yaml
openapi: 3.1.0
info:
  title: GenBI API
  version: 0.1.0
  description: Generative BI Platform API
servers:
  - url: http://localhost:8000/api/v1
    description: Development

paths:
  /health:
    get:
      operationId: health_check
      summary: Liveness probe
      responses:
        "200":
          description: OK
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/HealthResponse"

  /chat:
    post:
      operationId: chat
      summary: Natural language to SQL + chart + narrative
      security:
        - BearerAuth: []
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: "#/components/schemas/ChatRequest"
      responses:
        "200":
          description: Chat response with SQL, chart, and narrative
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/ChatResponse"

  /charts/render:
    post:
      operationId: render_chart
      summary: Render a chart via Flint MCP
      security:
        - BearerAuth: []
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: "#/components/schemas/ChartRenderRequest"
      responses:
        "200":
          description: Rendered chart
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/ChartRenderResponse"

components:
  securitySchemes:
    BearerAuth:
      type: http
      scheme: bearer

  schemas:
    HealthResponse:
      type: object
      properties:
        status:
          type: string
        version:
          type: string

    ChatRequest:
      type: object
      required: [query]
      properties:
        query:
          type: string
          minLength: 1
          maxLength: 5000
        conversation_id:
          type: string
        confirm_large_query:
          type: boolean
          default: false
          description: Explicit confirmation to run a query whose EXPLAIN row estimate exceeds 1M rows

    ChatResponse:
      type: object
      properties:
        conversation_id:
          type: string
        query:
          type: string
        sql:
```

```yaml
          type: string
        sql_explanation:
          type: string
        chart_spec:
          type: object
        narrative:
          type: string
        chart_image_base64:
          type: string
        warnings:
          type: array
          items:
            type: string
        requires_confirmation:
          type: boolean
          description: True when the query needs explicit confirmation before execution
        row_estimate:
          type: integer
          nullable: true
          description: EXPLAIN-estimated row count, when available

    ChartRenderRequest:
      type: object
      required: [spec]
      properties:
        spec:
          $ref: "#/components/schemas/ChartAssemblyInput"
        backend:
          type: string
          enum: [vegalite, echarts, chartjs]
          default: vegalite
        format:
          type: string
          enum: [png, svg]
          default: png

    ChartRenderResponse:
      type: object
      properties:
        success:
          type: boolean
        format:
          type: string
        image_base64:
          type: string
        svg:
          type: string
        warnings:
          type: array
          items:
            type: string
        errors:
          type: array
          items:
            type: string

    ChartAssemblyInput:
      type: object
      required: [chartType, data]
      properties:
        chartType:
          type: string
        encodings:
          type: object
        baseSize:
          type: object
          properties:
            width:
              type: integer
            height:
              type: integer
        semantic_types:
          type: object
        data:
          type: object
          properties:
            values:
              type: array
            url:
              type: string
```

docs/architecture-overview.md

146 lines · 8.4 KB

Architecture Overview

> See also: [ADR 002: Multi-Agent Architecture](adr/002-agent-architecture.md) | [ADR 001: Dual Chart Stack](adr/001-dual-chart-stack.md)

Four-Tier Architecture

```
┌──────────────────────────────────────────┐
│  Layer 1: Interface                      │
│  Next.js 15 App Router · shadcn/ui · SSE  │
├──────────────────────────────────────────┤
│  Layer 2: Orchestration (LangGraph Agents) │
│   ┌─────────┐ ┌─────────┐ ┌──────────┐  │
│   │ NL2SQL  │ │ ChartGen│ │ Narrative│  │
│   │ Agent   │ │ Agent   │ │ Agent    │  │
│   └─────────┘ └─────────┘ └──────────┘  │
├──────────────────────────────────────────┤
│  Layer 3: Semantic Layer                 │
│  dbt MetricFlow → Cube.dev → CubeClient  │
├──────────────────────────────────────────┤
│  Layer 4: Data Retrieval                 │
│  PostgreSQL · pgvector · Apache AGE      │
├──────────────────────────────────────────┤
│  Layer 5: Governance & Audit             │
│  AuditLog · Langfuse · Prompt logs       │
└──────────────────────────────────────────┘
```

Layer 1 — Interface

- **Framework:** Next.js 15 App Router, React 19, TypeScript 5.9
- **Styling:** Tailwind CSS v4 with a custom `brand` palette, shadcn/ui component primitives
- **Streaming:** SSE (`text/event-stream`) from backend agent pipeline to chat UI
- **Auth:** JWT-based with `AuthProvider` + `AuthGuard` wrapping the chat interface
- **Charts:** Rendered as SVG (inline `dangerouslySetInnerHTML`) or base64 PNG from Flint/Altair bridge

See: [Frontend Guide](frontend-guide.md)

Layer 2 — Orchestration

Six specialized agents orchestrated by LangGraph, inspired by DB-GPT's AWEL patterns:

Agent	Input	Output	Model
`RouterAgent`	User query	Intent + dispatch plan	Haiku-4
`NL2SQLAgent`	Query + schema context + metrics	SQL + explanation + table list	Opus-4 (thinking)
`ValidationAgent`	Generated SQL	Safety verdict + validated SQL	Deterministic
`ChartGenAgent`	Result set + intent	Flint ChartAssemblyInput spec + rendered image	Haiku-4
`NarrativeAgent`	Data summary + chart context	3-5 sentence insight paragraph	Haiku-4
`ChartValidator`	Chart spec + data	Validation issues + corrected spec	Deterministic

The pipeline is conditionally routed: not every query needs all agents. The `RouterAgent` classifies intent (`chat_data`, `chat_visualize`, `chat_report`, `chat_knowledge`, `chat_explore`) and the orchestrator skips irrelevant stages.

See: [Agent System](agent-system.md)

Layer 3 — Semantic Layer

Three-tier pipeline ensuring metrics are defined once and consumed everywhere:

1. **dbt MetricFlow** (`semantic/dbt/`) — Source-of-truth metric definitions, staging models, dimension/measure/entity specifications
2. **Cube.dev** (`semantic/cube/`) — REST + GraphQL API exposing those metrics to agents and frontend
3. **CubeClient** (`backend/app/semantic/`) — Async Python client with metric-to-LLM formatting

Column descriptions from dbt `schema.yml` are injected into LLM context. Schema embeddings are synced to pgvector nightly via `scripts/embed_schema.py`.

See: [Semantic Layer](semantic-layer.md)

Layer 4 — Data Retrieval

- **Primary store:** PostgreSQL 16 with extensions: `pgvector` (schema embeddings, semantic search), `apache_age` (graph queries for lineage/impact analysis)
- **Connectors:** `BaseConnector` → `PostgreSQLConnector` (read-only). All queries are read-only, capped at 30s via `SET LOCAL statement_timeout`, and injected with `SET TRANSACTION READ ONLY`
- **Cache:** Dual-tier — L1 (in-memory LRU, 1000 entries) + L2 (Redis, shared across instances)
- **Session:** Async SQLAlchemy with `asyncpg` driver

See: [Data Layer](data-layer.md) | [Core Services](core-services.md#4-cache-service)

Layer 5 — Governance & Audit

Every LLM call produces an `AuditLog` entry: session_id, user_id, tenant_id, hashed prompt, generated SQL, model name/version, token counts, latency. Langfuse provides LLM tracing. OpenTelemetry spans wrap agent calls and DB queries. Prometheus metrics export via `/metrics`.

Data Flow (Full Pipeline)

```
User Query (ChatView)
  → POST /api/v1/chat/stream (SSE)
    → ChatService.process_query_stream()
      ├─ RouterAgent        → intent: "chat_data"
      ├─ CubeClient         → metric definitions for LLM context
      ├─ NL2SQLAgent        → SQL (Opus-4, thinking, temp=0)
      ├─ ValidationAgent    → safety check (deterministic)
      ├─ PostgreSQLConnector → read-only execution (30s timeout)
      ├─ ChartGenAgent      → ChartAssemblyInput + Flint/Altair render (Haiku-4)
      ├─ ChartValidator     → hallucination detection + auto-correct
      └─ NarrativeAgent     → insight text (Haiku-4, temp=0.3)
  → SSE events: intent → sql → validation → data → chart → narrative → done
```

```
  → ChatView renders incrementally (stage badges, SQL block, chart card, narrative)
```

Project Structure

```
genbi/
├── backend/
│   ├── app/
│   │   ├── agents/        ← LangGraph agent definitions
│   │   │   ├── chart/     ← Flint bridge, AWEL operator, vis-flint protocol
│   │   │   ├── narrative/
│   │   │   ├── nl2sql/
│   │   │   └── validation/ ← SQL safety gate + chart validator
│   │   ├── api/v1/        ← FastAPI routers (versioned)
│   │   ├── connectors/    ← SQLAlchemy DB connectors (read-only)
│   │   ├── core/          ← config, auth, cache, llm_client, masking, observability
│   │   ├── db/            ← ORM models, session, graph schema, migrations
│   │   ├── models/        ← Pydantic schemas (chat, chart)
│   │   ├── semantic/      ← CubeClient (metrics API)
│   │   └── services/      ← ChatService (pipeline orchestrator)
│   └── tests/
├── frontend/
│   └── src/
│       ├── app/           ← Next.js App Router pages
│       ├── components/    ← shadcn/ui + chat + auth + charts
│       ├── lib/           ← api-client, validators (Zod), shadcn utils
│       └── types/         ← TypeScript interfaces
├── semantic/
│   ├── dbt/               ← MetricFlow metrics + staging models
│   └── cube/              ← Cube.dev schema
├── infra/
│   ├── docker-compose.yml
│   ├── docker-compose.dev.yml
│   └── postgres/init.sql
├── docs/                  ← documentation (you are here)
│   └── adr/               ← Architecture Decision Records
└── scripts/               ← embed_schema.py, seed_test_data.py
```

Key Design Decisions

Decision	ADR	Rationale
Multi-agent over monolithic	[002](adr/002-agent-architecture.md)	Specialized prompts, per-stage model selection, independent testing
LangGraph over raw chains	[002](adr/002-agent-architecture.md)	Conditional routing, checkpointing, per-agent streaming
Flint MCP + AntV G2 dual stack	[001](adr/001-dual-chart-stack.md)	Flint for static export, G2 for interactive dashboards
Vega-Lite (Altair) fallback	[004](adr/004-flint-bridge-fallback.md)	Ship without Flint dependency; same schema regardless of renderer
Deterministic safety gates	N/A	SQL validation and chart validation use pure rules — zero LLM calls, zero hallucination risk
Correction over rejection	N/A	Chart validator prefers fuzzy-match fix + chart-type downgrade over failing the request
Read-only by default	N/A	All connectors enforce read-only; writes require explicit service methods with auth

docs/core-services.md

252 lines · 9.8 KB

Core Services

> Backend infrastructure: configuration, logging, LLM client, caching, authentication, PII masking, observability.

---

1. Configuration

File: `backend/app/core/config.py` | **Type:** `pydantic-settings.BaseSettings`

Singleton: `settings = Settings()` — imported project-wide. Loads from `.env` with `extra="ignore"`.

Category	Key	Default	Purpose
App	`APP_ENV`	`"development"`	`development` / `staging` / `production`
App	`DEBUG`	`True`	Debug mode toggle
App	`LOG_LEVEL`	`"DEBUG"`	loguru log level
App	`CORS_ORIGINS`	`["http://localhost:3000"]`	Allowed CORS origins
LLM	`ANTHROPIC_API_KEY`	`""`	Anthropic API key
LLM	`LLM_REASONING_MODEL`	`"claude-opus-4"`	Used for NL2SQL (thinking enabled)
LLM	`LLM_FAST_MODEL`	`"claude-haiku-4"`	Used for routing, charts, narrative
Database	`DATABASE_URL`	`postgresql+asyncpg://genbi:genbi@localhost:5432/genbi`	Async connection
Database	`DATABASE_URL_SYNC`	`postgresql+psycopg2://genbi:genbi@localhost:5432/genbi`	Sync (Alembic)
Redis	`REDIS_URL`	`redis://localhost:6379/0`	Cache + session store
Semantic	`CUBE_API_URL`	`http://localhost:4000/cubejs-api/v1`	Cube.dev REST endpoint
Semantic	`CUBE_API_SECRET`	`""`	Cube API auth
Obs.	`LANGFUSE_SECRET_KEY`	`""`	Langfuse LLM tracing
Obs.	`LANGFUSE_PUBLIC_KEY`	`""`	Langfuse public key
Obs.	`LANGFUSE_HOST`	`"https://cloud.langfuse.com"`	Langfuse instance
Auth	`JWT_SECRET_KEY`	`"change-me"`	HS256 signing key
Auth	`JWT_ALGORITHM`	`"HS256"`	JWT algorithm
Auth	`JWT_EXPIRE_MINUTES`	`60`	Token lifetime
Tenant	`TENANT_ENCRYPTION_KEY`	`""`	PII encryption key
Flint	`FLINT_MCP_BACKENDS`	`"vegalite,echarts,chartjs"`	Enabled chart backends
Flint	`FLINT_MCP_DATA_ROOTS`	`"/tmp/genbi-charts"`	Temp data directory

---

2. Logging

File: `backend/app/core/logging.py`

Uses **loguru** (not stdlib logging). A single shared `logger` instance.

```python
from app.core.logging import logger
logger.info("LLM call complete", model="claude-opus-4", latency_ms=1200)
```

`setup_logging()` — called once at app startup:
- **Development:** colored stdout at `settings.LOG_LEVEL`
- **Production:** adds daily-rotated JSON file output to `logs/genbi_*.json`, 30-day retention

---

3. LLM Client

File: `backend/app/core/llm_client.py`

`LLMClient` (singleton)

All agent LLM calls route through `get_llm_client()`.

Primary method:
```python
async def invoke(
    self,
    messages: list[dict],
    system: str,
    use_reasoning: bool = False,
    options: LLMCallOptions | None = None,
    user_id: str = "",
    tenant_id: str = "",
    session_id: str = "",
    generated_sql: str = "",
) -> LLMCallResult
```

Behavior:
- `use_reasoning=True` → selects `LLM_REASONING_MODEL` (Opus-4), forces `thinking=True` and `temp=0.0`
- `use_reasoning=False` → selects `LLM_FAST_MODEL` (Haiku-4)
- **Retry:** up to 3 attempts, exponential backoff with jitter (`min(2^attempt + random, 30)` seconds)
- **JSON extraction:** three strategies — pure JSON, `` ```json `` code block, first `{...}` pair
- **Token budget:** warns via logger if `input+output > token_budget` (default 100K)
- **Audit logging:** calls audit callback after each successful invocation (SHA-256 of the prompt text — never raw — user/tenant/session context, tokenn counts, latency, generated SQL). Since Phase 12 the callback is wired at app startup to `app/services/audit.py::write_audit_entry`, which persists one `audit_log` row per LLM call (asyncpg on `genbi_app`, tenant GUC set per row, fail-open).

`LLMCallOptions`
```python
@dataclass
class LLMCallOptions:
    temperature: float = 0.0
    max_tokens: int = 4096
    thinking: bool = False
    response_format: str | None = None   # "json" triggers structured output extraction
    timeout_seconds: int = 60
    max_retries: int = 3
    token_budget: int = 100_000
```

`load_prompt(name: str) -> str`
Reads from `.claude/prompts/{name}.md` relative to project root.

Dependencies: `langchain_anthropic.ChatAnthropic`, `settings.ANTHROPIC_API_KEY`.

---

4. Cache Service

File: `backend/app/core/cache.py` | **Architecture:** Dual-tier (L1 in-memory LRU + L2 shared Redis)

Cache TTLs (`CacheTTL`)

Constant	Value	Purpose
`SCHEMA_EMBEDDINGS`	24h	Nightly-synced from information_schema
`METRIC_DEFINITIONS`	5 min	Cube.dev `/meta` response
`QUERY_RESULTS`	5 min	Read-heavy analytics
`LLM_RESPONSE`	1h	Same query = same SQL at temp=0
`CHART_SPEC`	1h	Vega-Lite/ECharts specs
`RATE_LIMIT`	1 min	Window counters
`SESSION_STATE`	30 min	Idle session expiry

`LRUCache` (Level 1)

In-memory `OrderedDict`-based. Max 1000 entries. TTL-based expiry on `get()`. LRU eviction on `set()` when over capacity.

`RedisCache` (Level 2)

Lazy-init via `redis.asyncio.from_url(settings.REDIS_URL)`. If Redis is unavailable, all operations become no-ops (graceful degradation).

`CacheService` (Unified facade)

Typed get/set pairs (all async):

Method pair	Key pattern	Tier
`get_schema_context` / `set_schema_context`	`genbi:{tenant}:schema:{query_hash}`	L1+L2
`get_metric_definitions` / `set_metric_definitions`	`genbi:{tenant}:metrics:all`	L1+L2
`get_query_result` / `set_query_result`	`genbi:{tenant}:query:{sql_hash}`	L1+L2
`get_llm_response` / `set_llm_response`	`genbi:{tenant}:llm:{prompt_hash}`	L2 only
`get_chart_spec` / `set_chart_spec`	`genbi:{tenant}:chart:{data_hash}`	L1+L2
`get_session` / `set_session`	`genbi:session:{session_id}`	L2 only
`check_rate_limit`	`genbi:ratelimit:{user}`	L2 only

Read path: L1.get() → hit? return. Miss → L2.get() → hit? promote to L1 → return. Miss → return None.

Rate limiting: `check_rate_limit(user_id, max_requests=30, window_seconds=60) → (allowed: bool, remaining: int)`. Uses Redis INCR + EXPIRE. Falls open (allows all) if Redis is down.

---

5. Authentication

File: `backend/app/core/auth.py`

JWT Creation and Verification

```python
from app.core.auth import create_access_token, decode_token

token = create_access_token(
    user_id="550e8400-e29b-41d4-a716-446655440000",
    tenant_id="00000000-0000-0000-0000-000000000001",
    roles=["user", "analyst"],
    expires_minutes=60
)

payload = decode_token(token)
# payload = {"sub": "...", "tenant_id": "...", "roles": [...], "exp": ..., "iat": ...}
```

`get_current_user` Dependency

FastAPI `Depends(get_current_user)` extracts the Bearer token from `HTTPBearer`, decodes it with `python-jose`, and returns the payload dict. On failure: HTTP 401.

Used by: All API routes except health checks (see [API Reference](api-reference.md)).

Login endpoint

End users obtain a token via `POST /auth/login` (see [API Reference](api-reference.md#auth)) — bcrypt password check against the `users` table through the login-only `genbi_auth` DB role (its cross-tenant SELECT is the single RLS carve-out; see [ADR 006](adr/006-enforced-rls-roles.md)). Failed attempts are rate-limited in Redis (5 failures → 429 for 15 minutes).

---

6. PII Masking

File: `backend/app/core/masking.py`

`PIIMasker`

Column-level masking for PII fields. Per-tenant, role-aware.

```python
masker = PIIMasker()  # uses DEFAULT_MASK_RULES
masked_rows = masker.mask_rows(rows)
```

Pre-configured rules for 14 column name patterns: `email`, `phone`, `ssn`, `credit_card`, `password`, `secret`, `api_key`, `token`, etc.

Mask strategies:
| MaskType | Input | Output |

|---|---|---|
`EMAIL`	`jdoe@company.com`	`j***@company.com`
`PHONE`	`+1-555-123-4567`	`***-***-4567`
`SSN`	`123-45-6789`	`***-**-6789`
`CREDIT_CARD`	`4111-1111-1111-1111`	`****-****-****-1111`
`FULL_MASK`	anything	`****`

Tenant-aware routing:
```python
load_tenant_mask_rules(tenant_id, custom_rules)
masker = get_masker_for_tenant(tenant_id)
```

---

7. Observability

File: `backend/app/core/observability.py`

Tracing (OpenTelemetry)
- `init_tracing()` — creates OTLP gRPC exporter + console exporter (debug), registers `genbi-backend` service
- `instrument_app(app)` — auto-instruments FastAPI, HTTPX, and SQLAlchemy
- `create_span(name, attributes)` — context manager for manual spans
- `get_current_trace_id()` — returns 32-hex OTel trace ID

Metrics (Prometheus)
`init_metrics()` registers these gauges/counters/histograms on the `genbi` meter:

Metric	Type	Description
`genbi.chat.requests`	Counter	Chat endpoint invocations
`genbi.chat.latency`	Histogram	End-to-end chat latency (ms)
`genbi.sql.generated`	Counter	SQL queries produced
`genbi.sql.validation_failures`	Counter	Failed validation checks
`genbi.llm.tokens`	Counter	LLM token consumption
`genbi.llm.latency`	Histogram	Per-LLM-call latency (ms)
`genbi.cache.hits`	Counter	Cache hit count
`genbi.cache.misses`	Counter	Cache miss count

Langfuse (LLM Tracing)
`LangfuseTracer.trace_llm_call()` creates a Langfuse trace with a sub-generation span. Truncates I/O to 2000 chars. Gracefully degrades if unavailable.

Cross-Cutting Patterns

Graceful degradation: Redis unavailable → no-op caching. Langfuse unavailable → skip tracing. OTel endpoint missing → skip export. Audit callback failure → log and continue.

Singletons: `get_cache()`, `get_llm_client()`, `get_langfuse_tracer()` use module-level lazy-init (no locking — safe under asyncio).

Tenant isolation: All cache keys embed `{tenant_id}`. JWT tokens carry `tenant_id`. Masking rules are per-tenant. Audit logs link to `tenants` via FK.

Async everywhere: All I/O-bound services use `async`/`await`.

docs/data-layer.md

249 lines · 7.8 KB

```
# Data Layer

> ORM models, database sessions, connectors, Apache AGE graph schema, and migrations.

---

## ORM Models

**File:** `backend/app/db/models.py`

### `Base` — SQLAlchemy `DeclarativeBase`

### `Tenant` (table: `tenants`)

| Column | Type | Notes |
|---|---|---|
| `id` | `UUID` | PK, `uuid4()` default |
| `name` | `String(255)` | Unique, not null |
| `created_at` | `TIMESTAMPTZ` | `utcnow()` default |
| `updated_at` | `TIMESTAMPTZ` | `utcnow()` default, onupdate |

### `AuditLog` (table: `audit_log`)

Every LLM call writes an entry — since Phase 12 via `app/services/audit.py`
(the LLMClient audit callback, wired at app startup): asyncpg on the
RLS-bound `genbi_app` role with the tenant GUC set per row, one
parameterized INSERT per call, fail-open (an audit outage never breaks the
pipeline). Schema:

| Column | Type | Notes |
|---|---|---|
| `id` | `UUID` | PK |
| `session_id` | `UUID` | Indexed, not null |
| `user_id` | `UUID` | Not null |
| `tenant_id` | `UUID` | FK → tenants, indexed |
| `input_prompt_hash` | `String(64)` | SHA-256 of prompt (not raw text) |
| `generated_sql` | `Text` | Nullable |
| `model_name` | `String(100)` | e.g. `claude-opus-4` |
| `model_version` | `String(50)` | Model version string |
| `input_tokens` | `Integer` | Default 0 |
| `output_tokens` | `Integer` | Default 0 |
| `latency_ms` | `Float` | Default 0.0 |
| `feedback_score` | `Integer` | Nullable, filled post-hoc |
| `hallucination_flag` | `Boolean` | Default False |
| `created_at` | `TIMESTAMPTZ` | |

### `Conversation` (table: `conversations`)

| Column | Type | Notes |
|---|---|---|
| `id` | `UUID` | PK |
| `user_id` | `UUID` | Not null |
| `tenant_id` | `UUID` | FK → tenants, indexed |
| `title` | `String(500)` | Nullable |
| `created_at` | `TIMESTAMPTZ` | |
| `updated_at` | `TIMESTAMPTZ` | |

---

## Database Session

**File:** `backend/app/db/session.py`

```python
engine = create_async_engine(
    settings.DATABASE_URL,
    echo=settings.DEBUG,
    pool_size=20,
    max_overflow=10,
    pool_pre_ping=True,
)

async_session = async_sessionmaker(
    engine,
    class_=AsyncSession,
    expire_on_commit=False,
)
```

`get_db()` — FastAPI Dependency

```python
async def get_db() -> AsyncSession:
    async with async_session() as session:
        try:
            yield session
        except Exception:
            await session.rollback()
            raise
        finally:
            await session.commit()
```

Connectors

`BaseConnector` (ABC)

File: `backend/app/connectors/base.py`
```

```python
class BaseConnector(ABC):
    name: str = "base"
    _read_only: bool = True     # All connectors enforce read-only

    @abstractmethod
    async def connect(self) -> None: ...
    @abstractmethod
    async def execute(self, sql: str) -> list[dict]: ...
    @abstractmethod
    async def get_schema(self) -> dict: ...
    @abstractmethod
    async def disconnect(self) -> None: ...
```

Every connector is read-only by design. Writes require explicit service methods with authorization.

`PostgreSQLConnector`

File: `backend/app/connectors/postgresql_connector.py` | **Name:** `"postgresql"`

Constructor:
```python
PostgreSQLConnector(
    connection_url=settings.DATABASE_URL,
    schema="public",
    pool_size=10,
    max_overflow=5,
    echo=False,
)
```

`execute(sql, params=None) → list[dict]`:
1. Requires `_connected = True`
2. Rejects non-SELECT queries: validates SQL starts with `SELECT`, `WITH`, `EXPLAIN`, `SHOW`, or `DESCRIBE`
3. Prepends `SET LOCAL statement_timeout = '30s'; SET LOCAL default_transaction_read_only = on;`
4. Executes inside `SET TRANSACTION READ ONLY` block
5. Converts rows to dicts; types converted with `isoformat()` for JSON serialization

`get_schema() → dict`:
Queries `information_schema.tables` + `information_schema.columns` for the configured schema. Includes table/column comments (`obj_description`, `col_description`) and approximate row counts.

`explain(sql) → dict`:
Runs `EXPLAIN (FORMAT JSON)` and extracts `estimated_cost` and `estimated_rows`.

`get_sample_rows(table_name, schema, limit=5) → list[dict]`:
Uses quoted identifiers for read-only sampling.

Connector Registry

File: `backend/app/connectors/registry.py`

```python
_registry: dict[str, type[BaseConnector]] = {}

def register_connector(name: str): ...  # decorator
def get_connector(name: str) -> type[BaseConnector] | None: ...
def list_connectors() -> list[str]: ...
```

Schema Retrieval (NL2SQL grounding, Phase 11)

File: `backend/app/services/schema_retrieval.py` | **Provider:** `backend/app/core/embeddings.py`

The user's query is embedded (OpenAI `text-embedding-3-small`, 1536 dims — matching the `VECTOR(1536)` columns) and matched against the tenant-scoped `schema_embeddings` (top-k tables) and `agent_examples` (top-k validated NL/SQL pairs) tables via cosine distance (`<=>`). Reads run through `PostgreSQLConnector` as the RLS-bound `genbi_app` role with the tenant GUC set, so retrieval is tenant-isolated at the database layer.

Wired in `ChatService._step_nl2sql` with the two-tier cache (`schema:{query_hash}` TTL 86400s, `fewshot:{query_hash}` TTL 3600s) and everything fails open — no key / no embeddings / DB down → empty context. Populate with `PYTHONPATH=backend uv run python scripts/embed_schema.py --examples` (embeds all tables + the 20 golden NL/SQL pairs; needs `OPENAI_API_KEY`; invalidates the schema cache after sync).

Apache AGE Graph Schema

File: `backend/app/db/graph_schema.py` | **Graph name:** `genbi_graph`

Vertex Labels

Label	Description
`TABLE`	Database table or Cube cube
`COLUMN`	Table column or Cube dimension
`METRIC`	Business metric (Cube measure)
`DASHBOARD`	User-created dashboard
`USER`	Platform user (for access control edges)

Edge Labels

Label	Direction	Description
`TABLE_CONTAINS`	TABLE → COLUMN	Table columns
`METRIC_SOURCE`	METRIC → TABLE	Which table a metric derives from
`METRIC_DEPENDS`	METRIC → METRIC	Derived metric dependency chain

```
| `DASHBOARD_USES` | DASHBOARD → METRIC/TABLE | Dashboard data sources |
| `USER_CAN_ACCESS` | USER → TABLE/METRIC | Row-level access control |
| `TABLE_JOINS` | TABLE → TABLE | Known join paths between tables |

### Key Functions

| Function | Purpose |
|---|---|
| `init_age_graph(db_session)` | Idempotent: creates AGE extension + graph |
| `ingest_table(db_session, ...)` | Creates Table + Column vertices with edges |
| `ingest_metric(db_session, ...)` | Creates Metric vertices linked to tables |
| `get_metric_lineage(db_session, metric_name)` | Full lineage path for a metric (all upstream tables) |
| `find_join_paths(db_session, table_a, table_b, max_depth=3)` | Used by NL2SQLAgent to discover join paths |
| `get_downstream_impact(db_session, table_name)` | Impact analysis — what metrics/dashboards are affected |
| `sync_semantic_layer_to_graph(db_session, cube_client)` | Nightly: syncs Cube cubes → AGE vertices |

---

## Alembic Migrations

**File:** `backend/app/db/migrations/env.py`

Standard Alembic configuration:
- Reads `sqlalchemy.url` from `settings.DATABASE_URL_SYNC`
- Uses `Base.metadata` for autogenerate support
- `poolclass=pool.NullPool` for migration safety

```bash
# Create a new migration
uv run alembic revision --autogenerate -m "add new column"

# Apply migrations
uv run alembic upgrade head
```

Database Rules (from CLAUDE.md)

- **Every table MUST have:** `id UUID PK DEFAULT gen_random_uuid()`, `created_at`, `updated_at`, `tenant_id FK`
- **RLS:** every user-data table enables row-level security
- **No dropping columns** — mark `deprecated_at TIMESTAMPTZ` first, remove after 2 sprints
- **Indexes:** all FKs and `tenant_id` columns indexed
- **Query timeout:** 30s via `SET LOCAL statement_timeout`
- **Migrations:** Alembic only — no manual `ALTER TABLE`
```

docs/frontend-guide.md

324 lines · 12.6 KB

```
# Frontend Guide

> Next.js 15 App Router, TypeScript 5.9, Tailwind CSS v4, shadcn/ui components.

## Tech Stack

| Layer | Choice |
|---|---|
| Framework | Next.js 15 (App Router) |
| Language | TypeScript 5.9 (strict) |
| Styling | Tailwind CSS v4 with custom `brand` palette |
| Components | shadcn/ui (Radix UI primitives) |
| Validation | Zod 3.24 |
| Icons | lucide-react |
| Markdown | react-markdown (rehype + remark plugins) |

---

## Project Structure

```
frontend/src/
├── app/
    ├── globals.css          ← Tailwind directives + brand palette
    ├── layout.tsx           ← RootLayout (metadata, AuthProvider)
    ├── page.tsx             ← Landing page (hero + auth-aware CTA)
    ├── login/
    │   └── page.tsx         ← Login page (LoginForm → redirects to /chat)
    ├── chat/
    │   └── page.tsx         ← Chat page (AuthGuard → ChatView)
    └── explore/
        └── page.tsx         ← Explore page (AuthGuard → ExploreView)
├── components/
    ├── auth/
    │   ├── auth-provider.tsx    ← AuthProvider + AuthGuard (redirects to /login)
    │   └── login-form.tsx       ← Email/password sign-in form
    ├── charts/
    │   └── chart-card.tsx       ← ChartCard + ChartGrid
    ├── chat/
    │   └── chat-view.tsx        ← Full ChatView with SSE consumption
    ├── explore/
    │   └── explore-view.tsx     ← Metric catalog + native query builder
    └── ui/                  ← 11 shadcn/ui primitives
├── lib/
    ├── api-client.ts        ← Centralized API client with JWT + SSE
    ├── auth-storage.ts      ← Session keys + localStorage helpers (genbi_token/genbi_user)
    ├── shadcn.ts            ← cn() utility (clsx + twMerge)
    └── validators.ts        ← Zod schemas with inferred types
└── types/
    ├── index.ts             ← Shared types
    └── chart.ts             ← ChartAssemblyInput, enums
```

API Client

File: `src/lib/api-client.ts`

Centralized fetch wrapper. All backend communication routes through this file.

Core Request Function

```typescript
async function request<T>(path: string, options: RequestInit = {}): Promise<T>
```

- Sets `Content-Type: application/json`
- Attaches JWT from `auth-storage` (`genbi_token` key)
- Throws `ApiError(status, code, message)` on non-OK responses
- Default base URL: `http://localhost:8000/api/v1`

`ApiError` Class

```typescript
class ApiError extends Error {
  status: number;
  code: string;
  message: string;
}
```

Exported Functions

```typescript
sendChat(req: ChatRequest): Promise<ChatResponse>              // POST /chat
streamChat(req: ChatRequest, onEvent, onError): AbortController  // POST /chat/stream (SSE)
renderChart(req: ChartRenderRequest): Promise<ChartRenderResponse>  // POST /charts/render
listMetrics(): Promise<MetricListResponse>                    // GET /metrics/list (Phase 10)
queryMetrics(req: MetricQueryRequest): Promise<MetricQueryResponse>  // POST /metrics/query (tenant-scoped)
listDatasources(): Promise<{ datasources: DatasourceSummary[]; count: number }>  // GET /datasources
healthCheck(): Promise<{ status: string; version: string }>     // GET /health
```

`streamChat` — SSE Implementation

Uses `ReadableStream` reader with a line buffer:
- Splits on `\n`
- Strips `data: ` prefix
- Parses JSON via `JSON.parse`
```

- Calls `onEvent(parsed)` for each event
- Invalid lines are silently skipped
- Returns `AbortController` for cancellation

Zod Validators

File: `src/lib/validators.ts`

All schemas use `z.object(...)` with inferred types:

Schema	Key Validations
`ChatRequestSchema`	`query: z.string().min(1).max(2000)`, optional `conversation_id: z.string().uuid()`
`ChatResponseSchema`	All optional except `conversation_id`, `query`, `warnings`
`SSEEventSchema`	`event: z.enum(["start","intent","sql","validation","data","chart","narrative","done"])`
`ChartAssemblyInputSchema`	`chartType: z.string().min(1)`, `encodings`, `baseSize` (positive ints)
`MetricDefinitionSchema`	`metric_type: z.enum([...9 types])`
`LoginRequestSchema`	`email: z.string().email()`, `password: z.string().min(6)`
`LoginResponseSchema`	`access_token`, `token_type`, `user: { id, email, name, tenant_id, roles }`

Chat View (SSE Streaming)

File: `src/components/chat/chat-view.tsx`

The main application component. Full-height flex layout with:

State
```typescript
messages: ChatMessage[];    // User + assistant message pairs
loading: boolean;            // In-flight request indicator
abortRef: AbortController;  // SSE cancellation
```

Pipeline Progress (Stage Badges)

A row of `Badge` components shows real-time pipeline progress: **Intent** → **SQL** → **Validated** → **Results** → **Chart** → **Insight** → **Done**. A pulsing badge appears while the corresponding stage is streaming.

SSE Event Handling

`updateMessageStage(msg, event)` incrementally populates the assistant message:

Event	Populates
`sql`	`msg.sql`
`validation`	`msg.sql` = `validated_sql`
`data`	Row count, data preview
`chart`	`msg.chartSpec`, `msg.chartSvg`, `msg.chartBase64`
`narrative`	`msg.narrative`
`done`	Finalizes content, sets `streaming: false`

Layout

- **Header:** Logo, view toggle ("Chat" / "Metrics"), settings, user avatar, logout
- **Messages area:** Scrolling container. Empty state: centered greeting with three suggestion chips ("Show revenue by region", "Monthly active users trend", "Top 10 customers by value"). User messages: right-aligned, `brand-600` background. Assistant messages: `Card` with stage badges, SQL block (dark terminal style with copy button), chart card, narrative text, amber warning boxes.
- **Input bar:** Fixed-bottom styled `Input` with Send/Cancel button. Disclaimer: "GenBI can make mistakes. Verify important data."

Chart Card

File: `src/components/charts/chart-card.tsx`

Props: `spec: ChartAssemblyInput`, `imageBase64?: string`, `svg?: string`, `title?: string`, `onDownload?: () => void`

Renders within a bordered `Card`:
- **Header:** chart type label + "Flint" badge, format toggle (SVG/PNG), Download button
- **Body:** Conditionally renders:
  - SVG via `dangerouslySetInnerHTML` with `class="vis-flint-chart"`
  - Base64 PNG `<img>` element
  - Placeholder "Chart rendering..." while streaming

`ChartGrid`: Responsive 1-column (mobile) / 2-column (desktop) grid layout for multiple charts.

Auth Provider

File: `src/components/auth/auth-provider.tsx`

Auth Flow

```typescript
interface AuthState {
  user: User | null;
  token: string | null;
  loading: boolean;
  login(email: string, password: string): Promise<void>;
  logout(): void;
  isAuthenticated: boolean;
}
```

- **Login:** POST to `/auth/login`, stores `genbi_token` + `genbi_user` in `localStorage` (via `lib/auth-storage.ts`)
- **Session restoration:** On mount, reads from `localStorage`. If parsing fails, clears both.
- **Loading guard:** While `loading` is true, renders nothing (or bouncing dots in `AuthGuard`).

`AuthGuard` Component

Wraps protected pages. Shows:
- Loading: bouncing dots animation
- Unauthenticated: `router.replace("/login")` redirect
- Authenticated: children

`LoginForm` Component

File: `src/components/auth/login-form.tsx` — extracted, reusable email/password form with `onSuccess` callback. Used by the `/login` page, which navigates to `/chat` after a successful sign-in.

UI Components (shadcn/ui)

File: `src/components/ui/*.tsx` | **Utility:** `cn()` from `@/lib/shadcn`

All 11 components use Radix UI primitives and Tailwind CSS with class-variance-authority (CVA) for variants.

Component	Primitive	Variants / Notes
`Avatar`	`@radix-ui/react-avatar`	Root (40x40, rounded-full), Image, Fallback (gray bg, centered initials)
`Badge`	Plain div	`default` (brand), `secondary`, `destructive`, `outline`, `success`, `warning`. Pill shape, text-xs
`Button`	`@radix-ui/react-slot` (asChild)	`default` (brand+shadow), `destructive`, `outline`, `secondary`, `ghost`, `link`. Sizes: `default`, `sm`, `lg`, `icon`
`Card`	Plain HTML	Compound: Card, CardHeader, CardTitle, CardDescription, CardContent, CardFooter. Rounded-xl border shadow
`Dialog`	`@radix-ui/react-dialog`	Overlay (bg-black/50), Content (centered, rounded-xl), with X close button
`DropdownMenu`	`@radix-ui/react-dropdown-menu`	Content (z-50, min-w-8rem), Items with keyboard shortcuts, Separator
`Input`	Plain `<input>`	h-10, rounded-lg border, focus ring-brand-600
`Separator`	Plain div	Horizontal (`h-px w-full`) or Vertical
`Skeleton`	Plain div	`animate-pulse rounded-md bg-gray-200`
`Tabs`	`@radix-ui/react-tabs`	List (inline-flex, bg-gray-100), Trigger (pill, active=white bg+shadow), Content
`Tooltip`	`@radix-ui/react-tooltip`	Content (bg-gray-900, text-xs, white text, shadow-md)

shadcn.cn Utility

```typescript
import { clsx, type ClassValue } from "clsx";
import { twMerge } from "tailwind-merge";
export function cn(...inputs: ClassValue[]) { return twMerge(clsx(inputs)); }
```

Type Definitions

File: `src/types/chart.ts`

```typescript
export interface ChartAssemblyInput {
  chartType: string;
  encodings: Record<string, { field: string }>;
  baseSize: { width: number; height: number };
  semantic_types?: Record<string, "Category" | "Quantity" | "Temporal">;
  data: { values?: Record<string, unknown>[]; url?: string };
}

export type ChartBackend = "vegalite" | "echarts" | "chartjs";
export type ChartOutputFormat = "png" | "svg";
```

Pages

`layout.tsx` — Root Layout

- HTML metadata: title `"GenBI — Generative Business Intelligence"`
- Applies `globals.css`, wraps children in `AuthProvider`
- Body: `min-h-screen bg-gray-50 text-gray-900 antialiased`

`page.tsx` — Landing Page

Minimal hero (logo, tagline, CTA). The CTA routes to `/login` when signed out and `/chat` when signed in.

`login/page.tsx` — Login Page

Redirects to `/chat` if already authenticated; otherwise renders `LoginForm`, which navigates to `/chat` on success.

`chat/page.tsx` — Chat Page

Wraps `ChatView` in `AuthGuard` (redirects to `/login` when signed out). The header's database icon navigates to `/explore`; the file icon to `/reports`.

`explore/page.tsx` — Explore Page (Phase 10)

Wraps `ExploreView` (`components/explore/explore-view.tsx`) in `AuthGuard`. The semantic-layer workbench:

- **Catalog**: metrics from `GET /metrics/list` as clickable cards (title, cube, `metric_type` badge); selecting one sets the query's measure.
- **Query builder**: native `<select>` elements (no new deps) for measure, group-by dimension (from the measure's dimensions), optional time granularity (day/month via its first time dimension), and row limit.
- **Run**: `POST /metrics/query` → results table (hand-built Tailwind `<table>`, cube-prefix-stripped keys) +, when sliced by a dimension, a bar chart: the client builds a `ChartAssemblyInput` from the result rows, renders via `POST /charts/render`, and displays through the existing `ChartCard`.
- Empty results show an RLS-aware hint (tenant has no data — `make seed`).
- API functions live in `lib/api-client.ts` (`listMetrics`, `queryMetrics`, `listDatasources`); Zod schemas (`MetricQueryRequestSchema` etc.) in `lib/validators.ts`.

`reports/page.tsx` — Reports Page (Phase 16)

Wraps `ReportsView` (`components/reports/reports-view.tsx`) in `AuthGuard`. The multi-chart report workbench:

- **Generator**: prompt input + section-count select (2–4) + Generate → `POST /reports/generate`.
- **Sidebar**: past reports from `GET /reports` (title, section count); select loads via `GET /reports/{id}`.
- **Report display**: title + overall summary + badges, per-section cards (heading with total/row count, `ChartCard` with the persisted SVG, optional narrative), warnings panel (skipped metrics, persistence notice).
- API functions: `generateReport`, `listReports`, `getReport` in `lib/api-client.ts`.

Configuration

`next.config.js`: `output: "standalone"` (containerized deployment)

`tailwind.config.ts`: Custom `brand` palette (indigo tones):
```typescript
brand: {
  50: '#edf2ff', 100: '#dbe4ff', ...,
  600: '#4c6ef5', 700: '#4263eb', ..., 900: '#1e2a8a'
}
```

Build Commands

```bash
pnpm dev          # Next.js dev server (port 3000)
pnpm build        # Production build
pnpm start        # Start production server
pnpm lint         # ESLint
pnpm typecheck    # tsc --noEmit
```

docs/index.md

55 lines · 3.0 KB

```
# GenBI Platform — Development Documentation

> **Version:** 0.1.0 | **Stack Tier:** Enterprise | **Last updated:** 2026-08-15

A **Generative BI (GenBI) platform** that enables business users to query enterprise data warehouses using natural language, auto-generates visualizat
ions and narrative insights, and delivers governed, explainable analytics at scale.

## Quick Links

### Architecture & Design
- [Architecture Overview](architecture-overview.md) — Four-tier architecture, data flow, technology stack
- [Agent System](agent-system.md) — Multi-agent LangGraph pipeline, agent roster, LLM model usage
- [Architecture Decision Records](adr/) — ADR 001–008 covering chart stack, agent topology, Flint bridge, AGE image, enforced RLS roles, Cube-native c
atalog, and per-tenant Cube data path

### API
- [API Reference](api-reference.md) — Complete REST API documentation, request/response schemas, SSE streaming

### Backend Services
- [Core Services](core-services.md) — Configuration, logging, LLM client, caching, auth, PII masking, observability
- [Data Layer](data-layer.md) — Database models, connectors (read-only), Apache AGE graph schema, migrations
- [Semantic Layer](semantic-layer.md) — Cube-native metric catalog → Cube.dev API → Python CubeClient

### Frontend
- [Frontend Guide](frontend-guide.md) — Next.js 15 App Router, shadcn/ui components, SSE chat, Explore metric workbench, Zod validators

### Operations
- [Infrastructure](infrastructure.md) — Docker Compose (dev + prod), CI/CD, environment variables, scripts

### Key Concepts

| Concept | Where |
|---|---|
| NL → SQL → Chart → Narrative pipeline | [Agent System](agent-system.md#pipeline-data-flow) |
| Dual chart stack (Flint MCP + AntV G2) | [ADR 001](adr/001-dual-chart-stack.md) |
| Schema embedding + pgvector semantic search | [Semantic Layer](semantic-layer.md) |
| Read-only SQL enforcement (Safety Gate) | [Agent System](agent-system.md#32-validationagent) |
| Dual-tier cache (L1 LRU + L2 Redis) | [Core Services](core-services.md#4-cache-service) |
| Multi-tenant JWT + RLS isolation | [Core Services](core-services.md#5-authentication) |
| SSE streaming from agent pipeline to chat UI | [API Reference](api-reference.md#streaming) |
| Chart hallucination detection + auto-correction | [Agent System](agent-system.md#35-chart-validator) |

### Tech Stack at a Glance

```
Backend: Python 3.12 · FastAPI · Uvicorn / Gunicorn
LLM Layer: Anthropic Claude (Opus for SQL, Haiku for speed)
               LangChain / LangGraph orchestration
               LlamaIndex for semantic indexing / RAG
Semantic Layer: Cube.dev (native data models — metric catalog)
Databases: PostgreSQL 16 · pgvector (embeddings) · Apache AGE (graph lineage)
               Redis (query cache, session state)
Frontend: Next.js 15 · TypeScript 5.9 · Tailwind CSS v4 · shadcn/ui
Charts: Flint MCP (static) + AntV G2 (interactive) · Vega-Lite / Altair fallback
Observability: OpenTelemetry · Langfuse · Prometheus + Grafana
CI/CD: GitHub Actions · Docker + Docker Compose
```
```

docs/infrastructure.md

245 lines · 8.6 KB

```
# Infrastructure

> Docker Compose, CI/CD, environment variables, and utility scripts.

---

## Docker Compose — Development

**File:** `infra/docker-compose.dev.yml`

| Service | Image / Build | Port | Volumes | Notes |
|---|---|---|---|---|
| **postgres** | `pgvector/pgvector:pg16` | `5432` | `postgres_data` (named) + `./postgres/init.sql` mount | `pg_isready` health check, 5s interval |
| **redis** | `redis:7-alpine` | `6379` | None | `redis-cli ping` health check, 5s interval |
| **backend** | Build from `../backend` | `8000` | `../backend:/app` + `flint_data:/tmp/genbi-charts` | Depends on healthy postgres + redis. `uvicorn --reload` hot-reload |
| **frontend** | Build from `../frontend` | `3000` | `../frontend:/app`, `/app/node_modules` (anonymous) | Depends on backend. `pnpm dev` |
| **cube** | `cubejs/cube:latest` | `4000` | None | Cube.dev semantic API |
| **prometheus** | `prom/prometheus:latest` | `9090` | `./prometheus/prometheus.yml` mount | |
| **grafana** | `grafana/grafana:latest` | `3001` | None | Depends on prometheus |
| **otel_collector** | `otel/opentelemetry-collector-contrib:latest` | `4318` (HTTP), `8888` (metrics) | None | |

### Key differences from production:
- Hot-reload: `uvicorn --reload` (backend), `pnpm dev` (frontend)
- Volume mounts for live code editing
- Health check dependencies between services
- Additional observability services (Prometheus, Grafana, OTel Collector)

---

## Docker Compose — Production

**File:** `infra/docker-compose.yml`

| Service | Image / Build | Port | Volumes | Restart |
|---|---|---|---|---|
| **postgres** | `pgvector/pgvector:pg16` | `5432` | `postgres_data` + `init.sql` | `unless-stopped` |
| **redis** | `redis:7-alpine` | `6379` | None | `unless-stopped` |
| **backend** | Build from `../backend` | `8000` | None | `unless-stopped` |
| **frontend** | Build from `../frontend` (target: `production`) | `3000` | None | `unless-stopped` |

- Backend CMD: `gunicorn` with 4 `uvicorn` workers
- Frontend env: `NEXT_PUBLIC_API_URL` defaults to `http://localhost:8000/api/v1`
- Credentials from `../backend/.env`

### Start Commands

```bash
# Development
docker compose -f infra/docker-compose.dev.yml up -d

# Production
docker compose -f infra/docker-compose.yml up -d

# Tear down (keep volumes)
docker compose down

# Tear down (remove volumes)
docker compose down -v

# Tail logs
docker compose logs -f backend
```

PostgreSQL Initialization

File: `infra/postgres/init.sql`

Extensions
- `pgcrypto` — UUID generation (`gen_random_uuid()`)
- `vector` — pgvector for embedding storage (1536-dim)
- `age` — Apache AGE for graph queries (lineage, impact analysis)

Schema Tables

Table	Purpose	Key columns
`tenants`	Multi-tenant root	`id UUID PK`, `name VARCHAR(255) UNIQUE`, `created_at`, `updated_at`
`users`	Per-tenant users	`id UUID PK`, `tenant_id FK`, `email UNIQUE`, `hashed_password`, `roles JSONB`
`audit_log`	NL2SQL audit trail	`session_id`, `user_id`, `tenant_id`, `input_prompt_hash`, `generated_sql`, `model_name`, `token_count`, `latency_ms`
`conversations`	Chat sessions	`id UUID PK`, `user_id`, `tenant_id`, `title`
`schema_embeddings`	Vector store	`table_schema`, `table_name`, `embedding VECTOR(1536)`, `description`
`agent_examples`	Few-shot storage	`agent_name`, `input_text`, `output_json`, `embedding VECTOR(1536)`

Indexes
- `schema_embeddings` table: IVFFlat index on `vector_cosine_ops` for semantic similarity search
- `audit_log`: indexed by `session_id`, `tenant_id`, `created_at`
- `agent_examples`: indexed by `agent_name` + IVFFlat
- All FKs and `tenant_id` columns indexed

Prometheus Configuration

File: `infra/prometheus/prometheus.yml`

```yaml
```
```

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: "genbi-backend"
    static_configs:
      - targets: ["backend:8000"]
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]
```

CI/CD Pipeline

File: `.github/workflows/ci.yml`

Triggers: Push to `main`/`develop`, PR to `main`

Jobs

1. Backend — Lint & Test (parallel with frontend)
- Python 3.12, `uv`
- PostgreSQL + Redis service containers
- Steps: checkout → `uv sync --frozen` → ruff lint + format check → alembic migrations check → pytest (non-e2e, with coverage) → golden NL2SQL eval → upload coverage to Codecov

2. Frontend — Lint & Build (parallel with backend)
- Node 20, `pnpm` 9
- Steps: checkout → `pnpm install` → `tsc --noEmit` → lint → build

3. Docker — Build Images (sequential, only on push to main/develop)
- Docker Buildx, GHCR login
- Build + push backend and frontend images tagged `:sha` and `:latest`
- GitHub Actions cache for Docker layers

4. Golden Eval Gate (on main branch only)
- PostgreSQL service container
- Runs golden NL2SQL eval suite
- Blocks merges below **90% accuracy threshold**

Environment Variables

File: `.env.example`

Category	Variable	Default
LLM	`ANTHROPIC_API_KEY`	(required)
	`LLM_REASONING_MODEL`	`claude-opus-4`
	`LLM_FAST_MODEL`	`claude-haiku-4`
Database	`DATABASE_URL`	`postgresql+asyncpg://genbi_app:genbi_app@localhost:5432/genbi` (RLS-bound runtime role)
	`DATABASE_URL_AUTH`	(optional) `postgresql+asyncpg://genbi_auth:genbi_auth@…` — login-only role; derived from `DATABASE_URL` if unset
	`DATABASE_URL_SYNC`	`postgresql://genbi:genbi@localhost:5432/genbi` (owner — Alembic + admin scripts only)
Redis	`REDIS_URL`	`redis://localhost:6379/0`
Semantic	`CUBE_API_URL`	`http://localhost:4000/cubejs-api/v1`
	`CUBE_API_SECRET`	(required)
Observability	`LANGFUSE_SECRET_KEY`	
	`LANGFUSE_PUBLIC_KEY`	
	`LANGFUSE_HOST`	`https://cloud.langfuse.com`
	`OTEL_EXPORTER_OTLP_ENDPOINT`	
Auth	`JWT_SECRET_KEY`	`change-me` (override in production!)
	`JWT_ALGORITHM`	`HS256`
	`JWT_EXPIRE_MINUTES`	`60`
Tenant	`TENANT_ENCRYPTION_KEY`	
Flint	`FLINT_MCP_BACKENDS`	`vegalite,echarts,chartjs`
	`FLINT_MCP_DATA_ROOTS`	`/tmp/genbi-charts`
App	`APP_ENV`	`development`
	`DEBUG`	`True`
	`LOG_LEVEL`	`DEBUG`
	`CORS_ORIGINS`	`["http://localhost:3000"]`

Dockerfiles

Backend (`backend/Dockerfile`)
```dockerfile
FROM python:3.12-slim
COPY --from=ghcr.io/astral-sh/uv:latest /uv /usr/local/bin/uv
# Install deps, copy code, expose 8000
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

Frontend (`frontend/Dockerfile`)
Three-stage build:
1. **deps** (`node:22-alpine`) — `pnpm install`
2. **builder** — `pnpm build`
3. **production** — copy `.next/standalone` + `.next/static` + `public`, expose 3000, run `node server.js`

Utility Scripts

`scripts/embed_schema.py`
Nightly schema embedding sync for pgvector semantic search.

```bash
uv run python scripts/embed_schema.py --schema public            # Full sync
uv run python scripts/embed_schema.py --domain revenue --dry-run # Preview only
```

- Queries `information_schema` + `pg_catalog.pg_description`

- Generates 1536-dim embeddings via `claude-embeddings-20250219`
- Upserts into `schema_embeddings` table
- Returns `{tables_processed, embeddings_generated, errors}`

`scripts/seed_test_data.py`
Synthetic test data generator for development.

```bash
uv run python scripts/seed_test_data.py                          # Default (1 tenant)
uv run python scripts/seed_test_data.py --tenants 3 --clear      # 3 tenants, fresh start
```

Generates realistic data for 10 tables:
`sales`, `customers`, `orders`, `transactions`, `users`, `products`, `regions`, `sales_representatives`, `deals`, `activity`

Each table has a dedicated generator function (realistic company names, product names, dates). Batch inserts in chunks of 100 with `ON CONFLICT DO NOT HING`.

Key Commands

```bash
# Backend
cd backend
uv run uvicorn app.main:app --reload --port 8000   # Dev server
uv run pytest tests/ -v --tb=short                 # Fast tests (skip e2e)
uv run pytest tests/ -v -m "not e2e"               # Skip E2E
uv run alembic upgrade head                        # Apply migrations
uv run alembic revision --autogenerate -m "msg"    # New migration
uv run ruff check . && uv run ruff format --check . # Lint + format

# Frontend
cd frontend
pnpm dev                    # Next.js dev (port 3000)
pnpm build && pnpm start    # Production
pnpm lint                   # ESLint
pnpm typecheck              # tsc --noEmit

# Semantic layer
cd semantic/dbt
dbt run --select +metrics.*   # Run dbt models
dbt test                      # Run dbt tests
```

docs/semantic-layer.md

134 lines · 5.9 KB

```
# Semantic Layer

> Single source of truth for business metric definitions:
> **Cube-native data models** (`semantic/cube/model/`) served by **Cube.dev**,
> consumed by the Python **CubeClient**. See
> [ADR 007](adr/007-cube-native-semantic-layer.md) for why the dbt tier was
> dropped.

## Architecture

```
semantic/cube/model/*.yml   (10 cubes, 23 measures — the catalog)
    │
    v
Cube.dev (cube.js, CUBEJS_DB_* env, cube_reader role)
    │  REST API: /meta (definitions), /load (data — Phase 10)
    v
CubeClient (backend/app/semantic/cube_client.py, JWT-authenticated)
    │  /meta → MetricDefinition map → LLM context formatting
    v
ChatService → NL2SQLAgent prompt ("## Available Metrics") + /api/v1/metrics/list
```

The Catalog

Location: `semantic/cube/model/` — one YAML per cube, over the ten
seeded analytics tables. Highlights:

Cube	Measures	Dimensions (non-time)
Sales	`revenue_total` (USD), `units_sold`, `avg_revenue_per_sale`, `sale_count`	region, product_name
Orders	`order_count`, `order_value` (USD), `avg_order_value`	status
Customers	`customer_count`, `active_customers` (status filter)	country, status
Transactions	`transaction_volume` (USD), `transaction_count`, `avg_transaction`	type, status
WebUsers	`user_count`, `active_users`	country, status
Activity	`event_count`, `unique_active_users` (countDistinct)	event_type
Deals	`pipeline_value` (USD), `deal_count`, `won_deals`, `win_rate` (calculated, %)	stage
SalesReps	`rep_count`	name
Products	`product_count`, `avg_price`	product_name, category
Regions	`region_count`	region_name

Rules enforced by `backend/tests/semantic/test_catalog.py`:

- unique cube names and measure names across the catalog
- joins reference existing cubes and are **symmetric** (both sides declared)
- every cube carries a `tenant_id` dimension (hidden; Phase-10 queryRewrite hook)
- every measure has a `type` or a `sql` expression
- the core metrics (revenue, orders, active users, pipeline, win rate...) exist

Joins follow real seed foreign keys only: Orders↔Customers,
Deals↔SalesReps↔Regions, Activity↔WebUsers. `sales.product_id`/`rep_id` are
dangling in the seed data, so Sales declares no joins.

Cube Runtime

- **Config:** `semantic/cube/cube.js` — API secret + dev mode from env.
- **Database:** standard `CUBEJS_DB_HOST/PORT/NAME/USER/PASS` env; Cube
  connects as the read-only `cube_reader` role (created by
  `infra/postgres/init.sql`).
- **Dev compose:** the `cube` service mounts `semantic/cube` at `/cube/conf`
  and healthchecks via `node -e fetch(...)`.
- **Secrets:** `scripts/gen-env.sh` writes `semantic/cube/.env`, keeping
  `CUBEJS_API_SECRET` aligned with the backend's `CUBE_API_SECRET`.

RLS interaction (important)

Analytics tables are FORCE-RLS tenant-scoped (Phase 8b / ADR 006), and data
queries are tenant-correct since Phase 10 (ADR 008): the backend mints JWTs
carrying a `tenantId` claim; Cube keys orchestrators (driver pools) by
tenant; each pool's connections set `app.current_tenant_id` via the pg
`options` parameter — the GUC the RLS policies consume. **Postgres remains
the enforcement layer**: if any link is misconfigured, queries return zero
rows (fail-closed), never another tenant's data. `POST /api/v1/metrics/query`
and the Explore page use this path; metadata (`/meta`) needs no tenant.

CubeClient

Location: `backend/app/semantic/cube_client.py`

- **Auth:** HS256 JWT signed with `CUBE_API_SECRET` (cached ~1h). Data
  queries add a `tenantId` claim — cached per tenant — which Cube surfaces
  as `securityContext` (ADR 008).
- **`get_meta()`**: fetches/parse `/meta` → `CubeMetaResponse{cubes,
  metrics, raw}`. The **raw payload** is what gets cached (Redis + in-memory,
  TTL 5 min) and re-parsed on hit — serializing the parsed form loses the
  metrics map.
- **`list_metrics()` / `get_metric(name)`**: MetricDefinition lookup, indexed
  both as `cube.measure` and bare `measure`.
- **`query(..., tenant_id=...)`**: POST `/load` with measures/dimensions/timeDimensions/
  filters — tenant-scoped through the per-tenant driver GUC; results cached
  per tenant in the two-tier cache (300s).
- **`get_agent_context(query=...)`**: the primary agent integration point.
  With ≤20 metrics, formats the full catalog; above that, keyword-ranks
  metrics against the query (name 3×, title/cube 2×, description 1×) and
  injects the top 20; no keyword overlap falls back to the full list.
- **`health_check()`**: returns bool, never raises.
- Failure semantics: callers fail open — ChatService continues without
  metric context when Cube is down.

MetricDefinition
```

```python
name            # "Sales.revenue_total"
title           # "Total Revenue"
description     # human + LLM readable
metric_type     # MetricType (sum/count/count_distinct/avg/min/max/...)
cube_name       # "Sales"
measure_name    # "revenue_total"
dimensions      # non-time dimension names
time_dimensions # time dimension names
format          # {"currency": "USD"} | {"percent": ...} | None
```

API Surface

- `GET /api/v1/metrics/list` — the catalog as JSON (JWT-authenticated;
  `503 {"detail": {"code": "CUBE_UNAVAILABLE"}}` when Cube is down).
- `POST /api/v1/metrics/query` — tenant-scoped metric query (ADR 008);
  measures validated against the catalog (`400 INVALID_METRIC`); results
  cached 300s per tenant.
- `GET /api/v1/datasources` — the cubes as datasource summaries.
- Frontend: the `/explore` page (catalog → query builder → table + chart).

Key Principle

Metrics are defined ONCE in `semantic/cube/model/`. Never re-implement
metric logic in agent prompts or service code — always consume via
CubeClient.

Adding a Metric

1. Edit the cube YAML (or add a cube file).
2. `uv run pytest tests/semantic/test_catalog.py` — invariants must hold.
3. `make restart` — `/meta` reflects the change (5-min client cache).

frontend/.dockerignore

6 lines · 0.1 KB

```
node_modules
.next
out
npm-debug.log*
pnpm-debug.log*
.git
```

frontend/Dockerfile

34 lines · 1.2 KB

```
# Stage 1: Dependencies
# pnpm-lock.yaml is copied if present (glob). --frozen-lockfile is used only
# when the lockfile exists; otherwise pnpm resolves and writes one. Commit the
# generated lockfile for reproducible builds (scripts/setup.sh does this).
# pnpm is PINNED to 9 (same as CI) — pnpm 10 rejects sharp's postinstall
# build script by default (ERR_PNPM_IGNORED_BUILDS) and would need
# pnpm.onlyBuiltDependencies config; pinning matches the lockfile format.
FROM node:22-alpine AS deps
WORKDIR /app
COPY package.json pnpm-lock.yaml* ./
RUN corepack enable && corepack prepare pnpm@9.15.9 --activate \
 && if [ -f pnpm-lock.yaml ]; then \
        pnpm install --frozen-lockfile; \
    else \
        pnpm install; \
    fi

# Stage 2: Build
FROM node:22-alpine AS builder
WORKDIR /app
COPY --from=deps /app/node_modules ./node_modules
COPY . .
RUN corepack enable && corepack prepare pnpm@9.15.9 --activate && pnpm build

# Stage 3: Production
FROM node:22-alpine AS production
WORKDIR /app
ENV NODE_ENV=production
COPY --from=builder /app/.next/standalone ./
COPY --from=builder /app/.next/static ./.next/static
COPY --from=builder /app/public ./public

EXPOSE 3000
CMD ["node", "server.js"]
```

frontend/eslint.config.mjs

24 lines · 0.6 KB

```
import { FlatCompat } from "@eslint/eslintrc";
import path from "node:path";
import { fileURLToPath } from "node:url";

const compat = new FlatCompat({
  baseDirectory: path.dirname(fileURLToPath(import.meta.url)),
});

const eslintConfig = [
  ...compat.extends("next/core-web-vitals", "next/typescript"),
  {
    ignores: [
      ".next/**",
      "out/**",
      "build/**",
      "node_modules/**",
      // Generated by Next.js on dev/build; its triple-slash references are
      // part of the generated contract, not project code.
      "next-env.d.ts",
    ],
  },
];

export default eslintConfig;
```

frontend/next-env.d.ts

6 lines · 0.3 KB

```
/// <reference types="next" />
/// <reference types="next/image-types/global" />
/// <reference path="./.next/types/routes.d.ts" />

// NOTE: This file should not be edited
// see https://nextjs.org/docs/app/api-reference/config/typescript for more information.
```

frontend/next.config.js

9 lines · 0.2 KB

```
/** @type {import('next').NextConfig} */
const nextConfig = {
  output: "standalone",
  images: {
    remotePatterns: [],
  },
};

module.exports = nextConfig;
```

frontend/package.json

46 lines · 1.2 KB

```
{
  "name": "genbi-frontend",
  "version": "0.1.0",
  "private": true,
  "description": "GenBI frontend — Next.js 15 with shadcn/ui and Flint chart rendering",
  "scripts": {
    "dev": "next dev --port 3000",
    "build": "next build",
    "start": "next start",
    "lint": "eslint .",
    "typecheck": "tsc --noEmit",
    "format": "prettier --write ."
  },
  "dependencies": {
    "next": "^15.1",
    "react": "^19.0",
    "react-dom": "^19.0",
    "tailwindcss": "^4.0",
    "@tailwindcss/postcss": "^4.0",
    "typescript": "^5.9",
    "zod": "^3.24",
    "lucide-react": "^0.468",
    "react-markdown": "^9.0",
    "rehype-raw": "^7.0",
    "rehype-sanitize": "^6.0",
    "remark-gfm": "^4.0",
    "class-variance-authority": "^0.7",
    "clsx": "^2.1",
    "tailwind-merge": "^2.6",
    "@radix-ui/react-dialog": "^1.1",
    "@radix-ui/react-dropdown-menu": "^2.1",
    "@radix-ui/react-slot": "^1.1",
    "@radix-ui/react-tooltip": "^1.1",
    "@radix-ui/react-scroll-area": "^1.2",
    "@radix-ui/react-tabs": "^1.1",
    "@radix-ui/react-avatar": "^1.1"
  },
  "devDependencies": {
    "@types/node": "^22.10",
    "@types/react": "^19.0",
    "@types/react-dom": "^19.0",
    "prettier": "^3.4",
    "eslint": "^9.17",
    "eslint-config-next": "^15.1"
  }
}
```

frontend/pnpm-lock.yaml

5707 lines · 195.3 KB

```yaml
lockfileVersion: '9.0'

settings:
  autoInstallPeers: true
  excludeLinksFromLockfile: false

importers:

  .:
    dependencies:
      '@radix-ui/react-avatar':
        specifier: ^1.1
        version: 1.2.6(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-dialog':
        specifier: ^1.1
        version: 1.1.23(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-dropdown-menu':
        specifier: ^2.1
        version: 2.1.24(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-scroll-area':
        specifier: ^1.2
        version: 1.2.18(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-slot':
        specifier: ^1.1
        version: 1.3.3(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-tabs':
        specifier: ^1.1
        version: 1.1.21(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-tooltip':
        specifier: ^1.1
        version: 1.2.16(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@tailwindcss/postcss':
        specifier: ^4.0
        version: 4.3.3
      class-variance-authority:
        specifier: ^0.7
        version: 0.7.1
      clsx:
        specifier: ^2.1
        version: 2.1.1
      lucide-react:
        specifier: ^0.468
        version: 0.468.0(react@19.2.8)
      next:
        specifier: ^15.1
        version: 15.5.23(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      react:
        specifier: ^19.0
        version: 19.2.8
      react-dom:
        specifier: ^19.0
        version: 19.2.8(react@19.2.8)
      react-markdown:
        specifier: ^9.0
        version: 9.1.0(@types/react@19.2.18)(react@19.2.8)
      rehype-raw:
        specifier: ^7.0
        version: 7.0.0
      rehype-sanitize:
        specifier: ^6.0
        version: 6.0.0
      remark-gfm:
        specifier: ^4.0
        version: 4.0.1
      tailwind-merge:
        specifier: ^2.6
        version: 2.6.1
      tailwindcss:
        specifier: ^4.0
        version: 4.3.3
      typescript:
        specifier: ^5.9
        version: 5.9.3
      zod:
        specifier: ^3.24
        version: 3.25.76
    devDependencies:
      '@types/node':
        specifier: ^22.10
        version: 22.20.1
      '@types/react':
        specifier: ^19.0
        version: 19.2.18
      '@types/react-dom':
        specifier: ^19.0
        version: 19.2.4(@types/react@19.2.18)
      eslint:
        specifier: ^9.17
        version: 9.39.5(jiti@2.7.0)
      eslint-config-next:
        specifier: ^15.1
        version: 15.5.23(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3)
      prettier:
        specifier: ^3.4
        version: 3.9.6

packages:

  '@alloc/quick-lru@5.2.0':
    resolution: {integrity: sha512-UrcABB+4bUrFABwbluTIBErXwvbsU/V7TZWfmbgJfbkwiBuziS9gxdODUyuiecfdGQ85jglMW6juS3+z5TsKLw==}
```

```
    engines: {node: '>=10'}

  '@emnapi/core@1.10.0':
    resolution: {integrity: sha512-yq6OkJ4p82CAfPl0u9mQebQHKPJkY7WrIuk205cTYnYe+k2Z8YBh11FrbRG/H6ihirqcacOgl2BIO8oyMQLeXw==}

  '@emnapi/runtime@1.10.0':
    resolution: {integrity: sha512-ewvYlk86xUoGI0zQRNq/mC+16R1QeDlKQy21Ki3oSYXNgLb45GV1P6A0M+/s6nyCuNDqe5VpaY84BzXGwVbwFA==}

  '@emnapi/runtime@1.11.3':
    resolution: {integrity: sha512-Xz4Tpyki7XyrpbUK1jR1AhdAdaXyhhY4lZ3neLodmhpuWfy2PAQN5B46sAiU4liOXGLkHypn/qU+jvfWSCYYLA==}

  '@emnapi/wasi-threads@1.2.1':
    resolution: {integrity: sha512-uTII7OYF+/Mes/MrcIOYp5yOtSMLBWSIoLPpcgwipoiKbli6k322tcoFsxoIIxPDqW01SQGAgko4EzZi2BNv2w==}

  '@eslint-community/eslint-utils@4.10.1':
    resolution: {integrity: sha512-cuadcxVFE8sDK6iWJbs8Sn0av2Nrh2QSGQhVlBW9AaAHqHwjWsZHT8LJ4hFGPh7ASBV2deFdM7H/DPjulmh8rg==}
    engines: {node: ^12.22.0 || ^14.17.0 || >=16.0.0}
    peerDependencies:
      eslint: ^6.0.0 || ^7.0.0 || >=8.0.0

  '@eslint-community/regexpp@4.12.2':
    resolution: {integrity: sha512-EriSTlt5OC9/7SXkRSCAhfSxxoSUgBm33OH+IkwbdpgoqsSsUg7y3uh+IICI/Qg4BBWr3U2i39RpmycbxMq4ew==}
    engines: {node: ^12.0.0 || ^14.0.0 || >=16.0.0}

  '@eslint/config-array@0.21.2':
    resolution: {integrity: sha512-nJl2KGTlrf9GjLimgIru+V/mzgSK0ABCDQRvxw5BjURL7WfH5uoWmizbH7QB6MmnMBd8cIC9uceWnezL1VZWWw==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}

  '@eslint/config-helpers@0.4.2':
    resolution: {integrity: sha512-gBrxN88gOIf3R7ja5K9slwNayVcZgK6SOUORm2uBzTeIEfeVaIhOpCtTox3P6R7o2jLFwLFTLnC7kU/RGcYEgw==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}

  '@eslint/core@0.17.0':
    resolution: {integrity: sha512-yL/sLrpmtDaFEiUj1osRP4TI2MDz1AddJL+jZ7KSqvBuliN4xqYY54IfdN8qD8Toa6g1iloph1fxQNkjOxrrpQ==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}

  '@eslint/eslintrc@3.3.6':
    resolution: {integrity: sha512-l2Ul9PrHsPCKcEY/ac7VgFj9D80C7S68sOKc618SyHDPK36s1XcFebXY0iTzUVn4Yq+YbwvSnDmCz9yxjX+QrA==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}

  '@eslint/js@9.39.5':
    resolution: {integrity: sha512-QywQuszQh77pIXCsq998c8hbhSTI/azTty1Z6N53dmAudKHhy573j3yvRLsX2BSp8YpLtoCEG8E9DJe+8zUh4A==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}

  '@eslint/object-schema@2.1.7':
    resolution: {integrity: sha512-VtAOaymWVfZcmZbp6E2mympDIHvyjXs/12LqWYjVw6qjrfF+VK+fyG33kChz3nnK+SU5/NeHOqrTEHS8sXO3OA==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}

  '@eslint/plugin-kit@0.4.1':
    resolution: {integrity: sha512-43/qtrDUokr7LJqoF2c3+RInu/t4zfrpYdoSDfYyhg52rwLV6TnOvdG4fXm7IkSB3wErkcmJS9iEhjVtOSEjjA==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}

  '@floating-ui/core@1.8.0':
    resolution: {integrity: sha512-0CIZ5itps/8x7BG8dEIhs53BvCUH2PCoogtakwRTut+Arm58sJooJ0AuZhLw2HJYIR5cMLNPBSS728sPho2khQ==}

  '@floating-ui/dom@1.8.0':
    resolution: {integrity: sha512-yXSrzeHZBTZadLOlfyhCkJHNeLJnHRnRInwdZ40L7ZiaAtrBwoYlsDrX3v5zB1Utk7CLfzcOVnVVWoXEky7Ceg==}

  '@floating-ui/react-dom@2.1.9':
    resolution: {integrity: sha512-JDjEFGCpImxDCA7JJKviA0M9+RtmJdj0m/NVU5IMgBK+AmZouAQQ7/+2GLH0GXXY0YMw9oXPB8hKdbPYg5QLYg==}
    peerDependencies:
      react: '>=16.8.0'
      react-dom: '>=16.8.0'

  '@floating-ui/utils@0.2.12':
    resolution: {integrity: sha512-HpCo8tmWzLVad5s2d19EhAz5zqrrQ6s69qd6moPMQvkOuSwDT1YgRfWSVuc4ennqrgv3OHppiOGMQ7oC13yIww==}

  '@humanfs/core@0.19.2':
    resolution: {integrity: sha512-UhXNm+CFMWcbChXywFwkmhqjs3PRCmcSa/hfBgLIb7oQ5HNb1wS0icWsGtSAUNgefHeI+eBrA8I1fxmbHsGdvA==}
    engines: {node: '>=18.18.0'}

  '@humanfs/node@0.16.8':
    resolution: {integrity: sha512-gE1eQNZ3R++kTzFUpdGlpmy8kDZD/MLyHqDwqjkVQI0JMdI1D51sy1H958PNXYkM2rAac7e5/CnIKZrHtPh3BQ==}
    engines: {node: '>=18.18.0'}

  '@humanfs/types@0.15.0':
    resolution: {integrity: sha512-ZZ1w0aoQkwuUuC7Yf+7sdeaNfqQiiLcSRbfI08oAxqLtpXQr9AIVX7Ay7HLDuiLYAaFPu8oBYNq/QIi9URHJ3Q==}
    engines: {node: '>=18.18.0'}

  '@humanwhocodes/module-importer@1.0.1':
    resolution: {integrity: sha512-bxveV4V8v5Yb4ncFTT3rPSgZBOpCkjfK0y4oVVVJwIuDVBRMDXrPyXRL988i5ap9m9bnyEEjWfm5WkBmtffLfA==}
    engines: {node: '>=12.22'}

  '@humanwhocodes/retry@0.4.3':
    resolution: {integrity: sha512-bV0Tgo9K4hfPCek+aMAn81RppFKv2ySDQeMoSZuvTASywNTnVJCArCZE2FWqpvIatKu7VMRLWlR1EazvVhDyhQ==}
    engines: {node: '>=18.18'}

  '@img/colour@1.1.0':
    resolution: {integrity: sha512-Td76q7j57o/tLVdgS746cYARfSyxk8iEfRxewL9h4OMzYhbW4TAcppl0mT4eyqXddh6L/jwoM75mo7ixa/pCeQ==}
    engines: {node: '>=18'}

  '@img/sharp-darwin-arm64@0.34.5':
    resolution: {integrity: sha512-imtQ3WMJXbMY4fxb/Ndp6HBTNVtWCUI0WdobyheGf5+ad6xX8VIDO8u2xE4qc/fr08CKG/7dDseFtn6M6g/r3w==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [arm64]
    os: [darwin]

  '@img/sharp-darwin-x64@0.34.5':
    resolution: {integrity: sha512-YNEFAF/4KQ/PeW0N+r+aVVsoIY0/qxxikF2SWdp+NRkmMB7y9LBZAVqQ4yhGCm/H3H270OSykqmQMKLBhBJDEw==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [x64]
    os: [darwin]

  '@img/sharp-libvips-darwin-arm64@1.2.4':
    resolution: {integrity: sha512-zqjjo7RatFfFoP0MkQ51jfuFZBnVE2pRiaydKJ1G/rHZvnsrHAOcQALIi9sA5co5xenQdTugCvtb1cuf78Vf4g==}
```

```
    cpu: [arm64]
    os: [darwin]

  '@img/sharp-libvips-darwin-x64@1.2.4':
    resolution: {integrity: sha512-1IOd5xfVhlGwX+zXv2N93k0yMONvUlANylbJw1eTah8K/Jtpi15KC+WSiaX/nBmbm2HxRM1gZ0nSdjSsrZbGKg==}
    cpu: [x64]
    os: [darwin]

  '@img/sharp-libvips-linux-arm64@1.2.4':
    resolution: {integrity: sha512-excjX8DfsIcJ10x1Kzr4RcWe1edC9PquDRRPx3YVCvQv+U5p7Yin2s32ftzikXojb1PIFc/9Mt28/y+iRklkrw==}
    cpu: [arm64]
    os: [linux]

  '@img/sharp-libvips-linux-arm@1.2.4':
    resolution: {integrity: sha512-bFI7xcKFELdiNCVov8e44Ia4u2byA+l3XtsAj+Q8tfCwO6BQ8iDojYdvoPMqsKDkuoOo+X6HZA0s0q11ANMQ8A==}
    cpu: [arm]
    os: [linux]

  '@img/sharp-libvips-linux-ppc64@1.2.4':
    resolution: {integrity: sha512-FMuvGijLDYG6lW+b/UvyilUWu5Ayu+3r2d1S8notiGCIyYU/76eig1UfMmkZ7vwgOrzKzlQbFSuQfgm7GYUPpA==}
    cpu: [ppc64]
    os: [linux]

  '@img/sharp-libvips-linux-riscv64@1.2.4':
    resolution: {integrity: sha512-oVDbcR4zUC0ce82teubSm+x6ETixtKZBh/qbREIOcI3cULzDyb18Sr/Wcyx7NRQeQzOiHTNbZFF1UwPS2scyGA==}
    cpu: [riscv64]
    os: [linux]

  '@img/sharp-libvips-linux-s390x@1.2.4':
    resolution: {integrity: sha512-qmp9VrzgPgMoGZyPvrQHqk02uyjA0/QrTO26Tqk6l4ZV0MPWIW6LTkqOIov+J1yEu7MbFQaDpwdwJKhbJvuRxQ==}
    cpu: [s390x]
    os: [linux]

  '@img/sharp-libvips-linux-x64@1.2.4':
    resolution: {integrity: sha512-tJxiiLsmHc9Ax1bz3oaOYBURTXGIRDODBqhveVHonrHJ9/+k89qbLl0bcJns+e4t4rvaNBxaEZsFtSfAdquPrw==}
    cpu: [x64]
    os: [linux]

  '@img/sharp-libvips-linuxmusl-arm64@1.2.4':
    resolution: {integrity: sha512-FVQHuwx1IIuNow9QAbYUzJ+En8KcVm9Lk5+uGUQJHaZmMECZmOlix9HnH7n1TRkXMS0pGxIJokIVB9SuqZGGXw==}
    cpu: [arm64]
    os: [linux]

  '@img/sharp-libvips-linuxmusl-x64@1.2.4':
    resolution: {integrity: sha512-+LpyBk7L44ZIXwz/VYfglaX/okxezESc6UxDSoyo2Ks6Jxc4Y7sGjpgU9s4PMgqgjj1gZCylTieNamqA1MF7Dg==}
    cpu: [x64]
    os: [linux]

  '@img/sharp-linux-arm64@0.34.5':
    resolution: {integrity: sha512-bKQzaJRY/bkPOXyKx5EVup7qkaojECG6NLYswgktOZjaXecSAeCWiZwwiFf3/Y+O1HrauiE3FVsGxFg8c24rZg==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [arm64]
    os: [linux]

  '@img/sharp-linux-arm@0.34.5':
    resolution: {integrity: sha512-9dLqsvwtg1uuXBGZKsxem9595+ujv0sJ6Vi8wcTANSFpwV/GONat5eCkzQo/1O6zRIkh0m/8+5BjrRr7jDUSZw==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [arm]
    os: [linux]

  '@img/sharp-linux-ppc64@0.34.5':
    resolution: {integrity: sha512-7zznwNaqW6YtsfrGGDA6BRkISKAAE1Jo0QdpNYXNMHu2+0dTrPflTLNkpc8l7MUP5M16ZJcUvysVWWrMefZquA==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [ppc64]
    os: [linux]

  '@img/sharp-linux-riscv64@0.34.5':
    resolution: {integrity: sha512-51gJuLPTKa7piYPaVs8GmByo7/U7/7TZOq+cnXJIHZKavIRHAP77e3N2HEl3dgiqdD/w0yUfiJnII77PuDDFdw==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [riscv64]
    os: [linux]

  '@img/sharp-linux-s390x@0.34.5':
    resolution: {integrity: sha512-nQtCk0PdKfho3eC5MrbQoigJ2gd1CgddUMkabUj+rBevs8tZ2cULOx46E7oyX+04WGfABgIwmMC0VqieTiR4jg==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [s390x]
    os: [linux]

  '@img/sharp-linux-x64@0.34.5':
    resolution: {integrity: sha512-MEzd8HPKxVxVenwAa+JRPwEC7QFjoPWuS5NZnBt6B3pu7EG2Ge0id1oLHZpPJdn3OQK+BQDiw9zStiHBTJQQQQ==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [x64]
    os: [linux]

  '@img/sharp-linuxmusl-arm64@0.34.5':
    resolution: {integrity: sha512-fprJR6GtRsMt6Kyfq44IsChVZeGN97gTD331weR1ex1c1rypDEABN6Tm2xa1wE6lYb5DdEnk03NZPqA7Id21yg==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [arm64]
    os: [linux]

  '@img/sharp-linuxmusl-x64@0.34.5':
    resolution: {integrity: sha512-Jg8wNT1MUzIvhBFxViqrEhWDGzqymo3sV7z7ZsaWbZNDLXRJZoRGrjulp60YYtV4wfY8VIKcWidjojlLcWrd8Q==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [x64]
    os: [linux]

  '@img/sharp-wasm32@0.34.5':
    resolution: {integrity: sha512-OdWTEiVkY2PHwqkbBI8frFxQQFekHaSSkUIJkwzclWZe64O1X4UlUjqqqLaPbUpMOQk6FBu/HtlGXNblIs0huw==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [wasm32]

  '@img/sharp-win32-arm64@0.34.5':
    resolution: {integrity: sha512-WQ3AgWCWYSb2yt+IG8mnC6Jdk9Whs7O0gxphblsLvdhSpSTtmu69ZG1Gkb6NuvxsNACwiPV6cNSZNzt0KPsw7g==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [arm64]
```

```
    os: [win32]

  '@img/sharp-win32-ia32@0.34.5':
    resolution: {integrity: sha512-FV9m/7NmeCmSHDD5j4+4pNI8Cp3aW+JvLoXcTUo0IqyjSfAZJ8dIUmijx1qaJsIiU+Hosw6xM5KijAWRJCSgNg==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [ia32]
    os: [win32]

  '@img/sharp-win32-x64@0.34.5':
    resolution: {integrity: sha512-+29YMsqY2/9eFEiW93eqWnuLcWcufowXewwSNIT6UwZdUUCrM3oFjMWH/Z6/TMmb4hlFenmfAVbpWeup2jryCw==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}
    cpu: [x64]
    os: [win32]

  '@jridgewell/gen-mapping@0.3.13':
    resolution: {integrity: sha512-2kkt/7niJ6MgEPxF0bYdQ6etZaA+fQvDcLKckhy1yIQOzaoKjBBjSj63/aLVjYE3qhRt5dvM+uUyfCg6UKCBbA==}

  '@jridgewell/remapping@2.3.5':
    resolution: {integrity: sha512-LI9u/+laYG4Ds1TDKSJW2YPrIlcVYOwi2fUC6xB43lueCjgxV4lffOCZCtYFiH6TNOX+tQKXx97T4IKHbhyHEQ==}

  '@jridgewell/resolve-uri@3.1.2':
    resolution: {integrity: sha512-bRISgCIjP20/tbWSPWMEi54QVPRZExkuD9lJL+UIxUKtwVJA8wW1Trb1jMs1RFXo1CBTNZ/5hpC9QvmKWdopKw==}
    engines: {node: '>=6.0.0'}

  '@jridgewell/sourcemap-codec@1.5.5':
    resolution: {integrity: sha512-cYQ9310grqxueWbl+WuIUIaiUaDcj7WOq5fVhEljNVgRfOUhY9fy2zTvfoqWsnebh8Sl70VScFbICvJnLKB0Og==}

  '@jridgewell/trace-mapping@0.3.31':
    resolution: {integrity: sha512-zzNR+SdQSDJzc8joaeP8QQoCQr8NuYx2dIIytl1QeBEZHJ9uW6hebsrYgbz8hJwUQao3TWCMtmfV8Nu1twOLAw==}

  '@napi-rs/wasm-runtime@1.2.3':
    resolution: {integrity: sha512-UMduMbqO5s5zF2NkNacMT/yK5Y5QiKvWr2+50bzIIxFDwVJ2h49b+oyjaCGPhJxd2/gC2x39EHv/gHVuu36x2Q==}
    engines: {node: ^20.19.0 || ^22.13.0 || >=23.5.0}
    peerDependencies:
      '@emnapi/core': ^1.7.1 || ^2.0.0-alpha.4
      '@emnapi/runtime': ^1.7.1 || ^2.0.0-alpha.4

  '@next/env@15.5.23':
    resolution: {integrity: sha512-Mv3Z9hVbFcPnoLevsZ6rnX1TBtyHb5E17yN7HTPDXSXxeNsGBjUFrdbjRXKKXIOhfth7/cg6Ay7PZ2UFawaWsQ==}

  '@next/eslint-plugin-next@15.5.23':
    resolution: {integrity: sha512-0KnCFpiWVIsbwBhByZ0uIcjYM5xqGrFzN2eOPbwru/wuy5Z1dmA+3gP+PRbi4gl1Ny7an66BqAM/NHkX/50rbw==}

  '@next/swc-darwin-arm64@15.5.23':
    resolution: {integrity: sha512-SrEwOROH/rhA03F59hHtdhgtfZMWGzr5duDBWgRQt2rS3mJhqMKOcnNx6txOd0/i3E3D3uFKYFvyHsEiwQxzag==}
    engines: {node: '>= 10'}
    cpu: [arm64]
    os: [darwin]

  '@next/swc-darwin-x64@15.5.23':
    resolution: {integrity: sha512-f0FpFbG2EhDCuptBGcfrLcYMDuQAhe6m1QA4VVfXFrIBoFXvXt/olGbBkYkloKlXQtmhuzvtdYyuu/6zf07GIg==}
    engines: {node: '>= 10'}
    cpu: [x64]
    os: [darwin]

  '@next/swc-linux-arm64-gnu@15.5.23':
    resolution: {integrity: sha512-WlNtfepUXKX2u2ZsJZ8c3c8+tJSRZqsYzoMwLOY72A8ucKCCgxgNhiePA3qzFYahVWrwcQd8jOeJmBinc+VFVQ==}
    engines: {node: '>= 10'}
    cpu: [arm64]
    os: [linux]

  '@next/swc-linux-arm64-musl@15.5.23':
    resolution: {integrity: sha512-W/6qKk7UG93mg14PmQC+2urt69MIdwTBLNQ6MJyeC4wOCIHCjz+VfgssvS1pK7mgYBtLC1g6VKNoHD9xB0WWGg==}
    engines: {node: '>= 10'}
    cpu: [arm64]
    os: [linux]

  '@next/swc-linux-x64-gnu@15.5.23':
    resolution: {integrity: sha512-vzefI32mi6VMk96RaTAyxApgfGbiFzQBXVsekEjsDv1fr48mlABTWx0sUYhaYCBHWqCalxmz3DxbxFcbFvzNtw==}
    engines: {node: '>= 10'}
    cpu: [x64]
    os: [linux]

  '@next/swc-linux-x64-musl@15.5.23':
    resolution: {integrity: sha512-qppK/3dTGOTI+aoWWBZc3DshFIhrzgL8guATlaN9V6M1QJxbkP/rhEZ22tdICsQ/2WWXopMZ2Jokzj2u3uKY3Q==}
    engines: {node: '>= 10'}
    cpu: [x64]
    os: [linux]

  '@next/swc-win32-arm64-msvc@15.5.23':
    resolution: {integrity: sha512-Wc29KFOdT7XBcII3Vtmw7aoU8Uk3Mes/FNJfhFeSHdYBFJWMcR/DsI8U9BCPUhq/uycsUVuqSKGthW15tLsigA==}
    engines: {node: '>= 10'}
    cpu: [arm64]
    os: [win32]

  '@next/swc-win32-x64-msvc@15.5.23':
    resolution: {integrity: sha512-/C7wRW4fa9s/PKA18zGPPpVmx8ycgVpP8yOxro4gzGTzjPJdscbAP3ODeFvgiIovxD176Z2J/SXO9t8PJKHLeQ==}
    engines: {node: '>= 10'}
    cpu: [x64]
    os: [win32]

  '@nodelib/fs.scandir@2.1.5':
    resolution: {integrity: sha512-vq24Bq3ym5HEQm2NKCr3yXDwjc7vTsEThRDnkp2DK9p1uqLR+DHurm/NOTo0KG7HYHU7eppKZj3MyqYuMBf62g==}
    engines: {node: '>= 8'}

  '@nodelib/fs.stat@2.0.5':
    resolution: {integrity: sha512-RkhPPp2zrqDAQA/2jNhnztcPAlv64XdhIp7a7454A5ovI7Bukxgt7MX7udwAu3zg1DcpPU0rz3VV1SeaqvY4+A==}
    engines: {node: '>= 8'}

  '@nodelib/fs.walk@1.2.8':
    resolution: {integrity: sha512-oGB+UxlgWcgQkgwo8GcEGwemoTFt3FIO9ababBmaGwXIoBKZ+GTy0pP185beGg7Llih/NSHSV2XAs1lnznocSg==}
    engines: {node: '>= 8'}

  '@nolyfill/is-core-module@1.0.39':
    resolution: {integrity: sha512-nn5ozdjYQpUCZlWGuxcJY/KpxkWQs4DcbMCmKojjyrYDEAGy4Ce19NN4v5MduafTwJlbKc99UA8YhSVqq9yPZA==}
```

```yaml
    engines: {node: '>=12.4.0'}

  '@radix-ui/number@1.1.3':
    resolution: {integrity: sha512-Road2bidD0uu/1BGDOWNdPI06g0lIRy6IF9GZcIrDK2KGItfor8IQwQa+yM2ERgHM1MmHxaxpTzk0/Jp42lNfA==}

  '@radix-ui/primitive@1.1.7':
    resolution: {integrity: sha512-rqWnm76nYT8HoNNqEjpgJ7Pw/DrBj5iBTrmEPo6HTX5+VJyBNOqTdv4g89G63HuR5g0AaENoAcH7Is5fF2kZ8Q==}

  '@radix-ui/react-arrow@1.1.15':
    resolution: {integrity: sha512-v4zggRcjadnI+ClKDuijlQEW4tw3NoaeHc/PwpKnLoLLKNUG4InLegkstooLcRIUWCs+8L22dGURCVuFfOKfnA==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-avatar@1.2.6':
    resolution: {integrity: sha512-4ULOTJ/mqy2hT9GlWa/MFHxHSvH3nJzHnZM1waNsc5Bonv7i70aNenghXmD97S6OJ81ekXONGGt4nT1r0PfEdA==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-collection@1.1.15':
    resolution: {integrity: sha512-9W+B9NPF0NaaPh/1NJd3+KqsnlLqU9H7T2rvww+fp+T/evVXdNAyYcnfRQZFOjkR1ajQp3yORlqnI8soawLvNA==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-compose-refs@1.1.5':
    resolution: {integrity: sha512-+48PbAAbq3didjJxa+OaWY2ZwgAKsNiRGyeHKszblZMQ+kcpd9pAaT11cMkGEie0vsOi3QdeTE6d5Fe3Gn61kA==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  '@radix-ui/react-context@1.2.2':
    resolution: {integrity: sha512-RHCUGwKHDr0hDGg4X7ma4JG4/+12qxw8rkh5QKdDldlCvtja6nUx1Ef/8HVrJze81lEsgLQlqjzjGNHantgnQA==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  '@radix-ui/react-dialog@1.1.23':
    resolution: {integrity: sha512-Ksw4WeROkO4rC9k/onilX/Ao2Cr1ku1unMNH+XSCcP4jSXYu7HDsg9n4ojMjVb22XpYjAQ9qfrFlVbru1vXDUA==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-direction@1.1.4':
    resolution: {integrity: sha512-5pzg4FGQNpExhnhT2zlrP1wZFaYCd1K0nYWoFAdcYoYK868IEigqMX3B3f8yIoRlAhAeDWciLI6ZdCKHF9P4Vg==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  '@radix-ui/react-dismissable-layer@1.1.19':
    resolution: {integrity: sha512-8g4pfOL9HoKKLWGiypT+dphVqjFfmcXO5GBnhsG6zI+lxAx/8feQpr+1LSN8Re3hiZ+XkLNS4O9ztK11/LzQ6w==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-dropdown-menu@2.1.24':
    resolution: {integrity: sha512-geq8l2rJkxvkXsT9RMgtUE3P8pITFpTsvYpbySi1IH4fZEABD/Gp85myayFgxk0ktljGMJnCbeFkyTusvSvv7g==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
```

```
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-focus-guards@1.1.6':
    resolution: {integrity: sha512-RNOJjfZMTyBM6xYmV3IVGXkPjIhcBAuv48POevAXwrGJhkWZ9p1rFoIS1JFooPuT193AZmRsCPhpoVJxx6OPoQ==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  '@radix-ui/react-focus-scope@1.1.16':
    resolution: {integrity: sha512-wmRZ2WWLvmt6KHy2rNPOdPUjwq5xOHY02+m+udwJTn0aNIox/rkskAvJTyTLGhPK6KgrUjlJUJpgmx/+wFiFIQ==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-id@1.1.4':
    resolution: {integrity: sha512-TMQp2llA+RYn7JcjnrMnz7wN4pcVttPZnRZo52PLQsoLVKzNlVwUeHmfePgTgRluXFvlD3GD5g5MOVVTJCO0qA==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  '@radix-ui/react-menu@2.1.24':
    resolution: {integrity: sha512-uW7RVuU6Lp/ZtfeY4b3kL32zccgEWvPv1+cf17ubYzHa9cL8AHokmk36cG/XEiH/smbQvumnieXX9j/e9RqJWA==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-popper@1.3.7':
    resolution: {integrity: sha512-UsJrrd7w4wuKKTdvd/DNERVlwSlUcyXzjhyDwBk+3aPOsCjOY6ZSbxuw8E6lZTjjfP8Cpd0J8VVkrYUWyGYXyg==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-portal@1.1.17':
    resolution: {integrity: sha512-vKQLcWypUnwZVvfV7UkGahH2g6ySe8M8R+zYBwPrv5byZ9QAW6cQVvNKo7GgmD+p8aYb6D9JBuvy8/WhOno2wQ==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-presence@1.1.10':
    resolution: {integrity: sha512-3wyzCQ6+ubRA+D4uv9m95JYLXxmOHp05qjrkjeA7uKHHtjpPggQzc6DAb0URl7j67oR0K2foO4ip27TiX037Bw==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-primitive@2.1.10':
    resolution: {integrity: sha512-MucOnzh6hR5mid6VpkbglRAMYMjKLqRnGBbjXkzjK52fuQDd1qbkx78a5P40mkcnVXJdEVxm26E9OPAiUq7nBg==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-roving-focus@1.1.19':
```

```yaml
    resolution: {integrity: sha512-V9jI6hDjT7l3jsCQD9bLNvDLM3tH/gdbOTp7Tefp3hbbgCGQoK7tUvrWiRlcoBHIZ809ElXwNQwVo0B98LuTXQ==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-scroll-area@1.2.18':
    resolution: {integrity: sha512-Zn5Cd171wxsO3Dfg8HaW6RifTb9CYTKQJHs/G4+LN1GfmJpaQMZQyQxMprVPHpaz7QY4l9BxK2JwQuzHsXC8nA==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-slot@1.3.3':
    resolution: {integrity: sha512-qx7oqnYbxnK9kYI9m317qmFmEgo6ywqWvbTogdj7cL9p3/yx4M48p7Rnw5z3H890cL/ow/EeWJsuTykeZVXP5Q==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  '@radix-ui/react-tabs@1.1.21':
    resolution: {integrity: sha512-UKxJlZid7FVtsk/WTxj4i4uSEgj2Au+KBbS7SQyTlzMhhn+86Cz3tISZdTa87bfEfcuvZezf2ZsxD4xuEKtkog==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-tooltip@1.2.16':
    resolution: {integrity: sha512-6EamKFRRnlpdadndbZ6LMwycfwkwPte1B42hs6QA0gYhjaOKqW4PZ4pjaW9UrlDX5eVt/OjncE7BFTPL5nmZhg==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/react-use-callback-ref@1.1.4':
    resolution: {integrity: sha512-R6OUY2e2fA6Yn6s+VSx5KBV6Nx8LQEhu+cz7LCej18rQ1HLyg9PSC9jP/ZNx0o6FAIK9c0F1kHylzSxKsdlkrQ==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  '@radix-ui/react-use-controllable-state@1.2.6':
    resolution: {integrity: sha512-uEQJGT97ZA/TgP/Hydw47lHu+/vQj6z/0jA+WeTbK1o9Rx45GImjpD0tc3W5ad3D6XTSR6e1yEO0FvGq6WQfVQ==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  '@radix-ui/react-use-effect-event@0.0.5':
    resolution: {integrity: sha512-7cshFL8HGS/7HEiHH+9kL9HBwp2sa9yX18Knwek6KYWmXwM7pegMgta2AXMQKI+rq3JnfSj9x8wYqFMTdG1Jgg==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  '@radix-ui/react-use-is-hydrated@0.1.3':
    resolution: {integrity: sha512-umO/aJ+82CpOnhDZUTbILCQf7kU/g0iv+oGs/Q8jw7IkhWBzaEP4sA268PhFAJTFetbwp3ICc6ktpI4TqtxcIw==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  '@radix-ui/react-use-layout-effect@1.1.4':
    resolution: {integrity: sha512-K20DkRkUwDnxEYMBPcg3Y6voLkEy5p5QQmszZgLngKKiC7dzBR/aEuK3w1qlx2JWDUNH6FluahYdgR3BP+QbYw==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
```

```yaml
  '@radix-ui/react-use-rect@1.1.4':
    resolution: {integrity: sha512-cSOCh6JlkmfjLyNcLiu2nB4v+nm+dkZ+Q5KHWk/soo4U7ZLiEQFKHK9/YmtBHjfCEaU43IBKQOc4/uJmCaiCTQ==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  '@radix-ui/react-use-size@1.1.4':
    resolution: {integrity: sha512-D3anSY15EJoxrihpsXI6SMrmmonnQtR2ni7arO+Lfdg3O95b9hNXxONk8jA5C8ANdF/h5HMAxejgs8PWJ6rlhw==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  '@radix-ui/react-visually-hidden@1.2.11':
    resolution: {integrity: sha512-NFS86RYYZb4/exihaESBGOpMJFz8MGLAfu3mOBSGByVnVPC9JPASfYubxd/8KbkQK0sYAv8lVQDEQukDX/qXvQ==}
    peerDependencies:
      '@types/react': '*'
      '@types/react-dom': '*'
      react: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
      react-dom: ^16.8 || ^17.0 || ^18.0 || ^19.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true
      '@types/react-dom':
        optional: true

  '@radix-ui/rect@1.1.3':
    resolution: {integrity: sha512-JtyZR+mqgBibTo8xea3B6ZRmzZiM/YeVBtUkas6zMuXjAlfIFIW2FgqeM9eLyvEaYX66vr6DJMK+4U6LV0KhNw==}

  '@rtsao/scc@1.1.0':
    resolution: {integrity: sha512-zt6OdqaDoOnJ1ZYsCYGt9YmWzDXl4vQdKTyJev62gFhRGKdx7mcT54V9KIjg+d2wi9EXsPvAPKe7i7WjfVWB8g==}

  '@rushstack/eslint-patch@1.16.1':
    resolution: {integrity: sha512-TvZbIpeKqGQQ7X0zSCvPH9riMSFQFSggnfBjFZ1mEoILW+UuXCKwOoPcgjMwiUtRqFZ8jWhPJc4um14vC6I4ag==}

  '@swc/helpers@0.5.15':
    resolution: {integrity: sha512-JQ5TuMi45Owi4/BIMAJBoSQoOJu12oOk/gADqlcUL9JEdHB8vyjUSsxqeNXnmXHjYKMi2WcYtezGEEhqUI/E2g==}

  '@tailwindcss/node@4.3.3':
    resolution: {integrity: sha512-/T8IKEsf9VTU6tLjgC7+sv2mOPtQxzE2jMw7u4Tt40Tx+QSZxpzh95/H6cMKoja9XuW7iMdLJYBB0o9G1CaAgg==}

  '@tailwindcss/oxide-android-arm64@4.3.3':
    resolution: {integrity: sha512-Y85A2gmPSkl5Ve5qR86GL4HT509cFqQh1aes9p3sSkyTPwt0Pppf3GkwGe4JPACcRYjgJIEhQgM6dBClnr0NYw==}
    engines: {node: '>= 20'}
    cpu: [arm64]
    os: [android]

  '@tailwindcss/oxide-darwin-arm64@4.3.3':
    resolution: {integrity: sha512-BiaWatpBcERQFDlOjRDpIVXuFK5PJez5SA4JMg6VYZdBYU+qKfV/vqjcIs+IYmtitf1xYQZTwXvU/8y4lfZUGw==}
    engines: {node: '>= 20'}
    cpu: [arm64]
    os: [darwin]

  '@tailwindcss/oxide-darwin-x64@4.3.3':
    resolution: {integrity: sha512-fAeUqfV5ndhxRwai8cXGzdLvul9utWOmeTkv69unv4ZXixjn61Z+p9lCWdwOwA3TYboG3BwdVuN/RDjhBRl0mw==}
    engines: {node: '>= 20'}
    cpu: [x64]
    os: [darwin]

  '@tailwindcss/oxide-freebsd-x64@4.3.3':
    resolution: {integrity: sha512-iyf5bV6+wnAlflVeEy7R25dupxTNECZN5QMI0qNT6eT+EgaGdZcKhGkr5SdoaWiLJ3spLqIY9VCeSGrwmtg4kw==}
    engines: {node: '>= 20'}
    cpu: [x64]
    os: [freebsd]

  '@tailwindcss/oxide-linux-arm-gnueabihf@4.3.3':
    resolution: {integrity: sha512-aAYUprJAJQWWbRrPvtjdroZ56Md+JM8pMiopS6xGEwDfLhqj+2ver2p4nU4Mb3CRqcMmNBjo8KkUgcxhkzVQGQ==}
    engines: {node: '>= 20'}
    cpu: [arm]
    os: [linux]

  '@tailwindcss/oxide-linux-arm64-gnu@4.3.3':
    resolution: {integrity: sha512-nDxldcEENOxZRzC2uu9jrutZdAAQtb+8WWDCSnWL1zvBk1+FN+x6MtDViPB5AJMfttVCUhehGWus3XBPgatM/w==}
    engines: {node: '>= 20'}
    cpu: [arm64]
    os: [linux]

  '@tailwindcss/oxide-linux-arm64-musl@4.3.3':
    resolution: {integrity: sha512-Md44bD6veX/PC5iyF8cDVnw4HBIANZepRZZ7a8DQOvkfo5WUBwcp6iAuCUz23u+4SUkhJlD3eL7hNdW8ezd/kA==}
    engines: {node: '>= 20'}
    cpu: [arm64]
    os: [linux]

  '@tailwindcss/oxide-linux-x64-gnu@4.3.3':
    resolution: {integrity: sha512-tx7us1muwOKAKWao2v/GaafFeQboE6aj88vC6ziN2NCGcRm8gWUhwjzg+YdVB1e4boAtdtma4L43onunI6NS4w==}
    engines: {node: '>= 20'}
    cpu: [x64]
    os: [linux]

  '@tailwindcss/oxide-linux-x64-musl@4.3.3':
    resolution: {integrity: sha512-SJxX60smvHgasZoBy11dX6YRjXJFovwWBoedhbQPOBzgFWBHGB+TVPWB9BxzR7TTxU8FQZAI2AyiNCMzFm8Img==}
    engines: {node: '>= 20'}
    cpu: [x64]
    os: [linux]

  '@tailwindcss/oxide-wasm32-wasi@4.3.3':
    resolution: {integrity: sha512-jx1+rPhY/5Ympkktd656HBWEBLxP7dH06losBLjjf5vgCODXvi9KhtftWcMIwTFIDqBr7cRnQkdLnAG+IOlGvQ==}
    engines: {node: '>=14.0.0'}
    cpu: [wasm32]
    bundledDependencies:
```

```yaml
      - '@napi-rs/wasm-runtime'
      - '@emnapi/core'
      - '@emnapi/runtime'
      - '@tybys/wasm-util'
      - '@emnapi/wasi-threads'
      - tslib

  '@tailwindcss/oxide-win32-arm64-msvc@4.3.3':
    resolution: {integrity: sha512-3rc292Ca2ceK6Ulcc/bAVnTs/3nDtoPhyEKlgPv+yQJQi/JS/AMJlqzxvlDacL1nekbrcf6bTqp/jV4qgnPxNQ==}
    engines: {node: '>= 20'}
    cpu: [arm64]
    os: [win32]

  '@tailwindcss/oxide-win32-x64-msvc@4.3.3':
    resolution: {integrity: sha512-yJ0pwIVc/nYeGoV02WtsN8KYyLQv7kyI2wDnkezyJlGGjkd4QLwDGAwl47YpPJeuI0M0ObaXGSPjvWDPeTPggw==}
    engines: {node: '>= 20'}
    cpu: [x64]
    os: [win32]

  '@tailwindcss/oxide@4.3.3':
    resolution: {integrity: sha512-krXjAikiaFSPaK/FkAQT5UTx3VormQaiZ5hBFlJZ9UFQGB/rwg1MZIhHAG9smMQRTdyJxP6Qt5MwMtdyU5FWrA==}
    engines: {node: '>= 20'}

  '@tailwindcss/postcss@4.3.3':
    resolution: {integrity: sha512-JTSZZGQi1AyKirbLN3azmjVzef92tcX7h+iSqPdaeStyFpGpDlKvvpxeOE8njhbUanbRwr3z8DyzhICWnMtQeg==}

  '@tybys/wasm-util@0.10.3':
    resolution: {integrity: sha512-F3fo1MYrRJYL3zER0OUOmkutjr1Vp23m7OsSgp7nq4SP6OqX6C/56XFIPAl5bt3zaBRjmW7SGz3u/6LwFpYcOg==}

  '@types/debug@4.1.13':
    resolution: {integrity: sha512-KSVgmQmzMwPlmtljOomayoR89W4FynCAi3E8PPs7vmDVPe84hT+vGPKkJfThkmXs0x0jAaa9U8uW8bbfyS2fWw==}

  '@types/estree-jsx@1.0.5':
    resolution: {integrity: sha512-52CcUVNFyfb1A2ALocQw/Dd1BQFNmSdkuC3BkZ6iqhdMfQz7JWOFRuJFloOzjk+6WijU56m9oKXFAXc7o3Towg==}

  '@types/estree@1.0.9':
    resolution: {integrity: sha512-GhdPgy1el4/ImP05X05Uw4cw2/M93BCUmnEvWZNStlCzEKME4Fkk+YpoA5OiHNQmoS7Cafb8Xa3Pya8m1Qrzeg==}

  '@types/hast@3.0.5':
    resolution: {integrity: sha512-rp/ezSWaD1m44dPKICGhiskI13nVr7qTloFwDa/IYkhhf5nzwP+zIQcIJh3WIFSBOy/H1PzB40jPjMDksN4F+g==}

  '@types/json-schema@7.0.15':
    resolution: {integrity: sha512-5+fP8P8MFNC+AyZCDxrB2pkZFPGzqQWUzpSeuuVLvm8VMcorNYavBqoFcxK8bQz4Qsbn4oUEEem4wDLfcysGHA==}

  '@types/json5@0.0.29':
    resolution: {integrity: sha512-dRLjCWHYg4oaA77cxO64oO+7JwCwnIzkZPdrrC71jQmQtlhM556pwKo5bUzqvZndkVbeFLIIi+9TC40JNF5hNQ==}

  '@types/mdast@4.0.4':
    resolution: {integrity: sha512-kGaNbPh1k7AFzgpud/gMdvIm5xuECykRR+JnWKQno9TAXVa6WIVCGTPvYGekIDL4uwCZQSYbUxNBSb1aUo79oA==}

  '@types/ms@2.1.0':
    resolution: {integrity: sha512-GsCCIZDE/p3i96vtEqx+7dBUGXrc7zeSK3wwPHIaRThS+9OhWIXRqzs4d6k1SVU8g91DrNRWxWUGhp5KXQb2VA==}

  '@types/node@22.20.1':
    resolution: {integrity: sha512-EANqOCF9QFyra+4pfxUcX9STKJpCLjMbObVzljIJomAWSnuSIEAvyzEU53GaajbXJEgdh0iEcPL+DGvpUd4k1Q==}

  '@types/react-dom@19.2.4':
    resolution: {integrity: sha512-Bsc+QHgp+P/F02XDzNCY9jnZNCUuLki36KT7VKrTXXLdHf+vHMNZnW1rVu5DNW/rCK+fya3DATySbLM4yhtKUw==}
    peerDependencies:
      '@types/react': ^19.2.0

  '@types/react@19.2.18':
    resolution: {integrity: sha512-AnzbBERsrLKtk2XSfTbYRLjQPdy116Sty4q+T+Bp3IC4l6jNBvreVPAHmpq9qhXQM7CXZPjLVmGMw9sy+hxQ3w==}

  '@types/unist@2.0.11':
    resolution: {integrity: sha512-CmBKiL6NNo/OqgmMn95Fk9Whlp2mtvIv+KNpQKN2F4SjvrEesubTRWGYSg+BnWZOnlCaSTU1sMpsBOzgbYhnsA==}

  '@types/unist@3.0.3':
    resolution: {integrity: sha512-ko/gIFJRv177XgZsZcBwnqJN5x/Gien8qNOn0D5bQU/zAzVf9Zt3BlcUiLqhV9y4ARk0GbT3tnUiPNgnTXzc/Q==}

  '@typescript-eslint/eslint-plugin@8.67.0':
    resolution: {integrity: sha512-Un7Heoyj65NREbKAyIrFxeM143NZpExWmy1Nep4DLeQOeLlTeumPjoNKnBrU5D5moWXbPJgRa5Uwcdu0faVNGQ==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}
    peerDependencies:
      '@typescript-eslint/parser': ^8.67.0
      eslint: ^8.57.0 || ^9.0.0 || ^10.0.0
      typescript: '>=4.8.4 <6.1.0'

  '@typescript-eslint/parser@8.67.0':
    resolution: {integrity: sha512-fUBfTuuEulWqX6V8+O3PtScV01tzYYRUDTAirHFKoRAt7nOzoGiPt0M/bB47wWNy0coOOcgEwAMUtBpykMxl6w==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}
    peerDependencies:
      eslint: ^8.57.0 || ^9.0.0 || ^10.0.0
      typescript: '>=4.8.4 <6.1.0'

  '@typescript-eslint/project-service@8.67.0':
    resolution: {integrity: sha512-cvE8c7ulYeXN9fYuszhCeCsbzyVEXuhrRCybnBre7TUmqb5nRmBfQAwCj0O3WJFDeyAZt4VYv51vMCC9LHSdYw==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}
    peerDependencies:
      typescript: '>=4.8.4 <6.1.0'

  '@typescript-eslint/scope-manager@8.67.0':
    resolution: {integrity: sha512-EgvsleTwS4E+WzzSvem8fAUubLwatMNF1B5hHSLQxcvs7q2dtRhGyujHwLJSYlG41niJ7GP24Aha2+0mb1b2kg==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}

  '@typescript-eslint/tsconfig-utils@8.67.0':
    resolution: {integrity: sha512-vV+LUSv5njUWsknE71fqKTlXUva+R76SaeORd6Zojcunk/6DvKFXONU3BrAs2H49mbygUXt6gbYunzwqNwlhdg==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}
    peerDependencies:
      typescript: '>=4.8.4 <6.1.0'

  '@typescript-eslint/type-utils@8.67.0':
    resolution: {integrity: sha512-aVWDXbRmdXO9siTfX4ditQI1T9+zVcNazT48EJCD0v40/9RIFoUgZ05CmGEq9H2gixRpjUn/iplwvlcvutJW/Q==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}
    peerDependencies:
```

```
      eslint: ^8.57.0 || ^9.0.0 || ^10.0.0
      typescript: '>=4.8.4 <6.1.0'

  '@typescript-eslint/types@8.67.0':
    resolution: {integrity: sha512-sBtgslww8nsMYUjhdPBiSyUqSzT8uR6g93A2QXnQC8+cGdjz0CyaOdqHDRJb1AtORbZCNUJBBeFA/tNR2uQmww==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}

  '@typescript-eslint/typescript-estree@8.67.0':
    resolution: {integrity: sha512-EKQBCE9yNlRJYm7jdTW5AhDacDUmSwQb0FAJAmK2EKYrNXIsa2vxcSZx6PvJ/dEdI6lS+Y9W+EXckLj0iPFGcw==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}
    peerDependencies:
      typescript: '>=4.8.4 <6.1.0'

  '@typescript-eslint/utils@8.67.0':
    resolution: {integrity: sha512-U9D1FdwEWBwok3hxxSdhclMb0twvt9QnjIQ0VfQ1AiX2epnpSgv2ubVDsayOFyY8K6FX+AQ7E0FKWVG3iKsj1A==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}
    peerDependencies:
      eslint: ^8.57.0 || ^9.0.0 || ^10.0.0
      typescript: '>=4.8.4 <6.1.0'

  '@typescript-eslint/visitor-keys@8.67.0':
    resolution: {integrity: sha512-fkv8dHRDqfGtTHuJeebdrQ7cX6Ad4WAS00rgHh9UGvMycF1mjBfsxry1XsLIFhWZ6Judlh6UdzK+TYlbpCXgnA==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}

  '@ungap/structured-clone@1.3.3':
    resolution: {integrity: sha512-60YRaenCQcVjYEKOcG824+DRGGIQ3VKErcBoAEDJZz5bKIs2ZG+X/H9Nk+Q6EVkwJk5QNApxbrc5QtBSwtrXAg==}

  '@unrs/resolver-binding-android-arm-eabi@1.12.2':
    resolution: {integrity: sha512-g5T90pqg1bo/7mytQx6F4iBNC0Wsh9cu+z9veDbFjc7HjpesJFWD7QMS0NGStXM075+7dJPPVvBbpZlnrdpi/w==}
    cpu: [arm]
    os: [android]

  '@unrs/resolver-binding-android-arm64@1.12.2':
    resolution: {integrity: sha512-YGCRZv/9GLhwmz6mYDeTsm/92BAyR28l6c2ReweVW5pWgfsitWLY8upvfRlGdoyD8HjeTHSYJWyZGD4KJA/nFQ==}
    cpu: [arm64]
    os: [android]

  '@unrs/resolver-binding-darwin-arm64@1.12.2':
    resolution: {integrity: sha512-u9DiNT1auQMO20A9SyTuG3wUgQWB9Z7KjAg0uFuCDR1FsAY8A0CG2S6JpHS1xwm/w1G08bjXZDcyOCjv1WAm2w==}
    cpu: [arm64]
    os: [darwin]

  '@unrs/resolver-binding-darwin-x64@1.12.2':
    resolution: {integrity: sha512-f7rPLi/T1HVKZu/u6t87lroib16n8vrSzcyxI7lg4BGO9UF26KhQL44sd9eOUgrTYhvRXtWOIZT5PejdPyJfUA==}
    cpu: [x64]
    os: [darwin]

  '@unrs/resolver-binding-freebsd-x64@1.12.2':
    resolution: {integrity: sha512-BpcOjWCJub6nRZUS2zA20pmLvjtqAtGejETaIyRLiZiQf++cbrjltLA5NN/xaXfqeOBOSlMFbemIl5/S5tljmg==}
    cpu: [x64]
    os: [freebsd]

  '@unrs/resolver-binding-linux-arm-gnueabihf@1.12.2':
    resolution: {integrity: sha512-vZTDvdSISZjJx66OzJqtsOhzifbqRjbmI1Mnu49fQDwog5GtDI4QidRiEAYbZCRj9C8YZEW+3ZjqsyS9GR4k2A==}
    cpu: [arm]
    os: [linux]

  '@unrs/resolver-binding-linux-arm-musleabihf@1.12.2':
    resolution: {integrity: sha512-BiPI+IrIlwcW4nLLMM21+B1dFPzd55yAVgVGrdgDjNef+ch03GdxrcyaIz8X9SsQirh/kCQ7mviyWlMxdh2D7g==}
    cpu: [arm]
    os: [linux]

  '@unrs/resolver-binding-linux-arm64-gnu@1.12.2':
    resolution: {integrity: sha512-zJc0H99FEPoFfSrNpa91HYfxzfAJCr502oxNK1cfdC9hlaFI43RT+JFCann9JUgZmLzzntChHyn13Sgn9ljHNg==}
    cpu: [arm64]
    os: [linux]

  '@unrs/resolver-binding-linux-arm64-musl@1.12.2':
    resolution: {integrity: sha512-KQ3Lki6l+Pz1k/eBipN41ES+YUK30beLGb9YqcB1O542cyLCNE6GaxrfcY3T6EezmGGk84wb5XyO9loTM9tkcA==}
    cpu: [arm64]
    os: [linux]

  '@unrs/resolver-binding-linux-loong64-gnu@1.12.2':
    resolution: {integrity: sha512-3SJGEh1DborhG6pyxvhPzCT4bbSIVihsvgJc13P1bHG7KLdNDaF9T3gsTwFc7Jw/5Y5/iWOjkEx7Zy0NvCGX3Q==}
    cpu: [loong64]
    os: [linux]

  '@unrs/resolver-binding-linux-loong64-musl@1.12.2':
    resolution: {integrity: sha512-jiuG/Obbel7uw1PwHNFfrkiKhLAF6mnyZ6aWlOAVN9WqKm8v0OFGnciJIHu8+CMvXLQ8AD51LPzAoUfT21D5Ew==}
    cpu: [loong64]
    os: [linux]

  '@unrs/resolver-binding-linux-ppc64-gnu@1.12.2':
    resolution: {integrity: sha512-q7xRvVpmcfeL+LlZg8Pbbo6QaTZwDU5BaGZbwfhkEsXJn3Was8xYfE0RBH266xZt0rM6B7i8xAYIvjthuUIWHg==}
    cpu: [ppc64]
    os: [linux]

  '@unrs/resolver-binding-linux-riscv64-gnu@1.12.2':
    resolution: {integrity: sha512-0CVdx6lcnT3Q9inOH8tsMIOJ6ImndllMjqJHg8RLVdB7Vq4SfkEXl9mCSsVNuNA4MCYycRicCUxPCabVHJRr6A==}
    cpu: [riscv64]
    os: [linux]

  '@unrs/resolver-binding-linux-riscv64-musl@1.12.2':
    resolution: {integrity: sha512-iOwlRo9vnp6R6ohHQS11n0NnfdXx/omhkocmIfaPRpQhKZ+3BDMkkdRVh53qjkFkpPddf+FETA28NwGN7l5l+w==}
    cpu: [riscv64]
    os: [linux]

  '@unrs/resolver-binding-linux-s390x-gnu@1.12.2':
    resolution: {integrity: sha512-HYJtLfXq94q8iZNFT1lknx258wlkkWhZeUXJRqzKBBUJ00CvZ+N33zgbCqimLjsyw5Va6uUxhVa12mI+kaveEw==}
    cpu: [s390x]
    os: [linux]

  '@unrs/resolver-binding-linux-x64-gnu@1.12.2':
    resolution: {integrity: sha512-mPsUhunKKDih5O96Y6enDQyHc1SqBPlY1E/SfMWDM3EdJ95Z9CArPeCVwCCqbP45ljvivdEk8Fxn+SIb1rDAJQ==}
    cpu: [x64]
```

```
    os: [linux]

  '@unrs/resolver-binding-linux-x64-musl@1.12.2':
    resolution: {integrity: sha512-azrt6+5ydLd8Vt210AAFis/lZevSfPw93EJRIJG+xPu4WCJ8K0kppCTpMyLPcKT7H15M4Jnt2tMp5bOvCkRC6A==}
    cpu: [x64]
    os: [linux]

  '@unrs/resolver-binding-openharmony-arm64@1.12.2':
    resolution: {integrity: sha512-YZ9hP4O0X9PQb8eO980qmLNGH4zT3I9+SZTdt0Pr0YyuGQhYKoOZkV02VzrzyOZJ5xIJ3UFIenKkUkGg8GjgWQ==}
    cpu: [arm64]
    os: [openharmony]

  '@unrs/resolver-binding-wasm32-wasi@1.12.2':
    resolution: {integrity: sha512-tYFDIkMxSflfEc/h92ZWNsZlHSwgimbNHSO3PL2JWQHfCuC2q316jMyYU9TIWZsFK2bQwyK5VAdYgn8ygPj69A==}
    engines: {node: '>=14.0.0'}
    cpu: [wasm32]

  '@unrs/resolver-binding-win32-arm64-msvc@1.12.2':
    resolution: {integrity: sha512-qzNyg3xL0VPQmCaUh+N5jSitce6k+uCBfMDesWRnlULOZaqUkaJ0ybdT+UqlAWJoQjuqfIU/0Ptx9bteN4D82g==}
    cpu: [arm64]
    os: [win32]

  '@unrs/resolver-binding-win32-ia32-msvc@1.12.2':
    resolution: {integrity: sha512-WD9sY00OfpHVGfsnHZoA8jVT+esS/Bg8z8jzxp5BnDCjjwsuKsPQrzswwpFy4J1AUJbXPRfkpcX0mXrzeXW79g==}
    cpu: [ia32]
    os: [win32]

  '@unrs/resolver-binding-win32-x64-msvc@1.12.2':
    resolution: {integrity: sha512-nAB74NfSNKknqQ1RrYj6uz8FcXEomu/MATJZxh/x+BArzN2U3JbOYC0APYzUIGhVY3m5hRxA8VPNdPBoG8txlA==}
    cpu: [x64]
    os: [win32]

  acorn-jsx@5.3.2:
    resolution: {integrity: sha512-rq9s+JNhf0IChjtDXxllJ7g41oZk5SlXtp0LHwyA5cejwn7vKmKp4pPri6YEePv2PU65sAsegbXtIinmDFDXgQ==}
    peerDependencies:
      acorn: ^6.0.0 || ^7.0.0 || ^8.0.0

  acorn@8.18.0:
    resolution: {integrity: sha512-lGq+9yr1/GuAWaVYIHRjvvySG5/4VfKIvC8EWxStPdcDh/Ka7FG3twP6v4d5BkravUilhIAsG4Qj83t02LWUPQ==}
    engines: {node: '>=0.4.0'}
    hasBin: true

  ajv@6.15.0:
    resolution: {integrity: sha512-fgFx7Hfoq60ytK2c7DhnF8jIvzYgOMxfugjLOSMHjLIPgenqa7S7oaagATUq99mV6IYvN2tRmC0wnTYX6iPbMw==}

  ansi-styles@4.3.0:
    resolution: {integrity: sha512-zbB9rCJAT1rbjiVDb2hqKFHNYLxgtk8NURxZ3IZwD3F6NtxbXZQCnnSi1Lkx+IDohdPlFp222wVALIheZJQSEg==}
    engines: {node: '>=8'}

  argparse@2.0.1:
    resolution: {integrity: sha512-8+9WqebbFzpX9OR+Wa6O29asIogeRMzcGtAINdpMHHyAg10f05aSFVBbcEqGf/PXw1EjAZ+q2/bEBg3DvurK3Q==}

  aria-hidden@1.2.6:
    resolution: {integrity: sha512-ik3ZgC9dY/lYVVM++OISsaYDeg1tb0VtP5uL3ouh1koGOaUMDPpbFIei4JkFimWUFPn90sbMNMXQAIVOlnYKJA==}
    engines: {node: '>=10'}

  aria-query@5.3.2:
    resolution: {integrity: sha512-COROpnaoap1E2F000S62r6A60uHZnmlvomhfyT2DlTcrY1OrBKn2UhH7qn5wTC9zMvD0AY7csdPSNwKP+7WiQw==}
    engines: {node: '>= 0.4'}

  array-buffer-byte-length@1.0.2:
    resolution: {integrity: sha512-LHE+8BuR7RYGDKvnrmcuSq3tDcKv9OFEXQt/HpbZhY7V6h0zlUXutnAD82GiFx9rdieCMjkvtcsPqBwgUl1Iiw==}
    engines: {node: '>= 0.4'}

  array-includes@3.1.9:
    resolution: {integrity: sha512-FmeCCAenzH0KH381SPT5FZmiA/TmpndpcaShhfgEN9eCVjnFBqq3l1xrI42y8+PPLI6hypzou4GXw00WHmPBLQ==}
    engines: {node: '>= 0.4'}

  array.prototype.findlast@1.2.5:
    resolution: {integrity: sha512-CVvd6FHg1Z3POpBLxO6E6zr+rSKEQ9L6rZHAaY7lLfhKsWYUBBOuMs0e9o24oopj6H+geRCX0YJ+TJLBK2eHyQ==}
    engines: {node: '>= 0.4'}

  array.prototype.findlastindex@1.2.6:
    resolution: {integrity: sha512-F/TKATkzseUExPlfvmwQKGITM3DGTK+vkAsCZoDc5daVygbJBnjEUCbgkAvVFsgfXfX4YIqZ/27G3k3tdXrTxQ==}
    engines: {node: '>= 0.4'}

  array.prototype.flat@1.3.3:
    resolution: {integrity: sha512-rwG/ja1neyLqCuGZ5YYrznA62D4mZXg0i1cIskIUKSiqF3Cje9/wXAls9B9s1Wa2fomMsIv8czB8jZcPmxCXFg==}
    engines: {node: '>= 0.4'}

  array.prototype.flatmap@1.3.3:
    resolution: {integrity: sha512-Y7Wt51eKJSyi80hFrJCePGGNo5ktJCslFuboqJsbf57CCPcm5zztluPlc4/aD8sWsKvlwatezpV4U1efk8kpjg==}
    engines: {node: '>= 0.4'}

  array.prototype.tosorted@1.1.4:
    resolution: {integrity: sha512-p6Fx8B7b7ZhL/gmUsAy0D15WhvDccw3mnGNbZpi3pmeJdxtWsj2jEaI4Y6oo3XiHfzuSgPwKc04MYt6KgvC/wA==}
    engines: {node: '>= 0.4'}

  arraybuffer.prototype.slice@1.0.4:
    resolution: {integrity: sha512-BNoCY6SXXPQ7gF2opIP4GBE+Xw7U+pHMYKuzjgCN3GwiaIR09UUeKfheyIry77QtrCBlC0KK0q5/TER/tYh3PQ==}
    engines: {node: '>= 0.4'}

  ast-types-flow@0.0.8:
    resolution: {integrity: sha512-OH/2E5Fg20h2aPrbe+QL8JZQFko0YZaF+j4mnQ7BGhfavO7OpSLa8a0y9sBwomHdSbkhTS8TQNayBfnW5DwbvQ==}

  async-function@1.0.0:
    resolution: {integrity: sha512-hsU18Ae8CDTR6Kgu9DYf0EbCr/a5iGL0rytQDobUcdpYOKokk8LEjVphnXkDkgpi0wYVsqrXuP0bZxJaTqdgoA==}
    engines: {node: '>= 0.4'}

  available-typed-arrays@1.0.7:
    resolution: {integrity: sha512-wvUjBtSGN7+7SjNpq/9M2Tg350UZD3q62IFZLbRAR1bSMlCo1ZaeW+BJ+D090e4hIIZLBcTDWe4Mh4jvUDajzQ==}
    engines: {node: '>= 0.4'}

  axe-core@4.13.0:
    resolution: {integrity: sha512-UzGt8zg7Ny8djbYMhxl2zuEevVa7r2gJjYY5Lwr1xM7+XU2nd6CkIWFTVcCIbAP63vSz71NaVyyuSk9lHKcy0A==}
```

```
    engines: {node: '>=4'}

  axobject-query@4.1.0:
    resolution: {integrity: sha512-qIj0G9wZbMGNLjLmg1PT6v2mE9AH2zlnADJD/2tC6E00hgmhUOfEB6greHPAfLRSufHqROIUTkw6E+M3lH0PTQ==}
    engines: {node: '>= 0.4'}

  bail@2.0.2:
    resolution: {integrity: sha512-0xO6mYd7JB2YesxDKplafRpsiOzPt9V02ddPCLbY1xYGPOX24NTyN50qnUxgCPcSoYMhKpAuBTjQoRZCAkUDRw==}

  balanced-match@1.0.2:
    resolution: {integrity: sha512-3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbxRDZOpH0KW9+CeX4bRAaX0Anxt0tx2MrpRpWwQaPwIlISEJhYU5Pw==}

  balanced-match@4.0.4:
    resolution: {integrity: sha512-BLrgEcRTwX2o6gGxGOCNyMvGSp35YofuYzw9h1IMTRmKqttAZZVU67bdb9Pr2vUHA8+j3i2tJfjO6C6+4myGTA==}
    engines: {node: 18 || 20 || >=22}

  brace-expansion@1.1.18:
    resolution: {integrity: sha512-Edep/X9fGqVNmzKBVsDYIOtD+z1tuezV70LBjdCst9Tqu76lsnvRiZ6oTic1n+/BIwX6QDGAO94PN4N2SADvtw==}

  brace-expansion@5.0.9:
    resolution: {integrity: sha512-ScQ4IuvIEF1TMlP7Zt+vjJ//9zlPb2SDcxWxM3bk8s6t6GGdJ7KO1dCcTidOPJKePW30LE/2cT7wCyPho9/Wxg==}
    engines: {node: 20 || >=22}

  braces@3.0.3:
    resolution: {integrity: sha512-yQbXgO/OSZVD2IsiLlro+7Hf6Q18EJrKSEsdoMzKePKXct3gvD8oLcOQdIzGupr5Fj+EDe8gO/lxc1BzfMpxvA==}
    engines: {node: '>=8'}

  call-bind-apply-helpers@1.0.2:
    resolution: {integrity: sha512-Sp1ablJ0ivDkSzjcaJdxEunN5/XvksFJ2sMBFfq6x0ryhQV/2b/KwFe21cMpmHtPOSij8K99/wSfoEuTObmuMQ==}
    engines: {node: '>= 0.4'}

  call-bind@1.0.9:
    resolution: {integrity: sha512-a/hy+pNsFUTR+Iz8TCJvXudKVLAnz/DyeSUo10I5yvFDQJBFU2s9uqQpoSrJlroHUKoKqzg+epxyP9lqFdzfBQ==}
    engines: {node: '>= 0.4'}

  call-bound@1.0.4:
    resolution: {integrity: sha512-+ys997U96po4Kx/ABpBCqhA9EuxJaQWDQg7295H4hBphv3IZg0boBKuwYpt4YXp6MZ5AmZQnU/tyMTlRpaSejg==}
    engines: {node: '>= 0.4'}

  callsites@3.1.0:
    resolution: {integrity: sha512-P8BjAsXvZS+VIDUI11hHCQEv74YT67YUi5JJFNWIqL235sBmjX4+qx9Muvls5ivyNENctx46xQLQ3aTuE7ssaQ==}
    engines: {node: '>=6'}

  caniuse-lite@1.0.30001809:
    resolution: {integrity: sha512-xxWVywk6a6Arlk+hymeycyn/VgqEfLDxupvhH/xiY5SJ/18kmi9o6MiO320DCUzypORHLtvh0I4i04tUhCNHNQ==}

  ccount@2.0.1:
    resolution: {integrity: sha512-eyrF0jiFpY+3drT6383f1qhkbGsLSifNAjA61IUjZjmLCWjItY6LB9ft9YhoDgwfmclB2zhu51Lc7+95b8NRAg==}

  chalk@4.1.2:
    resolution: {integrity: sha512-oKnbhFyRIXpUuez8iBMmyEa4nbj4IOQyuhc/wy9kY7/WVPcwIO9VA668Pu8RkO7+0G76SLROeyw9CpQ061i4mA==}
    engines: {node: '>=10'}

  character-entities-html4@2.1.0:
    resolution: {integrity: sha512-1v7fgQRj6hnSwFpq1Eu0ynr/CDEw0rXo2B61qXrLNdHZmPKgb7fqS1a2JwF0rISo9q77jDI8VMEHoApn8qDoZA==}

  character-entities-legacy@3.0.0:
    resolution: {integrity: sha512-RpPp0asT/6ufRm//AJVwpViZbGM/MkjQFxJccQRHmISF/22NBtsHqAWmL+/pmkPWoIUJdWyeVleTl1wydHATVQ==}

  character-entities@2.0.2:
    resolution: {integrity: sha512-shx7oQ0Awen/BRIdkjkvz54PnEEI/EjwXDSIZp86/KKdbafHh1Df/RYGBhn4hbe2+uKC9FnT5UCEdyPz3ai9hQ==}

  character-reference-invalid@2.0.1:
    resolution: {integrity: sha512-iBZ4F4wRbyORVsu0jPV7gXkOsGYjGHPmAyv+HiHG8gi5PtC9KI2j1+v8/tlibRvjoWX027ypmG/n0HtO5t7unw==}

  class-variance-authority@0.7.1:
    resolution: {integrity: sha512-Ka+9Trutv7G8M6WT6SeiRWz792K5qEqIGEGzXKhAE6xOWAY6pPH8U+9IY3oCMv6kqTmLsv7Xh/2w2RigkePMsg==}

  client-only@0.0.1:
    resolution: {integrity: sha512-IV3Ou0jSMzZrd3pZ48nLkT9DA7Ag1pnPzaiQhpW7c3Rbcqqzvzz Vu+L8gfqMp/8IM2MQtSiqaCxrrcfu8I8rMA==}

  clsx@2.1.1:
    resolution: {integrity: sha512-eYm0QWBtUrBWZWG0d386OGAw16Z995PiOVo2B7bjWSbHedGl5e0ZWaq65kOGgUSNesEIDkB9ISbTg/JK9dhCZA==}
    engines: {node: '>=6'}

  color-convert@2.0.1:
    resolution: {integrity: sha512-RRECPsj7iu/xb5oKYcsFHSppFNnsj/52OVTRKb4zP5onXwVF3zVmmToNcOfGC+CRDpfK/U584fMg38ZHCaElKQ==}
    engines: {node: '>=7.0.0'}

  color-name@1.1.4:
    resolution: {integrity: sha512-dOy+3AuW3a2wNbZHIuMZpTcgjGuLU/uBL/ubcZF9OXbDo8ff4O8yVp5Bf0efS8uEoYo5q4Fx7dY9OgQGXgAsQA==}

  comma-separated-tokens@2.0.3:
    resolution: {integrity: sha512-Fu4hJdvzeylCfQPp9SGWidpzrMs7tTrlu6Vb8XGaRGck8QSNZJJp538Wrb60Lax4fPwR64ViY468OIUTbRlGZg==}

  concat-map@0.0.1:
    resolution: {integrity: sha512-/Srv4dswyQNBfohGpz9o6Yb3Gz3SrUDqBH5rTuhGR7ahtlbYKnVxw2bCFMRljaA7EXHaXZ8wsHdodFvbkhKmqg==}

  cross-spawn@7.0.6:
    resolution: {integrity: sha512-uV2QOWP2nWzsy2aMp8aRibhi9dlzF5Hgh5SHaB9OiTGEyDTiJJyx0uy51QXdyWbtAHNua4XJzUKca3OzKUd3vA==}
    engines: {node: '>= 8'}

  csstype@3.2.3:
    resolution: {integrity: sha512-z1HGKcYy2xA8AGQfwrn0PAy+PB7X/GSj3UVJW9qKyn43xWa+gl5nXmU4qqLMRzWVLFC8KusUX8T/0kCiOYpAIQ==}

  damerau-levenshtein@1.0.8:
    resolution: {integrity: sha512-sdQSFB7+llfUcQHUQO3+B8ERRj0Oa4w9POWMI/puGtuf7gFywGmkaLCElnudfTiKZV+NvHqL0ifzdrI8Ro7ESA==}

  data-view-buffer@1.0.2:
    resolution: {integrity: sha512-EmKO5V3OLXh1rtK2wgXRansaK1/mtVdTUEiEI0W8RkvgT05kfxaH29PliLnpLP73yYO6142Q72QNa8Wx/A5CqQ==}
    engines: {node: '>= 0.4'}

  data-view-byte-length@1.0.2:
    resolution: {integrity: sha512-tuhGbE6CfTM9+5ANGf+oQb72Ky/0+s3xKUpHvShfiz2RxMFgFPjsXuRLBVMtvMs15awe45SRb83D6wH4ew6wlQ==}
    engines: {node: '>= 0.4'}
```

```yaml

  data-view-byte-offset@1.0.1:
    resolution: {integrity: sha512-BS8PfmtDGnrgYdOonGZQdLZslWIeCGFP9tpan0hi1Co2Zr2NKADsvGYA8XxuG/4UWgJ6Cjtv+YJnB6MM69QGlQ==}
    engines: {node: '>= 0.4'}

  debug@3.2.7:
    resolution: {integrity: sha512-CFjzYYAi4ThfiQvizrFQevTTXHtnCqWfe7x1AhgEscTz6ZbLbfoLRLPugTQyBth6f8ZERVUSyWHFD/7Wu4t1XQ==}
    peerDependencies:
      supports-color: '*'
    peerDependenciesMeta:
      supports-color:
        optional: true

  debug@4.4.3:
    resolution: {integrity: sha512-RGwwWnwQvkVfavKVt22FGLw+xYSdzARwm0ru6DhTVA3umU5hZc28V3kO4stgYryrTlLpuvgI9GiijltAjNbcqA==}
    engines: {node: '>=6.0'}
    peerDependencies:
      supports-color: '*'
    peerDependenciesMeta:
      supports-color:
        optional: true

  decode-named-character-reference@1.3.0:
    resolution: {integrity: sha512-GtpQYB283KrPp6nRw50q3U9/VfOutZOe103qlN7BPP6Ad27xYnOIWv4lPzo8HCAL+mMZofJ9KEy30fq6MfaK6Q==}

  deep-is@0.1.4:
    resolution: {integrity: sha512-oIPzksmTg4/MriiaYGO+okXDT7ztn/w3Eptv/+gSIdMdKsJo0u4CfYNFJPy+4SKMuCqGw2wxnA+URMg3t8a/bQ==}

  define-data-property@1.1.4:
    resolution: {integrity: sha512-rBMvIzlpA8v6E+SJZoo++HAYqsLrkg7MSfIinMPFhmkorw7X+dOXVJQs+QT69zGkzMyfDnIMN2Wid1+NbL3T+A==}
    engines: {node: '>= 0.4'}

  define-properties@1.2.1:
    resolution: {integrity: sha512-8QmQKqEASLd5nx0U1B1okLElbUuuttJ/AnYmRXbbbGDWh6uS208EjD4Xqq/I9wK7u0v6O08XhTWnt5XtEbR6Dg==}
    engines: {node: '>= 0.4'}

  dequal@2.0.3:
    resolution: {integrity: sha512-0je+qPKHEMohvfRTCEo3CrPG6cAzAYgmzKyxRiYSSDkS6eGJdyVJm7WaYA5ECaAD9wLB2T4EEeymA5aFVcYXCA==}
    engines: {node: '>=6'}

  detect-libc@2.1.2:
    resolution: {integrity: sha512-Btj2BOOO83o3WyH59e8MgXsxEQVcarkUOpEYrubB0urwnN10yQ364rsiByU11nZlqWYZm05i/of7io4mzihBtQ==}
    engines: {node: '>=8'}

  detect-node-es@1.1.0:
    resolution: {integrity: sha512-ypdmJU/TbBby2Dxibuv7ZLW3Bs1QEmM7nHjEANfohJLvE0XVujisn1qPJcZxg+qDucsr+bP6fLD1rPS3AhJ7EQ==}

  devlop@1.1.0:
    resolution: {integrity: sha512-RWmIqhcFf1lRYBvNmr7qTNuyCt/7/ns2jbpp1+PalgE/rDQcBT0fioSMUpJ93irlUhC5hrg4cYqe6U+0ImW0rA==}

  doctrine@2.1.0:
    resolution: {integrity: sha512-35mSku4ZXK0vfCuHEDAwt55dg2jNajHZ1odvF+8SSr82EsZY4QmXfuWso8oEd8zRhVObSN18aM0CjSdoBX7zIw==}
    engines: {node: '>=0.10.0'}

  dunder-proto@1.0.1:
    resolution: {integrity: sha512-KIN/nDJBQRcXw0MLVhZE9iQHmG68qAVIBg9CqmUYjmQIhgij9U5MFvrqkUL5FbtyyzZuOeOt0zdeRe4UY7ct+A==}
    engines: {node: '>= 0.4'}

  emoji-regex@9.2.2:
    resolution: {integrity: sha512-L18DaJsXSUk2+42pv8mLs5jJT2hqFkFE4j21wOmgbUqsZ2hL72NsUU785g9RXgo3s0ZNgVl42TiHp3ZtOv/Vyg==}

  enhanced-resolve@5.24.5:
    resolution: {integrity: sha512-L1l8TNvomm6UVW5B253AGxQagSQr+vGwhMlrrfRS2qmhx46AMpMVJKQYLvWYbysTMY8VoicOvzHzoHMbyzB+4A==}
    engines: {node: '>=10.13.0'}

  entities@6.0.1:
    resolution: {integrity: sha512-aN97NXWF6AWBTahfVOIrB/NShkzi5H7F9r1s9mD3cDj4Ko5f2qhhVoYMibXF7GlLveb/D2ioWay8lxI97Ven3g==}
    engines: {node: '>=0.12'}

  es-abstract-get@1.0.0:
    resolution: {integrity: sha512-6PMWXpdhshVvFp+FoWYs1EvG1Nj0tvk0dZM+XcK0xMEM1czRVcP6ohqPWHy6qPagSpC8j4+p89WXlT+xXJs/fg==}
    engines: {node: '>= 0.4'}

  es-abstract@1.24.2:
    resolution: {integrity: sha512-2FpH9Q5i2RRwyEP1AylXe6nYLR5OhaJTZwmlcP0dL/+JCbgg7yyEo/sEK6HeGZRf3dFpWwThaRHVApXSkW3xeg==}
    engines: {node: '>= 0.4'}

  es-define-property@1.0.1:
    resolution: {integrity: sha512-e3nRfgfUZ4rNGL232gUgX06QNyyez04KdjFrF+LTRoOXmrOgFKDg4BCdsjW8EnT69eqdYGmRpJwiPVYNrCaW3g==}
    engines: {node: '>= 0.4'}

  es-errors@1.3.0:
    resolution: {integrity: sha512-Zf5H2Kxt2xjTvbJvP2ZWLEICxA6j+hAmMzIlypy4xcBg1vKVnx89Wy0GbS+kf5cwCVFFzdCFh2XSCFNULS6csw==}
    engines: {node: '>= 0.4'}

  es-iterator-helpers@1.4.0:
    resolution: {integrity: sha512-c/A0P0oxkACDc+cKWw8evLXK83oBKgn0qPOqCYT4x9uolpCIJAcYvJC9QYKNDRPsTeGyCrQ326jrvgZWdCdK5Q==}
    engines: {node: '>= 0.4'}

  es-object-atoms@1.1.2:
    resolution: {integrity: sha512-HWcBoN6NileqtSydK2FqHbS/LoDd2pqrnQHLyJzBj4kOp/ky2MWMN694xOfkK8/SnUsW2DH7EfyVlydKCsm1Zw==}
    engines: {node: '>= 0.4'}

  es-set-tostringtag@2.1.0:
    resolution: {integrity: sha512-j6vWzfrGVfyXxge+O0x5sh6cvxAog0a/4Rdd2K36zCMV5eJ+/+tOAngRO8cODMNWbVRdVlmGZQL2YS3yR8bIUA==}
    engines: {node: '>= 0.4'}

  es-shim-unscopables@1.1.0:
    resolution: {integrity: sha512-d9T8ucsEhh8Bi1woXCf+TIKDIROLG5WCkxg8geBCbvk22kzwC5G2OnXVMO6FUsvQlgUUXQ2itephWDLqDzbeCw==}
    engines: {node: '>= 0.4'}

  es-to-primitive@1.3.4:
    resolution: {integrity: sha512-yPDz7wqpg1/mmHLmS3tcfTfbw5f1eryXvyghYBffGdERwe+mV7ZcWzTR8LR17Kvqt3qfPurjlonmnq3MKXIOXw==}
    engines: {node: '>= 0.4'}

```

```yaml
  escape-string-regexp@4.0.0:
    resolution: {integrity: sha512-TtpcNJ3XAzx3Gq8sWRzJaVajRs0uVxA2YAkdb1jm2YkPz4G6egUFAyA3n5vtEIZefPk5Wa4UXbKuS5fKkJWdgA==}
    engines: {node: '>=10'}

  escape-string-regexp@5.0.0:
    resolution: {integrity: sha512-/veY75JbMK4j1yjvuUxuVsiS/hr/4iHs9FTT6cgTexxdE0Ly/glccBAkloH/DofkjRbZU3bnoj38mOmhkZ0lHw==}
    engines: {node: '>=12'}

  eslint-config-next@15.5.23:
    resolution: {integrity: sha512-z4WcTXNqFHwMG4V8WHb2xrlEPJvwarZa+H/6CR28vxr53icRnQzGXviO11p748BwrMZGl52itdPRzZfzYo0SKw==}
    peerDependencies:
      eslint: ^7.23.0 || ^8.0.0 || ^9.0.0
      typescript: '>=3.3.1'
    peerDependenciesMeta:
      typescript:
        optional: true

  eslint-import-resolver-node@0.3.10:
    resolution: {integrity: sha512-tRrKqFyCaKict5hOd244sL6EQFNycnMQnBe+j8uqGNXYzsImGbGUU4ibtoaBmv5FLwJwcFJNeg1GeVjQfbMrDQ==}

  eslint-import-resolver-typescript@3.10.1:
    resolution: {integrity: sha512-A1rHYb06zjMGAxdLSkN2fXPBwuSaQ0iO5M/hdyS0Ajj1VBaRp0sPD3dn1FhME3c/JluGFbwSxyCfqdSbtQLAHQ==}
    engines: {node: ^14.18.0 || >=16.0.0}
    peerDependencies:
      eslint: '*'
      eslint-plugin-import: '*'
      eslint-plugin-import-x: '*'
    peerDependenciesMeta:
      eslint-plugin-import:
        optional: true
      eslint-plugin-import-x:
        optional: true

  eslint-module-utils@2.14.0:
    resolution: {integrity: sha512-W2WCRZ9Dqntd+2u8jJcVMV2PKulc6RdLgUUoh/yQr3uB6lo/ZOeGx11sv60/8S4QFFKNslAlWhr9u0Ef7ZW6Ig==}
    engines: {node: '>=4'}
    peerDependencies:
      '@typescript-eslint/parser': '*'
      eslint: '*'
      eslint-import-resolver-node: '*'
      eslint-import-resolver-typescript: '*'
      eslint-import-resolver-webpack: '*'
    peerDependenciesMeta:
      '@typescript-eslint/parser':
        optional: true
      eslint:
        optional: true
      eslint-import-resolver-node:
        optional: true
      eslint-import-resolver-typescript:
        optional: true
      eslint-import-resolver-webpack:
        optional: true

  eslint-plugin-import@2.32.0:
    resolution: {integrity: sha512-whOE1HFo/qJDyX4SnXzP4N6zOWn79WhnCUY/iDR0mPfQZO8wcYE4JClzI2oZrhBnnMUCBCHZhO6VQyoBU95mZA==}
    engines: {node: '>=4'}
    peerDependencies:
      '@typescript-eslint/parser': '*'
      eslint: ^2 || ^3 || ^4 || ^5 || ^6 || ^7.2.0 || ^8 || ^9
    peerDependenciesMeta:
      '@typescript-eslint/parser':
        optional: true

  eslint-plugin-jsx-a11y@6.10.2:
    resolution: {integrity: sha512-scB3nz4WmG75pV8+3eRUQOHZlNSUhFNq37xnpgRkCCELU3XMvXAxLk1eqWWyE22Ki4Q01Fnsw9BA3cJHDPgn2Q==}
    engines: {node: '>=4.0'}
    peerDependencies:
      eslint: ^3 || ^4 || ^5 || ^6 || ^7 || ^8 || ^9

  eslint-plugin-react-hooks@5.2.0:
    resolution: {integrity: sha512-+f15FfK64YQwZdJNELETdn5ibXEUQmW1DZL6KXhNnc2heoy/sg9VJJeT7n8TlMWouzWqSWavFkIhHyIbIAEapg==}
    engines: {node: '>=10'}
    peerDependencies:
      eslint: ^3.0.0 || ^4.0.0 || ^5.0.0 || ^6.0.0 || ^7.0.0 || ^8.0.0-0 || ^9.0.0

  eslint-plugin-react@7.37.5:
    resolution: {integrity: sha512-Qteup0SqU15kdocexFNAJMvCJEfa2xUKNV4CC1xsVMrIIqEy3SQ/rqyxCWNzfrd3/ldy6HMlD2e0JDVpDg2qIA==}
    engines: {node: '>=4'}
    peerDependencies:
      eslint: ^3 || ^4 || ^5 || ^6 || ^7 || ^8 || ^9.7

  eslint-scope@8.4.0:
    resolution: {integrity: sha512-sNXOfKCn74rt8RICKMvJS7XKV/Xk9kA7DyJr8mJik3S7Cwgy3qlkkmyS2uQB3jiJg6VNdZd/pDBJu0nvG2NlTg==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}

  eslint-visitor-keys@3.4.3:
    resolution: {integrity: sha512-wpc+LXeiyiisxPlEkUzU6svyS1frIO3Mgxj1fdy7Pm8Ygzguax2N3Fa/D/ag1WqbOprdI+uY6wMUl8/a2G+iag==}
    engines: {node: ^12.22.0 || ^14.17.0 || >=16.0.0}

  eslint-visitor-keys@4.2.1:
    resolution: {integrity: sha512-Uhdk5sfqcee/9H/rCOJikYz67o0a2Tw2hGRPOG2Y1R2dg7brRe1uG0yaNQDHu+TO/uQPF/5eCapvYSmHUjt7JQ==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}

  eslint-visitor-keys@5.0.1:
    resolution: {integrity: sha512-tD40eHxA35h0PEIZNeIjkHoDR4YjjJp34biM0mDvplBe//mB+IHCqHDGV7pxF+7MklTvighcCPPZC7ynWyjdTA==}
    engines: {node: ^20.19.0 || ^22.13.0 || >=24}

  eslint@9.39.5:
    resolution: {integrity: sha512-DgZS62aPLXKlnxILS/AYCoRvHaZeXceIzlXPkkGGzJWSow1aEk0lbTlxUSlyjC8jcaKxAdOnTDz+o1JFSBsyjw==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}
    hasBin: true
    peerDependencies:
      jiti: '*'
    peerDependenciesMeta:
```

```
      jiti:
        optional: true

  espree@10.4.0:
    resolution: {integrity: sha512-j6PAQ2uUr79PZhBjP5C5fhl8e39FmRnOjsD5lGnWrFU8i2G776tBK7+nP8KuQUTTyAZUwfQqXAgrVH5MbH9CYQ==}
    engines: {node: ^18.18.0 || ^20.9.0 || >=21.1.0}

  esquery@1.7.0:
    resolution: {integrity: sha512-Ap6G0WQwcU/LHsvLwON1fAQX9Zp0A2Y6Y/cJBl9r/JbW90Zyg4/zbG6zzKa2OTALELarYHmKu0GhpM5EO+7T0g==}
    engines: {node: '>=0.10'}

  esrecurse@4.3.0:
    resolution: {integrity: sha512-KmfKL3b6G+RXvP8N1vr3Tq1kL/oCFgn2NYXEtqP8/L3pKapUA4G8cFVaoF3SU323CD4XypR/ffioHmkti6/Tag==}
    engines: {node: '>=4.0'}

  estraverse@5.3.0:
    resolution: {integrity: sha512-MMdARuVEQziNTeJD8DgMqmhwR11BRQ/cBP+pLtYdSTnf3MIO8fFeiINEbX36ZdNlfU/7A9f3gUw49B3oQsvwBA==}
    engines: {node: '>=4.0'}

  estree-util-is-identifier-name@3.0.0:
    resolution: {integrity: sha512-hFtqIDZTIUZ9BXLb8y4pYGyk6+wekIivNVTcmvk8NoOh+VeRn5y6cEHzbURrWbfp1fIqdVipilzj+lfaadNZmg==}

  esutils@2.0.3:
    resolution: {integrity: sha512-kVscqXk4OCp68SZ0dkgEKVi6/8ij300KBWTJq32P/dYeWTSwK41WyTxalN1eRmA5Z9UU/LX9D7FWSmV9SAYx6g==}
    engines: {node: '>=0.10.0'}

  extend@3.0.2:
    resolution: {integrity: sha512-fjquC59cD7CyW6urNXK0FBufkZcoiGG80wTuPujX590cB5Ttln20E2UB4S/WARVqhXffZl2LNgS+gQdPIIim/g==}

  fast-deep-equal@3.1.3:
    resolution: {integrity: sha512-f3qQ9oQy9j2AhBe/H9VC91wLmKBCCU/gDOnKNAYG5hswO7BLKj09Hc5HYNz9cGI++xlpDCIgDaitVs03ATR84Q==}

  fast-glob@3.3.1:
    resolution: {integrity: sha512-kNFPyjhh5cKjrUltxs+wFx+ZkbRaxxmZ+X0ZU31SOsxCEtP9VPgtq2teZw1DebupL5GmDaNQ6yKMMVcM41iqDg==}
    engines: {node: '>=8.6.0'}

  fast-json-stable-stringify@2.1.0:
    resolution: {integrity: sha512-lhd/wF+Lk98HZoTCtlVraHtfh5XYijIjalXck7saUtuanSDyLMxnHhSXEDJqHxD7msR8D0uCmqlkwjCV8xvwHw==}

  fast-levenshtein@2.0.6:
    resolution: {integrity: sha512-DCXu6Ifhqcks7TZKY3Hxp3y6qphY5SJZmrWMDrKcERSOXWQdMhU9Ig/PYrzyw/ul9jOIyh0N4M0tbC5hodg8dw==}

  fastq@1.20.1:
    resolution: {integrity: sha512-GGToxJ/w1x32s/D2EKND7kTil4n8OVk/9mycTc4VDza13lOvpUZTGX3mFSCtV9ksdGBVzvsyAVLM6mHFThxXxw==}

  fdir@6.5.0:
    resolution: {integrity: sha512-tIbYtZbucOs0BRGqPJkshJUYdL+SDH7dVM8gjy+ERp3WAUjLEFJE+02kanyHtwjWOnwrKYBiwAmM0p4kLJAnXg==}
    engines: {node: '>=12.0.0'}
    peerDependencies:
      picomatch: ^3 || ^4
    peerDependenciesMeta:
      picomatch:
        optional: true

  file-entry-cache@8.0.0:
    resolution: {integrity: sha512-XXTUwCvisa5oacNGRP9SfNtYBNAMi+RPwBFmblZEF7N7swHYQS6/Zfk7SRwx4D5j3CH211YNRco1DEMNVfZCnQ==}
    engines: {node: '>=16.0.0'}

  fill-range@7.1.1:
    resolution: {integrity: sha512-YsGpe3WHLK8ZYi4tWDg2Jy3ebRz2rXowDxnld4bkQB00cc/1Zw9AWnC0i9ztDJitivtQvaI9KaLyKrc+hBW0yg==}
    engines: {node: '>=8'}

  find-up@5.0.0:
    resolution: {integrity: sha512-78/PXT1wlLLDgTzDs7sjq9hzz0vXD+zn+7wypEe4fXQxCmdmqfGsEPQxmiCSQI3ajFV91bVSsvNtrJRiW6nGng==}
    engines: {node: '>=10'}

  flat-cache@4.0.1:
    resolution: {integrity: sha512-f7ccFPK3SXFHpx15UIGyRJ/FJQctuKZ0zVuN3frBo4HnK3cay9VEW0R6yPYFHC0AgqhukPzKjq22t5DmAyqGyw==}
    engines: {node: '>=16'}

  flatted@3.4.4:
    resolution: {integrity: sha512-5+ybhBZANEJxaH3X5evAFatUxLfEHSr7n6kYJ+1Qd0mUqr4eu9gIf6GDbWHf8RJijHrjjO8G+la14SlL2SeS1Q==}

  for-each@0.3.5:
    resolution: {integrity: sha512-dKx12eRCVIzqCxFGplyFKJMPvLEWgmNtUrpTiJIR5u97zEhRG8ySrtboPHZXx7daLxQVrl643cTzbab2tkQjxg==}
    engines: {node: '>= 0.4'}

  function-bind@1.1.2:
    resolution: {integrity: sha512-7XHNxH7qX9xG5mIwxkhumTox/MIRNcOgDrxWsMt2pAr23WHp6MrRlN7FBSFpCpr+oVO0F744iUgR82nJMfG2SA==}

  function.prototype.name@1.2.0:
    resolution: {integrity: sha512-jObKIik1P2QjPHP5nz5BaOtUlfgS0fWo8IUByNXkM+o+02sJOi94em77GwJKQSJ3gfPHdgzLNrHc1uokV4P/ew==}
    engines: {node: '>= 0.4'}

  functions-have-names@1.2.3:
    resolution: {integrity: sha512-xckBUXyTIqT97tq2x2AMb+g163b5JFysYk0x4qxNFwbfQkmNZoiRHb6sPzI9/QV33WeuvVYBUIiD4NzNIyqaRQ==}

  generator-function@2.0.1:
    resolution: {integrity: sha512-SFdFmIJi+ybC0vjlHN0ZGVGHc3lgE0DxPAT0djjVg+kjOnSqclqmj0KQ7ykTOLP6YxoqOvuAODGdcHJn+43q3g==}
    engines: {node: '>= 0.4'}

  get-intrinsic@1.3.0:
    resolution: {integrity: sha512-9fSjSaos/fRIVIp+xSJlE6lfwhES7LNtKaCBIamHsjr2na1BiABJPo0mOjjz8GJDURarmCPGqaiVg5mfjb98CQ==}
    engines: {node: '>= 0.4'}

  get-nonce@1.0.1:
    resolution: {integrity: sha512-FJhYRoDaiatfEkUK8HKlicmu/3SGFD51q3itKDGoSTysQJBnfOcxU5GxnhE1E6soB76MbT0MBtnKJuXyAx+96Q==}
    engines: {node: '>=6'}

  get-proto@1.0.1:
    resolution: {integrity: sha512-sTSfBjoXBp89JvIKIefqw7U2CCebsc74kiY6awiGogKtoSGbgjYE/G/+l9sF3MWFPNc9IcoOC4ODfKHfxFmp0g==}
    engines: {node: '>= 0.4'}

  get-symbol-description@1.1.0:
    resolution: {integrity: sha512-w9UMqWwJxHNOvoNzSJ2oPF5wvYcvP7jUvYzhp67yEhTi17ZDBBC1z9pTdGuzjD+EFIqLSYRweZjqfiPzQ06Ebg==}
```

```
    engines: {node: '>= 0.4'}

  get-tsconfig@4.14.2:
    resolution: {integrity: sha512-XpwZALwwl/BaKTAyC6+c5T8y6kCg2jk+XGqOVrKIQmW49pNypYLMRjCUXqa28tQgJlhS2RlzP7sc+Rx7W6qsfw==}

  glob-parent@5.1.2:
    resolution: {integrity: sha512-AOIgSQCepiJYwP3ARnGx+5VnTu2HBYdzbGP45eLw1vr3zB3vZLeyed1sC9hnbcOc9/SrMyM5RPQrkGz4aS9Zow==}
    engines: {node: '>= 6'}

  glob-parent@6.0.2:
    resolution: {integrity: sha512-XxwI8EOhVQgWp6iDL+3b0r86f4d6AX6zSU55HfB4ydCEuXLXc5FcYeOu+nnGftS4TEju/11rt4KJPTMgbfmv4A==}
    engines: {node: '>=10.13.0'}

  globals@14.0.0:
    resolution: {integrity: sha512-oahGvuMGQlPw/ivIYBjVSrWAfWLBeku5tpPE2fOPLi+WHffIWbuh2tCjhyQhTBPMf5E9jDEH4FOmTYgYwbKwtQ==}
    engines: {node: '>=18'}

  globalthis@1.0.4:
    resolution: {integrity: sha512-DpLKbNU4WylpxJykQujfCcwYWiV/Jhm50Goo0wrVILAv5jOr9d+H+UR3PhSCD2rCCEIg0uc+G+muBTwD54JhDQ==}
    engines: {node: '>= 0.4'}

  gopd@1.2.0:
    resolution: {integrity: sha512-ZUKRh6/kUFoAiTAtTYPZJ3hw9wNxx+BIBOijnlG9PnrJsCcSjs1wyyD6vJpaYtgnzDrKYRSqf3OO6Rfa93xsRg==}
    engines: {node: '>= 0.4'}

  graceful-fs@4.2.11:
    resolution: {integrity: sha512-RbJ5/jmFcNNCcDV5o9eTnBLJ/HszWV0P73bc+Ff4nS/rJj+YaS6IGyiOL0VoBYX+l1Wrl3k63h/KrH+nhJ0XvQ==}

  has-bigints@1.1.0:
    resolution: {integrity: sha512-R3pbpkcIqv2Pm3dUwgjclDRVmWpTJW2DcMzcIhEXEx1oh/CEMObMm3KLmRJOdvhM7o4uQBnwr8pzRK2sJWIqfg==}
    engines: {node: '>= 0.4'}

  has-flag@4.0.0:
    resolution: {integrity: sha512-EykJT/Q1KjTWctppgIAgfSO0tKVuZUjhgMr17kqTumMl6Afv3EISleU7qZUzoXDFTAHTDC4NOoG/ZxU3EvlMPQ==}
    engines: {node: '>=8'}

  has-property-descriptors@1.0.2:
    resolution: {integrity: sha512-55JNKuIW+vq4Ke1BjOTjM2YctQIvCT7GFzHwmfZPGo5wnrgkid0YQtnAleFSqumZm4az3n2BS+erby5ipJdgrg==}

  has-proto@1.2.0:
    resolution: {integrity: sha512-KIL7eQPfHQRC8+XluaIw7BHUwwqL19bQn4hzNgdr+1wXoU0KKj6rufu47lhY7KbJR2C6T6+PfyN0Ea7wkSS+qQ==}
    engines: {node: '>= 0.4'}

  has-symbols@1.1.0:
    resolution: {integrity: sha512-1cDNdwJ2Jaohmb3sg4OmKaMBwuC48sYni5HUw2DvsC8LjGTLK9h+eb1X6RyuOHe4hT0ULCW68iomhjUoKUqlPQ==}
    engines: {node: '>= 0.4'}

  has-tostringtag@1.0.2:
    resolution: {integrity: sha512-NqADB8VjPFLM2V0VvHUewwwsw0ZWBaIdgo+ieHtK3hasLz4qeCRjYcqfB6AQrBggRKppKF8L52/VqdVsO47Dlw==}
    engines: {node: '>= 0.4'}

  hasown@2.0.4:
    resolution: {integrity: sha512-T2UbfbBEF32wiepXIsMlTW9+dDYC6wMh/t/vYA4tuOMKqWz/n3vr1NFSxQiyP+zk2mXsoMA/i/7qV6LKut1t1A==}
    engines: {node: '>= 0.4'}

  hast-util-from-parse5@8.0.3:
    resolution: {integrity: sha512-3kxEVkEKt0zvcZ3hCRYI8rqrgwtlIOFMWkbclACvjlDw8Li9S2hk/d51OI0nr/gIpdMHNepwgOKqZ/sy0Clpyg==}

  hast-util-parse-selector@4.0.0:
    resolution: {integrity: sha512-wkQCkSYoOGCRKERFWcxMVMOcYE2K1AaNLU8DXS9arxnLOUEWbOXKXiJUNzEpqZ3JOKpnha3jkFrumEjVliDe7A==}

  hast-util-raw@9.1.0:
    resolution: {integrity: sha512-Y8/SBAHkZGoNkpzqqfCldijcuUKh7/su31kEBp67cFY09Wy0mTRgtsLYsiIxMJxlu0f6AA5SUTbDR8K0rxnbUw==}

  hast-util-sanitize@5.0.2:
    resolution: {integrity: sha512-3yTWghByc50aGS7JlGhk61SPenfE/p1oaFeNwkOOyrscaOkMGrcW9+Cy/QAIOBpZxP1yqDIzFMR0+Np0i0+usg==}

  hast-util-to-jsx-runtime@2.3.6:
    resolution: {integrity: sha512-zl6s8LwNyo1P9uw+XJGvZtdFF1GdAkOg8ujOw+4Pyb76874fLps4ueHXDhXWdk6YHQ6OgUtinliG7RsYvCbbBg==}

  hast-util-to-parse5@8.0.1:
    resolution: {integrity: sha512-MlWT6Pjt4CG9lFCjiz4BH7l9wmrMkfkJYCxFwKQic8+RTZgWPuWxwAfjJElsXkex7DJjfSJsQIt931ilUgmwdA==}

  hast-util-whitespace@3.0.0:
    resolution: {integrity: sha512-88JUN06ipLwsnv+dVn+OIYOvAuvBMy/Qoi6O7mQHxdPXpjy+Cd6xRkWwux7DKO+4sYILtLBRIKgsdpS2gQc7qw==}

  hastscript@9.0.1:
    resolution: {integrity: sha512-g7df9rMFX/SPi34tyGCyUBREQoKkapwdY/T04Qn9TDWfHhAYt4/I0gMVirzK5wEzeUqIjEB+LXC/ypb7Aqno5w==}

  html-url-attributes@3.0.1:
    resolution: {integrity: sha512-ol6UPyBWqsrO6EJySPz2O7ZSr856WDrEzM5zMqp+FJJLGMW35cLYmmZnl0vztAZxRUoNZJFTCohfjuIJ8I4QBQ==}

  html-void-elements@3.0.0:
    resolution: {integrity: sha512-bEqo66MRXsUGxWHV5IP0PUiAWwoEjba4VCzg0LjFJBpchPaTfyfCKTG6bc5F8ucKec3q5y6qOdGyYTSBEvhCrg==}

  ignore@5.3.2:
    resolution: {integrity: sha512-hsBTNUqQTDwkWtcdYI2i06Y/nUBEsNEDJKjWdigLvegy8kDuJAS8uRlpkkcQpyEXL0Z/pjDy5HBmMjRCJ2gq+g==}
    engines: {node: '>= 4'}

  ignore@7.0.6:
    resolution: {integrity: sha512-BAg6QkE8W+TuQLrrw0Ugr7HegXduRuuj8/ti2kSOc+jz1dmx8/WNcjr6XGnq5YpDWxFwwaavqD0+jIUOKelTsw==}
    engines: {node: '>= 4'}

  import-fresh@3.3.1:
    resolution: {integrity: sha512-TR3KfrTZTYLPB6jUjfx6MF9WcWrHL9su5TObK4ZkYgBdWKPOFoSoQIdEuTuR82pmtxH2spWG9h6etwfr1pLBqQ==}
    engines: {node: '>=6'}

  imurmurhash@0.1.4:
    resolution: {integrity: sha512-JmXMZ6wuvDmLiHEml9ykzqO6lwFbof0GG4IkcGaENdCRDDmMVnny7s5HsIgHCbaq0w2MyPhDqkhTUgS2LU2PHA==}
    engines: {node: '>=0.8.19'}

  inline-style-parser@0.2.7:
    resolution: {integrity: sha512-Nb2ctOyNR8DqQoR0OwRG95uNWIC0C1lCgf5Naz5H6Ji72KZ8OcFZLz2P5sNgwlyoJ8Yif11oMuYs5pBQa86csA==}

  internal-slot@1.1.0:
```

```
    resolution: {integrity: sha512-4gd7VpWNQNB4UKKCFFVcp1AVv+FMOgs9NKzjHKusc8jTMhd5eL1NqQqOpE0KzMds804/yHlglp3uxgluOqAPLw==}
    engines: {node: '>= 0.4'}

  is-alphabetical@2.0.1:
    resolution: {integrity: sha512-FWyyY60MeTNyeSRpkM2Iry0G9hpr7/9kD40mD/cGQEuilcZYS4okz8SN2Q6rLCJ8gbCt6fN+rC+6tMGS99LaxQ==}

  is-alphanumerical@2.0.1:
    resolution: {integrity: sha512-hmbYhX/9MUMF5uh7tOXyK/n0ZvWpad5caBA17GsC6vyuCqaWliRG5K1qS9inmUhEMaOBIW7/whAnSwveW/LtZw==}

  is-array-buffer@3.0.5:
    resolution: {integrity: sha512-DDfANUiiG2wC1qawP66qlTugJeL5HyzMpfr8lLK+jMQirGzNod0B12cFB/9q838Ru27sBwfw78/rdoU7RERz6A==}
    engines: {node: '>= 0.4'}

  is-async-function@2.1.1:
    resolution: {integrity: sha512-9dgM/cZBnNvjzaMYHVoxxfPj2QXt22Ev7SuuPrs+xav0ukGB0S6d4ydZdEiM48kLx5kDV+QBPrpVnFyefL8kkQ==}
    engines: {node: '>= 0.4'}

  is-bigint@1.1.0:
    resolution: {integrity: sha512-n4ZT37wG78iz03xPRKJrHTdZbe3IicyucEtdRsV5yglwc3GyUfbAfpSeD0FJ41NbUNSt5wbhqfp1fS+BgnvDFQ==}
    engines: {node: '>= 0.4'}

  is-boolean-object@1.2.2:
    resolution: {integrity: sha512-wa56o2/ElJMYqjCjGkXri7it5FbebW5usLw/nPmCMs5DeZ7eziSYZhSmPRn0txqeW4LnAmQQU7FgqLpsEFKM4A==}
    engines: {node: '>= 0.4'}

  is-bun-module@2.0.0:
    resolution: {integrity: sha512-gNCGbnnnnFAUGKeZ9PdbyeGYJqewpmc2aKHUEMO5nQPWU9lOmv7jcmQIv+qHD8fXW6W7qfuCwX4rY9LNRjXrkQ==}

  is-callable@1.2.7:
    resolution: {integrity: sha512-1BC0BVFhS/p0qtw6enp8e+8OD0UrK0oFLztSjNzhcKA3WDuJxxAPXzPuPtKkjEY9UUoEWlX/8fgKeu2S8i9JTA==}
    engines: {node: '>= 0.4'}

  is-core-module@2.16.2:
    resolution: {integrity: sha512-evOr8xfXKxE6qSR0hSXL2r3sd7ALj8+7jQEUvPYcm5sgZFdJ+AYzT6yNmJenvIYQBgIGwfwz08sL8zoL7yq2BA==}
    engines: {node: '>= 0.4'}

  is-data-view@1.0.2:
    resolution: {integrity: sha512-RKtWF8pGmS87i2D6gqQu/l7EYRlVdfzemCJN/P3UOs//x1QE7mfhvzHIApBTRf7axvT6DMGwSwBXYCT0nfB9xw==}
    engines: {node: '>= 0.4'}

  is-date-object@1.1.0:
    resolution: {integrity: sha512-PwwhEakHVKTdRNVOw+/Gyh0+MzlCl4R6qKvkhuvLtPMggI1WAHt9sOwZxQLSGpUaDnrdyDsomoRgNnCfKNSXXg==}
    engines: {node: '>= 0.4'}

  is-decimal@2.0.1:
    resolution: {integrity: sha512-AAB9hiomQs5DXWcRB1rqsxGUstbRroFOPPVAomNk/3XHR5JyEZChOyTWe2oayKnsSsr/kcGqF+z6yuH6HHpN0A==}

  is-document.all@1.0.0:
    resolution: {integrity: sha512-+XSoyS05OdBbhFuELhgTCpFNHkpBOJqtsZfUFFpe5QTw+9Sjbh8zitxhQkYAo6wV7e1Vb8cAPvpCk9jGam/82g==}
    engines: {node: '>= 0.4'}

  is-extglob@2.1.1:
    resolution: {integrity: sha512-SbKbANkN603Vi4jEZv49LeVJMn4yGwsbzZworEoyEiutsN3nJYdbO36zfhGJ6QEDpOZIFkDtnq5JRxmvl3jsoQ==}
    engines: {node: '>=0.10.0'}

  is-finalizationregistry@1.1.1:
    resolution: {integrity: sha512-1pC6N8qWJbWoPtEjgcL2xyhQOP491EQjeUo3qTKcmV8YSDDJrOepfG8pcC7h/QgnQHYSv0mJ3Z/ZWxmatVrysg==}
    engines: {node: '>= 0.4'}

  is-generator-function@1.1.2:
    resolution: {integrity: sha512-upqt1SkGkODW9tsGNG5mtXTXtECizwtS2kA161M+gJPc1xdb/Ax629af6YrTwcOeQHbewrPNlE5Dx7kzvXTizA==}
    engines: {node: '>= 0.4'}

  is-glob@4.0.3:
    resolution: {integrity: sha512-xelSayHH36ZgE7ZWhli7pW34hNbNl8Ojv5KVmkJD4hBdD3th8Tfk9vYasLM+mXWOZhFkgZfxhLSnrwRr4elSSg==}
    engines: {node: '>=0.10.0'}

  is-hexadecimal@2.0.1:
    resolution: {integrity: sha512-DgZQp241c8oO6cA1SbTEWiXeoxV42vlcJxgH+B3hi1AiqqKruZR3ZGF8In3fj4+/y/7rHvlOZLZtgJ/4ttYGZg==}

  is-map@2.0.3:
    resolution: {integrity: sha512-1Qed0/Hr2m+YqxnM09CjA2d/i6YZNfF6R2oRAOj36eUdS6qIV/huPJNSEpKbupewFs+ZsJlxsjjPbc0/afW6Lw==}
    engines: {node: '>= 0.4'}

  is-negative-zero@2.0.3:
    resolution: {integrity: sha512-5KoIu2Ngpyek75jXodFvnafB6DJgr3u8uuK0LEZJjrU19DrMD3EVERaR8sjz8CCGgpZvxPl9SuE1GMVPFHx1mw==}
    engines: {node: '>= 0.4'}

  is-number-object@1.1.1:
    resolution: {integrity: sha512-lZhclumE1G6VYD8VHe35wFaIif+CTy5SJIi5+3y4psDgWu4wPDoBhF8NxUOinEc7pHgiTsT6MaBb92rKhhD+Xw==}
    engines: {node: '>= 0.4'}

  is-number@7.0.0:
    resolution: {integrity: sha512-41Cifkg6e8TylSpdTpeLVMqvSBEVzTttHvERD741+pnZ8ANv0004MRL43QKPDlK9cGvNp6NZWZUBlbGXYxxng==}
    engines: {node: '>=0.12.0'}

  is-plain-obj@4.1.0:
    resolution: {integrity: sha512-+Pgi+vMuUNkJyExiMBt5IlFoMyKnr5zhJ4Uspz58WOhBF5QoIZkFyNHIbBAtHwzVAgk5RtndVNsDRN61/mmDqg==}
    engines: {node: '>=12'}

  is-regex@1.2.1:
    resolution: {integrity: sha512-MjYsKHO5O7mCsmRGxWcLWheFqN9DJ/2TmngvjKXihe6efViPqc274+Fx/4fYj/r03+ESvBdTXK0V6tA3rgez1g==}
    engines: {node: '>= 0.4'}

  is-set@2.0.3:
    resolution: {integrity: sha512-iPAjerrse27/ygGLxw+EBR9agv9Y6uLeYVJMu+QNCoouJ1/1ri0mGrcWpfCqFZuzzx3WjtwxG098X+n4OuRkPg==}
    engines: {node: '>= 0.4'}

  is-shared-array-buffer@1.0.4:
    resolution: {integrity: sha512-ISWac8drv4ZGfwKl5slpHG9OwPNty4jOWPRIhBpxOoD+hqITiwuipOQ2bNthAzwA3B4fIjO4Nln74N0S9byq8A==}
    engines: {node: '>= 0.4'}

  is-string@1.1.1:
    resolution: {integrity: sha512-BtEeSsoaQjlSPBemMQIrY1MY0uM6vnS1g5fmufYOtnxLGUZM2178PKbhsk7Ffv58IX+ZtcvoGwccYsh0PglkAA==}
    engines: {node: '>= 0.4'}
```

```yaml

  is-symbol@1.1.1:
    resolution: {integrity: sha512-9gGx6GTtCQM73BgmHQXfDmLtfjjTUDSyoxTCbp5WtoixAhfgsDirWIcVQ/IHpvI5Vgd5i/J5F7B9cN/WlVbC/w==}
    engines: {node: '>= 0.4'}

  is-typed-array@1.1.15:
    resolution: {integrity: sha512-p3EcsicXjit7SaskXHs1hA91QxgTw46Fv6EFKKGS5DRFLD8yKnohjF3hxoju94b/OcMZoQukzpPpBE9uLVKzgQ==}
    engines: {node: '>= 0.4'}

  is-weakmap@2.0.2:
    resolution: {integrity: sha512-K5pXYOm9wqY1RgjpL3YTkF39tni1XajUIkawTLUo9EZEVUFga5gSQJF8nNS7ZwJQ02y+1YCNYcMh+HIf1ZqE+w==}
    engines: {node: '>= 0.4'}

  is-weakref@1.1.1:
    resolution: {integrity: sha512-6i9mGWSlqzNMEqpCp93KwRS1uUOodk2OJ6b+sq7ZPDSy2WuI5NFIxp/254TytR8ftefexkWn5xNiHUNpPOfSew==}
    engines: {node: '>= 0.4'}

  is-weakset@2.0.4:
    resolution: {integrity: sha512-mfcwb6IzQyOKTs84CQMrOwW4gQcaTOAWJ0zzJCl2WSPDrWk/OzDaImWFH3djXhb24g4eudZfLRozAvPGw4d9hQ==}
    engines: {node: '>= 0.4'}

  isarray@2.0.5:
    resolution: {integrity: sha512-xHjhDr3cNBK0BzdUJSPXZntQUx/mwMS5Rw4A7lPJ90XGAO6ISP/ePDNuo0vhqOZU+UD5JoodwCAAoZQd3FeAKw==}

  isexe@2.0.0:
    resolution: {integrity: sha512-RHxMLp9lnKHGHRng9QFhRCMbYAcVpn69smSGcq3f36xjgVVWThj4qqLbTLlq7Ssj8B+fIQ1EuCEGI2lKsyQeIw==}

  iterator.prototype@1.1.5:
    resolution: {integrity: sha512-H0dkQoCa3b2VEeKQBOxFph+JAbcrQdE7KC0UkqwpLmv2EC4P41QXP+rqo9wYodACiG5/WM5s9oDApTU8utwj9g==}
    engines: {node: '>= 0.4'}

  jiti@2.7.0:
    resolution: {integrity: sha512-AC/7JofJvZGrrneWNaEnJeOLUx+JlGt7tNa0wZiRPT4MY1wmfKjt2+6O2p2uz2+skll8OZZmJMNqeke7kKbNgQ==}
    hasBin: true

  js-tokens@4.0.0:
    resolution: {integrity: sha512-RdJUflcE3cUzKiMqQgsCu06FPu9UdIJO0beYbPhHN4k6apgJtifcoCtT9bcxOpYBtpD2kCM6Sbzg4CausW/PKQ==}

  js-yaml@4.3.1:
    resolution: {integrity: sha512-CY6crGq313MX8GkwvB7tzgp99vjQxY1++5y10/BKN/GUfHqWaOGQMNZkBvqSzsZKWk/ijwHlWzzkLulsGHhjWQ==}
    hasBin: true

  json-buffer@3.0.1:
    resolution: {integrity: sha512-4bV5BfR2mqfQTJm+V5tPPdf+ZpuhiIvTuAB5g8kcrXOZpTT/QwwVRWBywX1ozr6lEuPdbHxwaJlm9G6mI2sfSQ==}

  json-schema-traverse@0.4.1:
    resolution: {integrity: sha512-xbbCH5dCYU5T8LcEhhuh7HJ88HXuW3qsI3Y0zOZFKfZEHcpWiHU/Jxzk629Brsab/mMiHQti9wMP+845RPe3Vg==}

  json-stable-stringify-without-jsonify@1.0.1:
    resolution: {integrity: sha512-Bdboy+l7tA3OGW6FjyFHWkP5LuByj1Tk33Ljyq0axyzdk9//JSi2u3fP1QSmd1KNwq6VOKYGlAu87CisVir6Pw==}

  json5@1.0.2:
    resolution: {integrity: sha512-g1MWMLBiz8FKi1e4w0UyVL3w+iJceWAFBAaBnnGKOpNa5f8TLktkbre1+s6oICydWAm+HRUGTmI+//xv2hvXYA==}
    hasBin: true

  jsx-ast-utils@3.3.5:
    resolution: {integrity: sha512-ZZow9HBI5O6EPgSJLUb8n2NKgmVWTwCvHGwFuJlMjvLFqlGG6pjirPhtdsseaLZjSibD8eegzmYpUZwoIlj2cQ==}
    engines: {node: '>=4.0'}

  keyv@4.5.4:
    resolution: {integrity: sha512-oxVHkHR/EJf2CNXnWxRLW6mg7JyCCUcG0DtEGmL2ctUo1PNTin1PUil+r/+4r5MpVgC/fn1kjsx7mjSujKqIpw==}

  language-subtag-registry@0.3.23:
    resolution: {integrity: sha512-0K65Lea881pHotoGEa5gDlMxt3pctLi2RplBb7Ezh4rRdLEOtgi7n4EwK9lamnUCkKBqaeKRVebTq6BAxSkpXQ==}

  language-tags@1.0.9:
    resolution: {integrity: sha512-MbjN408fEndfiQXbFQ1vnd+1NoLDsnQW41410oQBXiyXDMYH5z505juWa4KUE1LqxRC7DgOgZDbKLxHIwm27hA==}
    engines: {node: '>=0.10'}

  levn@0.4.1:
    resolution: {integrity: sha512-+bT2uH4E5LGE7h/n3evcS/sQlJXCpIp6ym8OWJ5eV6+67Dsql/LaaT7qJBAt2rzfoa/5QBGBhxDix1dMt2kQKQ==}
    engines: {node: '>= 0.8.0'}

  lightningcss-android-arm64@1.32.0:
    resolution: {integrity: sha512-YK7/ClTt4kAK0vo6w3X+Pnm0D2cf2vPHbhOXdoNti1Ga0al1P4TBZhwjATvjNwLEBCnKvjJc2jQgHXH0NEwlAg==}
    engines: {node: '>= 12.0.0'}
    cpu: [arm64]
    os: [android]

  lightningcss-darwin-arm64@1.32.0:
    resolution: {integrity: sha512-RzeG9Ju5bag2Bv1/lwlVJvBE3q6TtXskdZLLCyfg5pt+HLz9BqlICO7LZM7VHNTTn/5PRhHFBSjk5lc4cmscPQ==}
    engines: {node: '>= 12.0.0'}
    cpu: [arm64]
    os: [darwin]

  lightningcss-darwin-x64@1.32.0:
    resolution: {integrity: sha512-U+QsBp2m/s2wqpUYT/6wnlagdZbtZdndSmut/NJqlCcMLTWp5muCrID+K5UJ6jqD2BFshejCYXniPDbNh73V8w==}
    engines: {node: '>= 12.0.0'}
    cpu: [x64]
    os: [darwin]

  lightningcss-freebsd-x64@1.32.0:
    resolution: {integrity: sha512-JCTigedEksZk3tHTTthnMdVfGf61Fky8Ji2E4YjUTEQX14xiy/lTzXnu1vwiZe3bYe0q+SpsSH/CTeDXK6WHig==}
    engines: {node: '>= 12.0.0'}
    cpu: [x64]
    os: [freebsd]

  lightningcss-linux-arm-gnueabihf@1.32.0:
    resolution: {integrity: sha512-x6rnnpRa2GL0zQOkt6rts3YDPzduLpWvwAF6EMhXFVZXD4tPrBkEFqzGowzCsIWsPjqSK+tyNEODUBXeeVHSkw==}
    engines: {node: '>= 12.0.0'}
    cpu: [arm]
    os: [linux]

  lightningcss-linux-arm64-gnu@1.32.0:
    resolution: {integrity: sha512-0nnMyoyOLRJXfbMOilaSRcLH3Jw5z9HDNGfT/gwCPgaDjnx0i8w7vBzFLFR1f6CMLKF8gVbebmkUN3fa/kQJpQ==}
```

```
    engines: {node: '>= 12.0.0'}
    cpu: [arm64]
    os: [linux]

  lightningcss-linux-arm64-musl@1.32.0:
    resolution: {integrity: sha512-UpQkoenr4UJEzgVIYpI80lDFvRmPVg6oqboNHfoH4CQIfNA+HOrZ7Mo7KZP02dC6LjghPQJeBsvXhJod/wnIBg==}
    engines: {node: '>= 12.0.0'}
    cpu: [arm64]
    os: [linux]

  lightningcss-linux-x64-gnu@1.32.0:
    resolution: {integrity: sha512-V7Qr52IhZmdKPVr+Vtw8o+WLsQJYCTd8loIfpDaMRWGUZfBOYEJeyJIkqGIDMZPwPx24pUMfwSxxI8phr/MbOA==}
    engines: {node: '>= 12.0.0'}
    cpu: [x64]
    os: [linux]

  lightningcss-linux-x64-musl@1.32.0:
    resolution: {integrity: sha512-bYcLp+Vb0awsiXg/80uCRezCYHNg1/l3mt0gzHnWV9XP1W5sKa5/TCdGWaR/zBM2PeF/HbsQv/j2URNOiVuxWg==}
    engines: {node: '>= 12.0.0'}
    cpu: [x64]
    os: [linux]

  lightningcss-win32-arm64-msvc@1.32.0:
    resolution: {integrity: sha512-8SbC8BR40pS6baCM8sbtYDSwEVQd4JlFTOlaD3gWGHfThTcABnNDBda6eTZeqbofalIJhFx0qKzgHJmcPTnGdw==}
    engines: {node: '>= 12.0.0'}
    cpu: [arm64]
    os: [win32]

  lightningcss-win32-x64-msvc@1.32.0:
    resolution: {integrity: sha512-Amq9B/SoZYdDi1kFrojnoqPLxYhQ4Wo5XiL8EVJrVsB8ARoC1PWW6VGtT0WKCemjy8aC+louJnjS7U18x3b06Q==}
    engines: {node: '>= 12.0.0'}
    cpu: [x64]
    os: [win32]

  lightningcss@1.32.0:
    resolution: {integrity: sha512-NXYBzinNrblfraPGyrbPoD19C1h9lfI/1mzgWYvXUTe414Gz/X1FD2XBZSZM7rRTrMA8JL3OtAaGifrIKhQ5yQ==}
    engines: {node: '>= 12.0.0'}

  locate-path@6.0.0:
    resolution: {integrity: sha512-iPZK6eYjbxRu3uB4/WZ3EsEIMJFMqAoopl3R+zuq0UjcAm/MO6KCweDgPfP3elTztoKP3KtnVHxTn2NHBSDVUw==}
    engines: {node: '>=10'}

  lodash.merge@4.6.2:
    resolution: {integrity: sha512-0KpjqXRVvrYyCsX1swR/XTK0va6VQkQM6MNo7PqW77ByjAhoARA8EfrP1N4+KlKj8YS0ZUCtRT/YUuhyYDujIQ==}

  longest-streak@3.1.0:
    resolution: {integrity: sha512-9Ri+o0JYgehTaVBBDoMqIl8GXtbWg711O3srftcHhZ0dqnETqLaoIK0x17fUw9rFSlK/0NlsKe0Ahhyl5pXE2g==}

  loose-envify@1.4.0:
    resolution: {integrity: sha512-lyuxPGr/Wfhrlem2CL/UcnUc1zcqKAImBDzukY7Y5F/yQiNdko6+fRLevlw1HgMySw7f611UIY408EtxRSoK3Q==}
    hasBin: true

  lucide-react@0.468.0:
    resolution: {integrity: sha512-6koYRhnM2N0GGZIdXzSeiNwguv1gt/FAjZOiPl76roBi3xKEXa4WmfpxgQwTTL4KipXjefrnf3oV4IsYhi4JFA==}
    peerDependencies:
      react: ^16.5.1 || ^17.0.0 || ^18.0.0 || ^19.0.0-rc

  magic-string@0.30.21:
    resolution: {integrity: sha512-vd2F4YUyEXKGcLHoq+TEyCjxueSeHnFxyyjNp80yg0XV4vUhnDer/lvvlqM/arB5bXQN5K2/3oinyCRyx8T2CQ==}

  markdown-table@3.0.4:
    resolution: {integrity: sha512-wiYz4+JrLyb/DqW2hkFJxP7Vd7JuTDm77fvbM8VfEQdmSMqcImWeeRbHwZjBjIFki/VaMK2BhFi7oUUZeM5bqw==}

  math-intrinsics@1.1.0:
    resolution: {integrity: sha512-/IXtbwEk5HTPyEwyKX6hGkYXxM9nbj64B+ilVJnC/R6B0pH5G4V3b0pVbL7DBj4tkhBAppbQUlf6F6Xl9LHu1g==}
    engines: {node: '>= 0.4'}

  mdast-util-find-and-replace@3.0.2:
    resolution: {integrity: sha512-Tmd1Vg/m3Xz43afeNxDIhWRtFZgM2VLyaf4vSTYwudTyeuTneoL3qtWMA5jeLyz/O1vDJmmV4QuScFCA2tBPwg==}

  mdast-util-from-markdown@2.0.3:
    resolution: {integrity: sha512-W4mAWTvSlKvf8L6J+VN9yLSqQ9AOAAvHuoDAmPkz4dHf553m5gVj2ejadHJhoJmcmxEnOv6Pa8XJhpxE93kb8Q==}

  mdast-util-gfm-autolink-literal@2.0.1:
    resolution: {integrity: sha512-5HVP2MKaP6L+G6YaxPNjuL0BPrq9orG3TsrZ9YXbA3vDw/ACI4MEsnoDpn6ZNm7GnZgtAcONJyPhOP8tNJQavQ==}

  mdast-util-gfm-footnote@2.1.0:
    resolution: {integrity: sha512-sqpDWlsHn7Ac9GNZQMeUzPQSMzR6Wv0WKRNvQRg0KqHh02fpTz69Qc1QSseNX29bhz1ROIyNyxExfawVKTm1GQ==}

  mdast-util-gfm-strikethrough@2.0.0:
    resolution: {integrity: sha512-mKKb915TF+OC5ptj5bJ7WFRPdYtuHv0yTRxK2tJvi+BDqbkiG7h7u/9SI89nRAYcmap2xHQL9D+QG/6wSrTtXg==}

  mdast-util-gfm-table@2.0.0:
    resolution: {integrity: sha512-78UEvebzz/rJIxLvE7ZtDd/vIQ0RHv+3Mh5DR96p7cS7HsBhYIICDBCu8csTNWNO6tBWfqXPWekRuj2FNOGOZg==}

  mdast-util-gfm-task-list-item@2.0.0:
    resolution: {integrity: sha512-IrtvNvjxC1o06taBAVJznEnkiHxLFTzgonUdy8hzFVeDun0uTjxxrRGVaNFqkU1wJR3RBPEfsxmU6jDWPofrTQ==}

  mdast-util-gfm@3.1.0:
    resolution: {integrity: sha512-0ulfdQOM3ysHhCJ1p06l0b0VKlhU0wuQs3thxZQagjcjPrlFRqY215uZGHHJan9GEAXd9MbfPjFJz+qMkVR6zQ==}

  mdast-util-mdx-expression@2.0.1:
    resolution: {integrity: sha512-J6f+9hUp+ldTZqKRSg7Vw5V6MqjATc+3E4gf3CFNcuZNWD8XdyI6zQ8GqH7f8169MM6P7hMBRDVGnn7oHB9kXQ==}

  mdast-util-mdx-jsx@3.2.0:
    resolution: {integrity: sha512-lj/z8v0r6ZtsN/cGNNtemmmfoLAFZnjMbNyLzBafjzikOM+glrjNHPlf6lQDOTccj9n5b0PPihEBbhneMyGs1Q==}

  mdast-util-mdxjs-esm@2.0.1:
    resolution: {integrity: sha512-EcmOpxsZ96CvlP03NghtH1EsLtr0n9Tm4lPUJUBccV9RwUOneqSycg19n5HGzCf+10LozMRSObtVr3ee1WoHtg==}

  mdast-util-phrasing@4.1.0:
    resolution: {integrity: sha512-TqICwyvJJpBwvGAMZjj4J2n0X8QWp21b9l0o7eXyVJ25YNWYbJDVIyD1bZXE6WtV6RmKJVYmQAKWa0zWOABz2w==}

  mdast-util-to-hast@13.2.1:
    resolution: {integrity: sha512-cctsq2wp5vTsLIcaymblUriiTcZd0CwWtCbLvrOzYCDZoWyMNV8sZ7krj09FSnsiJi3WVsHLM4k6Dq/yaPyCXA==}
```

```
    mdast-util-to-markdown@2.1.2:
      resolution: {integrity: sha512-xj68wMTvGXVOKonmog6LwyJKrYXZPvlwabaryTjLh9LuvovB/KAH+kvi8Gjj+7rJjsFi23nkUxRQv1KqSroMqA==}

    mdast-util-to-string@4.0.0:
      resolution: {integrity: sha512-0H44vDimn51F0YwvxSJSm0eCDOJTRlmN0R1yBh4HLj9wiV1Dn0QoXGbvFAWj2hSItVTlCmBF1hqKlIyUBVFLPg==}

    merge2@1.4.1:
      resolution: {integrity: sha512-8q7VEgMJW4J8tcfVPy8g09NcQwZdbwFEqhe/WZkoIzjn/3TGDwtOCYtXGxA3O8tPzpczCCDgv+P2P5y00ZJOOg==}
      engines: {node: '>= 8'}

    micromark-core-commonmark@2.0.3:
      resolution: {integrity: sha512-RDBrHEMSxVFLg6xvnXmb1Ayr2WzLAWjeSATAoxwKYJV94TeNavgoIdA0a9ytzDSVzBy2YKFK+emCPOEibLeCrg==}

    micromark-extension-gfm-autolink-literal@2.1.0:
      resolution: {integrity: sha512-oOg7knzhicgQ3t4QCjCWgTmfNhvQbDDnJeVu9v81r7NltNCVmhPy1fJRX27pISafdjL+SVc4d3l48Gb6pbRypw==}

    micromark-extension-gfm-footnote@2.1.0:
      resolution: {integrity: sha512-/yPhxI1ntnDNsiHtzLKYnE3vf9JZ6cAisqVDauhp4CEHxlb4uoOTxOCJ+9s51bIB8U1N1FJ1RXOKTIlD5B/gqw==}

    micromark-extension-gfm-strikethrough@2.1.0:
      resolution: {integrity: sha512-ADVjpOOkjz1hhkZLlBiYA9cR2Anf8F4HqZUO6e5eDcPQd0Txw5fxLzzxnEkSkfnD0wziSGiv7sYhk/ktvbf1uw==}

    micromark-extension-gfm-table@2.1.1:
      resolution: {integrity: sha512-t2OU/dXXioARrC6yWfJ4hqB7rct14e8f7m0cbI5hUmDyyIlwv5vEtooptH8INkbLzOatzKuVbQmAYcbWoyz6Dg==}

    micromark-extension-gfm-tagfilter@2.0.0:
      resolution: {integrity: sha512-xHlTOmuCSotIA8TW1mDIM6X2O1SiX5P9IuDtqGonFhEK0qgRI4yeC6vMxEV2dgyr2TiD+2PQ10o+cOhdVAcwfg==}

    micromark-extension-gfm-task-list-item@2.1.0:
      resolution: {integrity: sha512-qIBZhqxqI6fjLDYFTBIa4eivDMnP+OZqsNwmQ3xNLE4Cxwc+zfQEfbs6tzAo2Hjq+bh6q5F+Z8/cksrLFYWQQw==}

    micromark-extension-gfm@3.0.0:
      resolution: {integrity: sha512-vsKArQsicm7t0z2GugkCKtZehqUm31oeGBV/KVSorWSy8ZlNAv7ytjFhvaryUiCUJYqs+NoE6AFhpQvBTM6Q4w==}

    micromark-factory-destination@2.0.1:
      resolution: {integrity: sha512-Xe6rDdJlkmbFRExpTOmRj9N3MaWmbAgdpSrBQvCFqhezUn4AHqJHbaEnfbVYYiexVSs//tqOdY/DxhjdCiJnIA==}

    micromark-factory-label@2.0.1:
      resolution: {integrity: sha512-VFMekyQExqIW7xIChcXn4ok29YE3rnuyveW3wZQWWqF4Nv9Wk5rgJ99KzPvHjkmPXF93FXIbBp6YdW3t71/7Vg==}

    micromark-factory-space@2.0.1:
      resolution: {integrity: sha512-zRkxjtBxxLd2Sc0d+fbnEunsTj46SWXgXciZmHq0kDYGnck/ZSGj9/wULTV95uoeYiK5hRXP2mJ98Uo4cq/LQg==}

    micromark-factory-title@2.0.1:
      resolution: {integrity: sha512-5bZ+3CjhAd9eChYTHsjy6TGxpOFSKgKKJPJxr293jTbfry2KDoWkhBb6TcPVB4NmzaPhMs1Frm9AZH7OD4Cjzw==}

    micromark-factory-whitespace@2.0.1:
      resolution: {integrity: sha512-Ob0nuZ3PKt/n0hORHyvoD9uZhr+Za8sFoP+OnMcnWK5lngSzALgQYKMr9RJVOWLqQYuyn6ulqGWSXdwf6F80lQ==}

    micromark-util-character@2.1.1:
      resolution: {integrity: sha512-wv8tdUTJ3thSFFFJKtpYKOYiGP2+v96Hvk4Tu8KpCAsTMs6yi+nVmGh1syvSCsaxz45J6Jbw+9DD6g97+NV67Q==}

    micromark-util-chunked@2.0.1:
      resolution: {integrity: sha512-QUNFEOPELfmvv+4xiNg2sRYeS/P84pTW0TCgP5zc9FpXetHY0ab7SxKyAQCNCc1eK0459uoLI1y5oO5Vc1dbhA==}

    micromark-util-classify-character@2.0.1:
      resolution: {integrity: sha512-K0kHzM6afW/MbeWYWLjoHQv1sgg2Q9EccHEDzSkxiP/EaagNzCm7T/WMKZ3rjMbvIpvBiZgwR3dKMygtA4mG1Q==}

    micromark-util-combine-extensions@2.0.1:
      resolution: {integrity: sha512-OnAnH8Ujmy59JcyZw8JSbK9cGpdVY44NKgSM7E9Eh7DiLS2E9RNQf0dONaGDzEG9yjEl5hcqeIsj4hfRkLH/Bg==}

    micromark-util-decode-numeric-character-reference@2.0.2:
      resolution: {integrity: sha512-ccUbYk6CwVdkmCQMyr64dXz42EfHGkPQlBj5p7YVGzq8I7CtjXZJrubAYezf7Rp+bjPseiROqe7G6foFd+lEuw==}

    micromark-util-decode-string@2.0.1:
      resolution: {integrity: sha512-nDV/77Fj6eH1ynwscYTOsbK7rR//Uj0bZXBwJZRfaLEJ1iGBR6kIfNmlNqaqJf649EP0F3NWNdeJi03elllNUQ==}

    micromark-util-encode@2.0.1:
      resolution: {integrity: sha512-c3cVx2y4KqUnwopcO9b/SCdo2O67LwJJ/UyqGfbigahfegL9myoEFoDYZgkT7f36T0bLrM9hZTAaAyH+PCAXjw==}

    micromark-util-html-tag-name@2.0.1:
      resolution: {integrity: sha512-2cNEiYDhCWKI+Gs9T0Tiysk136SnR13hhO8yW6BGNyhOC4qYFnwF1nKfD3HFAIXA5c45RrIG1ub11GiXeYd1xA==}

    micromark-util-normalize-identifier@2.0.1:
      resolution: {integrity: sha512-sxPqmo70LyARJs0w2UclACPUUEqltCkJ6PhKdMIDuJ3gSf/Q+/GIe3WKl0Ijb/GyH9lOpUkRAO2wp0GVkLvS9Q==}

    micromark-util-resolve-all@2.0.1:
      resolution: {integrity: sha512-VdQyxFWFT2/FGJgwQnJYbe1jjQoNTS4RjglmSjTUlpUMa95Htx9NHeYW4rGDJzbjvCsl9eLjMQwGeElsqmzcHg==}

    micromark-util-sanitize-uri@2.0.1:
      resolution: {integrity: sha512-9N9IomZ/YuGGZZmQec1MbgxtlgougxTodVwDzzEouPKo3qFWvymFHWcnDi2vzV1ff6kas9ucW+o3yzJK9YB1AQ==}

    micromark-util-subtokenize@2.1.0:
      resolution: {integrity: sha512-XQLu552iSctvnEcgXw6+Sx75GflAPNED1qx7eBJ+wydBb2KCbRZe+NwvIEEMM83uml1+2WSXpBAcp9IUCgCYWA==}

    micromark-util-symbol@2.0.1:
      resolution: {integrity: sha512-vs5t8Apaud9N28kgCrRUdEed4UJ+wWNvicHLPxCa9ENlYuAY31M0ETy5y1vA33YoNPDFTghEbnh6efaE8h4x0Q==}

    micromark-util-types@2.0.2:
      resolution: {integrity: sha512-Yw0ECSpJoViF1qTU4DC6NwtC4aWGt1EkzaQB8KPPyCRR8z9TWeV0HbEFGTO+ZY1wB22zmxnJqhPyTpOVCpeHTA==}

    micromark@4.0.2:
      resolution: {integrity: sha512-zpe98Q6kvavpCr1NPVSCMebCKfD7CA2NqZ+rykeNhONIJBpc1tFKt9hucLGwha3jNTNI8lHpctWJWoimVF4PfA==}

    micromatch@4.0.8:
      resolution: {integrity: sha512-PXwfBhYu0hBCPw8Dn0E+WDYb7af3dSLVWKi3HGv84IdF4TyFoC0ysxFd0Goxw7nSv4T/PzEJQxsYsEiFCKo2BA==}
      engines: {node: '>=8.6'}

    minimatch@10.2.6:
      resolution: {integrity: sha512-vpLQEs+VLCr1nU0BXS07maYoFwlDAH0gngQuuttxIwutDFEMHq2blX+8vpgxDdK3J1PwjCJiep77OitTZ4Ll1A==}
      engines: {node: 18 || 20 || >=22}

    minimatch@3.1.5:
      resolution: {integrity: sha512-VgjWUsnnT6n+NUk6eZq77zeFdpW2LWDzP6zFGrCbHXiYNul5Dzqk2HHQ5uFH2DNW5Xbp8+jVzaeNt94ssEEl4w==}
```

```yaml

  minimist@1.2.8:
    resolution: {integrity: sha512-2yyAR8qBkN3YuheJanUpWC5U3bb5osDywNB8RzDVlDwDHbocAJveqqj1u8+SVD7jkWT4yvsHCpWqqWqAxb0zCA==}

  ms@2.1.3:
    resolution: {integrity: sha512-6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTlA==}

  nanoid@3.3.18:
    resolution: {integrity: sha512-DTg4MJbGMWkfi6VZFdNt2/caMbQy4Ou+Op/hJQvGEWcnVfoA1QA+xzRKAzw9jD6+GVOOeYr/mIcuDSdug6F6+w==}
    engines: {node: ^10 || ^12 || ^13.7 || ^14 || >=15.0.1}
    hasBin: true

  napi-postinstall@0.3.4:
    resolution: {integrity: sha512-PHI5f1O0EP5xJ9gQmFGMS6IZcrVvTjpXjz7Na41gTE7eE2hK11lg04CECCYEEjdc17EV4DO+fkGEtt7TpTaTiQ==}
    engines: {node: ^12.20.0 || ^14.18.0 || >=16.0.0}
    hasBin: true

  natural-compare@1.4.0:
    resolution: {integrity: sha512-OWND8ei3VtNC9h7V60qff3SVobHr996CTwgxubgyQYEpg290h9J0buyECNNJexkFm5sOajh5G116RYA1c8ZMSw==}

  next@15.5.23:
    resolution: {integrity: sha512-Gvd2WKgvxIXCGotxcI1im/Uf3rS3J3oZGw0g/uskg6AVBZhyE3aAbujkYWzS3xLmEPEtTLfkaVQUKK0KMTSIkA==}
    engines: {node: ^18.18.0 || ^19.8.0 || >= 20.0.0}
    hasBin: true
    peerDependencies:
      '@opentelemetry/api': ^1.1.0
      '@playwright/test': ^1.51.1
      babel-plugin-react-compiler: '*'
      react: ^18.2.0 || 19.0.0-rc-de68d2f4-20241204 || ^19.0.0
      react-dom: ^18.2.0 || 19.0.0-rc-de68d2f4-20241204 || ^19.0.0
      sass: ^1.3.0
    peerDependenciesMeta:
      '@opentelemetry/api':
        optional: true
      '@playwright/test':
        optional: true
      babel-plugin-react-compiler:
        optional: true
      sass:
        optional: true

  node-exports-info@1.6.2:
    resolution: {integrity: sha512-kXs9Go0cah0qHVV2v389IXQLdLCeE1xfFtjOAF+iobu0OIoG1pje8At2vMHyaPMiPMnG/LWP50twML21eMcAag==}
    engines: {node: '>= 0.4'}

  object-assign@4.1.1:
    resolution: {integrity: sha512-rJgTQnkUnH1sFw8yT6VSU3zD3sWmu6sZhIseY8VX+GRu3P6F7Fu+JNDoXfklElbLJSnc3FUQHVe4cU5hj+BcUg==}
    engines: {node: '>=0.10.0'}

  object-inspect@1.13.4:
    resolution: {integrity: sha512-W67iLl4J2EXEGTbfeHCffrjDfitvLANg0UlX3wFUUSTx92KXRFegMHUVgSqE+wvhAbi4WqjGg9czysTV2Epbew==}
    engines: {node: '>= 0.4'}

  object-keys@1.1.1:
    resolution: {integrity: sha512-NuAESUOUMrlIXOfHKzD6bpPu3tYt3xvjNdRIQ+FeT0lNb4K8WR70CaDxhuNguS2XG+GjkyMwOzsN5ZktImfhLA==}
    engines: {node: '>= 0.4'}

  object.assign@4.1.7:
    resolution: {integrity: sha512-nK28WOo+QIjBkDduTINE4JkF/UJJKyf2EJxvJKfblDpyg0Q+pkOHNTL0Qwy6NP6FhE/EnzV73BxxqcJaXY9anw==}
    engines: {node: '>= 0.4'}

  object.entries@1.1.9:
    resolution: {integrity: sha512-8u/hfXFRBD1O0hPUjioLhoWFHRmt6tKA4/vZPyckBr18l1KE9uHrFaFaUi8MDRTpi4uak2goyPTSNJLXX2k2Hw==}
    engines: {node: '>= 0.4'}

  object.fromentries@2.0.8:
    resolution: {integrity: sha512-k6E21FzySsSK5a21KRADBd/NGneRegFO5pLHfdQLpRDETUNJueLXs3WCzyQ3tFRDYgbq3KHGXfTbi2bs8WQ6rQ==}
    engines: {node: '>= 0.4'}

  object.groupby@1.0.3:
    resolution: {integrity: sha512-+Lhy3TQTuzXI5hevh8sBGqbmurHbbIjAi0Z4S63nthVLmLxfbj4T54a4CfZrXIrt9iP4mVAPYMo/v99taj3wjQ==}
    engines: {node: '>= 0.4'}

  object.values@1.2.1:
    resolution: {integrity: sha512-gXah6aZrcUxjWg2zR2MwouP2eHlCBzdV4pygudehaKXSGW4v2AsRQUK+lwwXhii6KFZcunEnmSUoYp5CXibxtA==}
    engines: {node: '>= 0.4'}

  optionator@0.9.4:
    resolution: {integrity: sha512-6IpQ7mKUxRcZNLIObR0hz7lxsapSSIYNZJwXPGeF0mTVqGKFIXj1DQcMoT22S3ROcLyY/rz0PWaWZ9ayWmad9g==}
    engines: {node: '>= 0.8.0'}

  own-keys@1.0.2:
    resolution: {integrity: sha512-19YVAg7T+WTrxggPukVq7DjTv6+PJ867TmhCvBsYwmbFCsZd344rq2Ld1p0wo8f8Qrrhgp82c6FJRqdXWtSEhg==}
    engines: {node: '>= 0.4'}

  p-limit@3.1.0:
    resolution: {integrity: sha512-TYOanM3wGwNGsZN2cVTYPArw454xnXj5qmWF1bEoAc4+cU/ol7GVh7odevjp1FNHduHc3KZMcFduxU5Xc6uJRQ==}
    engines: {node: '>=10'}

  p-locate@5.0.0:
    resolution: {integrity: sha512-LaNjtRWUBY++zB5nE/NwcaoMylSPk+S+ZHNB1TzdbMJMny6dynpAGt7X/tl/QYq3TIeE6nxHppbo2LGymrG5Pw==}
    engines: {node: '>=10'}

  parent-module@1.0.1:
    resolution: {integrity: sha512-GQ2EWRpQV8/o+Aw8YqtfZZPfNRWZYkbidE9k5rpl/hC3vtHHBfGm2Ifi6qWV+coDGkrUKZAxE3Lot5kcsRlh+g==}
    engines: {node: '>=6'}

  parse-entities@4.0.2:
    resolution: {integrity: sha512-GG2AQYWoLgL877gQIKeRPGO1xF9+eG1ujIb5soS5gPvLQ1y2o8FL90w2QWNdf9I361Mpp7726c+lj3U0qK1uGw==}

  parse5@7.3.0:
    resolution: {integrity: sha512-IInvU7fabl34qmi9gY8XOVxhYyMyuH2xUNpb2q8/Y+7552KlejkRvqvD19nMoUW/uQGGbqNpA6Tufu5FL5BZgw==}

  path-exists@4.0.0:
    resolution: {integrity: sha512-ak9Qy5Q7jYb2Wwcey5Fpvg2KoAc/ZIhLSLOSBmRmygPsGwkVVt0fZa0qrtMz+m6tJTAHfZQ8FnmB4MG4LWy7/w==}
```

```
    engines: {node: '>=8'}

  path-key@3.1.1:
    resolution: {integrity: sha512-ojmeN0qd+y0jszEtoY48r0Peq5dwMEkIlCOu6Q5f41lfkswXuKtYrhgoTpLnyIcHm24Uhqx+5Tqm2InSwLhE6Q==}
    engines: {node: '>=8'}

  path-parse@1.0.7:
    resolution: {integrity: sha512-LDJzPVEEEPR+y48z93A0Ed0yXb8pAByGWo/k5YYdYgpY2/2EsOsksJrq7lOHxryrVOn1ejG6oAp8ahvOIQD8sw==}

  picocolors@1.1.1:
    resolution: {integrity: sha512-xceH2snhtb5M9liqDsmEw56le376mTZkEX/jEb/RxNFyegNul7eNslCXP9FDj/Lcu0X8KEyMceP2ntpaHrDEVA==}

  picomatch@2.3.2:
    resolution: {integrity: sha512-V7+vQEJ06Z+c5tSye8S+nHUfI51xoXIXjHQ99cQtKUkQqqO1kO/KCJUfZXuB47h/YBlDhah2H3hdUGXn8ie0oA==}
    engines: {node: '>=8.6'}

  picomatch@4.0.5:
    resolution: {integrity: sha512-RvwwcruNjI1ncT5xRakeyS9Lf8lcItv34KD+aif+VH9kduAyfYBipGh12274xtenIPZ119/R9BdTBa8gAwSh0A==}
    engines: {node: '>=12'}

  possible-typed-array-names@1.1.0:
    resolution: {integrity: sha512-/+5VFTchJDoVj3bhoqi6UeymcD00DAwb1nJwamzPvHEszJ4FpF6SNNbUbOS8yI56qHzdV8eK0qEfOSiodkTdxg==}
    engines: {node: '>= 0.4'}

  postcss@8.4.31:
    resolution: {integrity: sha512-PS08Iboia9mts/2ygV3eLpY5ghnUcfLV/EXTOW1E2qYxJKGGBUtNjN76FYHnMs36RmARn41bC0AZmn+rR0OVpQ==}
    engines: {node: ^10 || ^12 || >=14}

  postcss@8.5.26:
    resolution: {integrity: sha512-u82N74LFzG8ca+dD8puPnplTXoGH4fTPpVGuIbt36G3qvNlkvfD0lEAZSxaly3KX8TS/L1A1gsCEmvKmBcVbkQ==}
    engines: {node: ^10 || ^12 || >=14}

  prelude-ls@1.2.1:
    resolution: {integrity: sha512-vkcDPrRZo1QZLbn5RLGPpg/WmIQ65qoWWhcGKf/b5eplkkarX0m9z8ppCat4mlOqUsWpyNuYgO3VRyrYHSzX5g==}
    engines: {node: '>= 0.8.0'}

  prettier@3.9.6:
    resolution: {integrity: sha512-OpN0zzVdiaiAhxpuuj5efpIS4sY9j7bY6uR5mnj5yPzGkdkjNKSJeUThPb60Jw29QuAZgA4o+/iB49kFiaBX6g==}
    engines: {node: '>=14'}
    hasBin: true

  prop-types@15.8.1:
    resolution: {integrity: sha512-oj87CgZICdulUohogVAR7AjlC0327U4el4L6eAvOqCeudMDVU0NThNaV+b9Df4dXgSP1gXMTnPdhfe/2qDH5cg==}

  property-information@7.2.0:
    resolution: {integrity: sha512-IAtzIB6sUiWaJYrX9smp3V46pBGbBeLFRGdh25kg1334VcBlD8HzhPeNIWQH9zhGmo2itIe25EHt9dQP7G5hmg==}

  punycode@2.3.1:
    resolution: {integrity: sha512-vYt7UD1U9Wg6138shLtLOvdAu+8DsC/ilFtEVHcH+wydcSpNE20AfSOduf6MkRFahL5FY7X1oU7nKVZFtfq8Fg==}
    engines: {node: '>=6'}

  queue-microtask@1.2.3:
    resolution: {integrity: sha512-NuaNSa6flKT5JaSYQzJok04JzTL1CA6aGhv5rfLW3PgqA+M2ChpZQnAC8h8i4ZFkBS8X5RqkDBHA7r4hej3K9A==}

  react-dom@19.2.8:
    resolution: {integrity: sha512-rVprimfGBG3DR+Tq0IQG2DT5PxKth1WIGDmj5yPmlzr4YBe7uyE+Du4oVqTDXZSHGGGXRtTJEGSSePyQCMBglQ==}
    peerDependencies:
      react: ^19.2.8

  react-is@16.13.1:
    resolution: {integrity: sha512-24e6ynE2H+OKt4kqsOvNd8kBpV65zoxbA4BVsEOB3ARVWQki/DHzaUoC5KuON/BiccDaCCTZBuOcfZs70kR8bQ==}

  react-markdown@9.1.0:
    resolution: {integrity: sha512-xaijuJB0kzGiUdG7nc2MOMDUDBWPyGAjZtUrow9XxUeua8IqeP+VlIfAZ3bphpcLTnSZXz6z9jcVC/TCwbfgdw==}
    peerDependencies:
      '@types/react': '>=18'
      react: '>=18'

  react-remove-scroll-bar@2.3.8:
    resolution: {integrity: sha512-9r+yi9+mgU33AKcj6IbT9oRCO78WriSj6t/cF8DWBZJ9aOGPOTEDvdUDz1FwKim7QXWwmHqtdHnRJfhAxEG46Q==}
    engines: {node: '>=10'}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8.0 || ^17.0.0 || ^18.0.0 || ^19.0.0
    peerDependenciesMeta:
      '@types/react':
        optional: true

  react-remove-scroll@2.7.2:
    resolution: {integrity: sha512-Iqb9NjCCTt6Hf+vOdNIZGdTiH1QSqr27H/Ek9sv/a97gfueI/5h1s3yRi1nngzMUaOOToin5dI1dXKdXiF+u0Q==}
    engines: {node: '>=10'}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8.0 || ^17.0.0 || ^18.0.0 || ^19.0.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  react-style-singleton@2.2.3:
    resolution: {integrity: sha512-b6jSvxvVnyptAiLjbkWLE/lOnR4lfTtDAl+eUC7RZy+QQWc6wRzIV2CE6xBuMmDxc2qIihtDCZD5NPOFl7fRBQ==}
    engines: {node: '>=10'}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8.0 || ^17.0.0 || ^18.0.0 || ^19.0.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  react@19.2.8:
    resolution: {integrity: sha512-PWaYA1L/q9u2u7xYQi+Y3L3Yfnie7XyLeaJICV1MGD6LprsBxcAqGjYyr0eY3p+QdsA+x/Irkt4Qif8D63+Sbw==}
    engines: {node: '>=0.10.0'}

  reflect.getprototypeof@1.0.10:
    resolution: {integrity: sha512-00o4I+DVrefhv+nX0ulyi3biSHCPDe+yLv5o/p6d/UVlirijB8E16FtfwSAi4g3tcqrQ4lRAqQSoFEZJehYEcw==}
    engines: {node: '>= 0.4'}
```

```
  regexp.prototype.flags@1.5.4:
    resolution: {integrity: sha512-dYqgNSZbDwkaJ2ceRd9ojCGjBq+mOm9LmtXnAnEGyHhN/5R7iDW2TRw3h+o/jCFxus3P2LfWIIiwowAjANm7IA==}
    engines: {node: '>= 0.4'}

  rehype-raw@7.0.0:
    resolution: {integrity: sha512-/aE8hCfKlQeA8LmyeyQvQF3eBiLRGNlfBJEvWH7ivp9sBqs7TNqBL5X3v157rM4IFETqDnIOO+z5M/biZbo9Ww==}

  rehype-sanitize@6.0.0:
    resolution: {integrity: sha512-CsnhKNsyI8Tub6L4sm5ZFsme4puGfc6pYylvXo1AeqaGbjOYyzNv3qZPwvs0oMJ39eryyeOdmxwUIo94IpEhqg==}

  remark-gfm@4.0.1:
    resolution: {integrity: sha512-1quofZ2RQ9EWdeN34S79+KExV1764+wCUGop5CPL1WGdD0ocPpu91lzPGbwWMECpEpd42kJGQwzRfyov9j4yNg==}

  remark-parse@11.0.0:
    resolution: {integrity: sha512-FCxlKLNGknS5ba/1lmpYijMUzX2esxW5xQqjWxw2eHFfS2MSdaHVINFmhjo+qN1WhZhNimq0dZATN9pH0IDrpA==}

  remark-rehype@11.1.2:
    resolution: {integrity: sha512-Dh7l57ianaEoIpzbp0PC9UKAdCSVklD8E5Rpw7ETfbTl3FqcOOgq5q2LVDhgGCkaBv7p24JXikPdvhhmHvKMsw==}

  remark-stringify@11.0.0:
    resolution: {integrity: sha512-1OSmLd3awB/t8qdoEOMazZkNsfVTeY4fTsgzcQFdXNq8ToTN4ZGwrMnlda4K6smTFKD+GRV6O48i6Z4iKgPPpw==}

  resolve-from@4.0.0:
    resolution: {integrity: sha512-pb/MYmXstAkysRFx8piNI1tGFNQIFA3vkE3Gq4EuA1dF6gHp/+vgZqsCGJapvy8N3Q+4o7FwvquPJcnZ7RYy4g==}
    engines: {node: '>=4'}

  resolve-pkg-maps@1.0.0:
    resolution: {integrity: sha512-seS2Tj26TBVOC2NIc2rOe2y2ZO7efxITtLZcGSOnHHNOQ7CkiUBfw0Iw2ck6xkIhPwLhKNLS8BO+hEpngQlqzw==}

  resolve@2.0.0-next.7:
    resolution: {integrity: sha512-tqt+NBWwyaMgw3zDsnygx4CByWjQEJHOPMdslYhppaQSJUtL/D4JO9CcBBlhPoI8lz9oJIDXkwXfhF4aWqP8xQ==}
    engines: {node: '>= 0.4'}
    hasBin: true

  reusify@1.1.0:
    resolution: {integrity: sha512-g6QUff04oZpHs0eG5p83rFLhHeV00ug/Yf9nZM6fLeUrPguBTkTQOdpAWWspMh55TZfVQDPaN3NQJfbVRAxdIw==}
    engines: {iojs: '>=1.0.0', node: '>=0.10.0'}

  run-parallel@1.2.0:
    resolution: {integrity: sha512-5l4VyZR86LZ/lDxZTR6jqL8AFE2S0IFLMP26AbjsLVADxHdhB/c0GUsH+y39UfCi3dzz8OlQuPmnaJOMoDHQBA==}

  safe-array-concat@1.1.4:
    resolution: {integrity: sha512-wtZlHyOje6OZTGqAoaDKxFkgRtkF9CnHAVnCHKfuj200wAgL+bSJhdsCD2l0Qx/2ekEXjPWcyKkfGb5CPboslg==}
    engines: {node: '>=0.4'}

  safe-push-apply@1.0.0:
    resolution: {integrity: sha512-iKE9w/Z7xCzUMIZqdBsp6pEQvwuEebH4vdpjcDWnyzaI6yl6O9FHvVpmGelvEHNsoY6wGblkxR6Zty/h00WiSA==}
    engines: {node: '>= 0.4'}

  safe-regex-test@1.1.0:
    resolution: {integrity: sha512-x/+Cz4YrimQxQccJf5mKEbIa1NzeCRNI5Ecl/ekmlYaampdNLPalVyIcCZNNH3MvmqBugV5TMYZXv0ljslUlaw==}
    engines: {node: '>= 0.4'}

  scheduler@0.27.0:
    resolution: {integrity: sha512-eNv+WrVbKu1f3vbYJT/xtiF5syA5HPIMtf9IgY/nKg0sWqzAUEvqY/xm7OcZc/qafLx/iO9FgOmeSAp4v5ti/Q==}

  semver@6.3.1:
    resolution: {integrity: sha512-BR7VvDCVHO+q2xBEWskxS6DJE1qRnb7DxzUrogb71CWoSficBxYsiAGd+Kl0mmq/MprG9yArRkyrQxTO6XjMzA==}
    hasBin: true

  semver@7.8.5:
    resolution: {integrity: sha512-Y7/KDsb8LjooZpwaqGyulO6DQlksgCncchHGk+sZIY4SBvUocMBEFH5Ur1fI4dV+Jvl0w6cjvucaIi40puRioA==}
    engines: {node: '>=10'}
    hasBin: true

  set-function-length@1.2.2:
    resolution: {integrity: sha512-pgRc4hJ4/sNjWCSS9AmnS40x3bNMDTknHgL5UaMBTMyJnU90EgWh1Rz+MC9eFu4BuN/UwZjKQuY/1v3rM7HMfg==}
    engines: {node: '>= 0.4'}

  set-function-name@2.0.2:
    resolution: {integrity: sha512-7PGFlmtwsEADb0WYyvCMa1t+yke6daIG4Wirafur5kcf+MhUnPms1UeR0CKQdTZD81yESwMHbtn+TR+dMviakQ==}
    engines: {node: '>= 0.4'}

  set-proto@1.0.0:
    resolution: {integrity: sha512-RJRdvCo6IAnPdsvP/7m6bsQqNnn1FCBX5ZNtFL98MmFF/4xAIJTIg1YbHW5DC2W5SKZanrC6i4HsJqlajw/dZw==}
    engines: {node: '>= 0.4'}

  sharp@0.34.5:
    resolution: {integrity: sha512-Ou9I5Ft9WNcCbXrU9cMgPBcCK8LiwLqcbywW3t4oDV37n1pzpuNLsYiAV8eODnjbtQlSDwZ2cUEeQz4E54Hltg==}
    engines: {node: ^18.17.0 || ^20.3.0 || >=21.0.0}

  shebang-command@2.0.0:
    resolution: {integrity: sha512-kHxr2zZpYtdmrN1qDjrrX/Z1rR1kG8Dx+gkpK1G4eXmvXswmcE1hTWBWYUzlraYw1/yZp6YuDY77YtvbN0dmDA==}
    engines: {node: '>=8'}

  shebang-regex@3.0.0:
    resolution: {integrity: sha512-7++dFhtcx3353uBaq8DDR4NuxBetBzC7ZQOhmTQInHEd6bSrXdiEyzCvG07Z44UYdLShWUyXt5M/yhz8ekcb1A==}
    engines: {node: '>=8'}

  side-channel-list@1.0.1:
    resolution: {integrity: sha512-mjn/0bi/oUURjc5Xl7IaWi/OJJJumuoJFQJfDDyO46+hBWsfaVM65TBHq2eoZBhzl9EchxOijpkbRC8SVBQU0w==}
    engines: {node: '>= 0.4'}

  side-channel-map@1.0.1:
    resolution: {integrity: sha512-VCjCNfgMsby3tTdo02nbjtM/ewra6jPHmpThenkTYh8pG9ucZ/1P8So4u4FGBek/BjpOVsDCMoLA/iuBKIFXRA==}
    engines: {node: '>= 0.4'}

  side-channel-weakmap@1.0.2:
    resolution: {integrity: sha512-WPS/HvHQTYnHisLo9McqBHOJk2FkHO/tlpvldyrnem4aeQp4hai3gythswg6p01oSoTl58rcpiFAjF2br2Ak2A==}
    engines: {node: '>= 0.4'}

  side-channel@1.1.1:
    resolution: {integrity: sha512-6x6dK6zJdpTzF4sQeNYxwtvBzf6Eg4GtlesS94HOvTudUeyK2WXAaIfmDgsyslYrRBeFIlsi54AYsFGUuhmvrQ==}
    engines: {node: '>= 0.4'}
```

```yaml
    source-map-js@1.2.1:
      resolution: {integrity: sha512-UXWMKhLOwVKb728IUtQPXxfYU+usdybtUrK/8uGE8CQMvrhOpwvzDBwj0QhSL7MQc7vIsISBG8VQ8+IDQxpfQA==}
      engines: {node: '>=0.10.0'}

    space-separated-tokens@2.0.2:
      resolution: {integrity: sha512-PEGlAwrG8yXGXRjW32fGbg66JAlOAwbObuqVoJpv/mRgoWDQfgH1wDPvtzWyUSNAXBGSk8h755YDbbcEy3SH2Q==}

    stable-hash@0.0.5:
      resolution: {integrity: sha512-+L3ccpzibovGXFK+Ap/f8LOS0ahMrHTf3xu7mMLSpEGU0EO9ucaysSylKo9eRDFNhWve/y275iPmIZ4z39a9iA==}

    stop-iteration-iterator@1.1.0:
      resolution: {integrity: sha512-eLoXW/DHyl62zxY4SCaIgnRhuMr6ri4juEYARS8E6sCEqzKpOiE521Ucofdx+KnDZl5xmvGYaaKCk5FEOxJCoQ==}
      engines: {node: '>= 0.4'}

    string.prototype.includes@2.0.1:
      resolution: {integrity: sha512-o7+c9bW6zpAdJHTtujeePODAhkuicdAryFsfVKwA+wGw89wJ4GTY484WTucM9hLtDEOpOvI+aHnzqnC5lHp4Rg==}
      engines: {node: '>= 0.4'}

    string.prototype.matchall@4.0.12:
      resolution: {integrity: sha512-6CC9uyBL+/48dYizRf7H7VAYCMCNTBeM78x/VTUe9bFEaxBepPJDa1Ow99LqI/1yF7kuy7Q3cQsYMrcjGUcskA==}
      engines: {node: '>= 0.4'}

    string.prototype.repeat@1.0.0:
      resolution: {integrity: sha512-0u/TldDbKD8bFCQ/4f5+mNRrXwZ8hg2w7ZR8wa16e8z9XpePWl3eGEcUD0OXpEH/VJH/2G3gjUtR3ZOiBe2S/w==}

    string.prototype.trim@1.2.11:
      resolution: {integrity: sha512-PwvK7BU+CMTJGYQCTZb5RWXIML92lftJLhQz1tBzgKiqGxJaMlBAa48POXaNAC2s4y8jr3EFqrkF9+44neS46w==}
      engines: {node: '>= 0.4'}

    string.prototype.trimend@1.0.10:
      resolution: {integrity: sha512-2+3aDAOmPTmuFwjDnmJG2ctEkQKVki7vOSqaxkv42Mowj1V6PnvuwFCRrR5lChUux1TBskPjfkeTOhqczDMxTw==}
      engines: {node: '>= 0.4'}

    string.prototype.trimstart@1.0.8:
      resolution: {integrity: sha512-UXSH262CSZY1tfu3G3Secr6uGLCFVPMhIqHjlgCUtCCcgihYc/xKs9djMTMUOb2j1mVSeU8EU6NWc/iQKU6Gfg==}
      engines: {node: '>= 0.4'}

    stringify-entities@4.0.4:
      resolution: {integrity: sha512-IwfBptatlO+QCJUo19AqvrPNqlVMpW9YEL2LIVY+Rpv2qsjCGxaDLNRgeGsQWJhfItebuJhsGSLjaBbNSQ+ieg==}

    strip-bom@3.0.0:
      resolution: {integrity: sha512-vavAMRXOgBVNF6nyEEmL3DBK19iRpDcoIwW+swQ+CbGiu7lju6t+JklA1MHweoWtadgt4ISVUsXLyDq34ddcwA==}
      engines: {node: '>=4'}

    strip-json-comments@3.1.1:
      resolution: {integrity: sha512-6fPc+R4ihwqP6N/aIv2f1gMH8lOVtWQHoqC4yK6oSDVVocumAsfCqjkXnqiYMhmMwS/mEHLp7Vehlt3ql6lEig==}
      engines: {node: '>=8'}

    style-to-js@1.1.21:
      resolution: {integrity: sha512-RjQetxJrrUJLQPHbLku6U/ocGtzyjbJMP9lCNK7Ag0CNh690nSH8woqWH9u16nMjYBAok+i7JO1NP2pOy8IsPQ==}

    style-to-object@1.0.14:
      resolution: {integrity: sha512-LIN7rULI0jBscWQYaSswptyderlarFkjQ+t79nzty8tcIAceVomEVlLzH5VP4Cmsv6MtKhs7qaAiwlcp+Mgaxw==}

    styled-jsx@5.1.6:
      resolution: {integrity: sha512-qSVyDTeMotdvQYoHWLNGwRFJHC+i+ZvdBRYosOFgC+Wg1vx4frN2/RG/NA7SYqqvKNLf39P2LSRA2pu6n0XYZA==}
      engines: {node: '>= 12.0.0'}
      peerDependencies:
        '@babel/core': '*'
        babel-plugin-macros: '*'
        react: '>= 16.8.0 || 17.x.x || ^18.0.0-0 || ^19.0.0-0'
      peerDependenciesMeta:
        '@babel/core':
          optional: true
        babel-plugin-macros:
          optional: true

    supports-color@7.2.0:
      resolution: {integrity: sha512-qpCAvRl9stuOHveKsn7HncJRvv501qIacKzQlO/+Lwxc9+0q2wLyv4Dfvt80/DPn2pqOBsJdDiogXGR9+OvwRw==}
      engines: {node: '>=8'}

    supports-preserve-symlinks-flag@1.0.0:
      resolution: {integrity: sha512-ot0WnXS9fgdkgIcePe6RHNk1WA8+muPa6cSjeR3V8K27q9BB1rTE3R1p7Hv0z1ZyAc8s6Vvv8DIyWf681MAt0w==}
      engines: {node: '>= 0.4'}

    tailwind-merge@2.6.1:
      resolution: {integrity: sha512-Oo6tHdpZsGpkKG88HJ8RR1rg/RdnEkQEfMoEk2x1XRI3F1AxeU+ijRXpiVUF4UbLfcxxRGw6TbUINKYdWVsQTQ==}

    tailwindcss@4.3.3:
      resolution: {integrity: sha512-gOhV3P7ufE62QDGg1zVaTgCR+EtPv92k2nIhVcVKcLmxT1sUBsQGhnZj175j+MqRt4zLF7ic+sCYjfhxMxj7YQ==}

    tapable@2.3.3:
      resolution: {integrity: sha512-uxc/zpqFg6x7C8vOE7lh6Lbda8eEL9zmVm/PLeTPBRhh1xCgdWaQ+J1CUieGpIfm2HdtsUpRv+HshiasBMcc6A==}
      engines: {node: '>=6'}

    tinyglobby@0.2.17:
      resolution: {integrity: sha512-wXR/dYpcqKmfWpEdZjiKJOwCNFndD0DMnrW/cYjVGttEkBfVgcLFHoNrlj47mjOVic9yyNu65alsgF4NQyTa2g==}
      engines: {node: '>=12.0.0'}

    to-regex-range@5.0.1:
      resolution: {integrity: sha512-65P7iz6X5yEr1cwcgvQxbbIw7Uk3gOy5dIdtZ4rDveLqhrdJP+Li/Hx6tyK0NEb+2GCyneCMJiGqrADCSNk8sQ==}
      engines: {node: '>=8.0'}

    trim-lines@3.0.1:
      resolution: {integrity: sha512-kRj8B+YHZCc9kQYdWfJB2/oUl9rA99qbowYYBtr4ui4mZyAQ2JpvVBd/6U2YloATfqBhBTSMhTpgBHtU0Mf3Rg==}

    trough@2.2.0:
      resolution: {integrity: sha512-tmMpK00BjZiUyVyvrBK7knerNgmgvcV/KLVyuma/SC+TQN167GrMRciANTz09+k3zW8L8t60jWO1GpfkZdjTaw==}

    ts-api-utils@2.5.0:
      resolution: {integrity: sha512-OJ/ibxhPlqrMM0UiNHJ/0CKQkoKF243/AEmplt3qpRgkW8VG7IfOS41h7V8TjITqdByHzrjcS/2si+y4lIh8NA==}
      engines: {node: '>=18.12'}
      peerDependencies:
        typescript: '>=4.8.4'
```

```yaml

  tsconfig-paths@3.15.0:
    resolution: {integrity: sha512-2Ac2RgzDe/cn48GvOe3M+o82pEFewD3UPbyoUHHdKasHwJKjds4fLXWf/Ux5kATBKN20oaFGu+jbElp1pos0mg==}

  tslib@2.8.1:
    resolution: {integrity: sha512-oJFu94HQb+KVduSUQL7wnpmqnfmLsOA/nAh6b6EH0wCEoK0/mPeXU6c3wKDV83MkOuHPRHtSXKKU99IBazS/2w==}

  type-check@0.4.0:
    resolution: {integrity: sha512-XleUoc9uwGXqjWwXaUTZAmzMcFZ5858QA2vvx1Ur5xIcixXIP+8LnFDgRplU30us6teqdlskFfu+ae4K79Ooew==}
    engines: {node: '>= 0.8.0'}

  typed-array-buffer@1.0.3:
    resolution: {integrity: sha512-nAYYwfY3qnzX30IkA6AQZjVbtK6duGontcQm1WSG1MD94YLqK0515GNApXkoxKOWMusVssAHWLh9SeaoefYFGw==}
    engines: {node: '>= 0.4'}

  typed-array-byte-length@1.0.3:
    resolution: {integrity: sha512-BaXgOuIxz8n8pIq3e7Atg/7s+DpiYrxn4vdot3w9KbnBhcRQq6o3xemQdIfynqSeXeDrF32x+WvfzmOjPiY9lg==}
    engines: {node: '>= 0.4'}

  typed-array-byte-offset@1.0.4:
    resolution: {integrity: sha512-bTlAFB/FBYMcuX81gbL4OcpH5PmlFHqlCCpAl8AlEzMz5k53oNDvN8p1PNOWLEmI2x4orp3raOFB51tv9X+MFQ==}
    engines: {node: '>= 0.4'}

  typed-array-length@1.0.8:
    resolution: {integrity: sha512-phPGCwqr2+Qo0fwniCE8e4pKnGu/yFb5nD5Y8bf0EEeiI5GklnACYA9GFy/DrAeRrKHXvHn+1SUsOWgJp6RO+g==}
    engines: {node: '>= 0.4'}

  typescript@5.9.3:
    resolution: {integrity: sha512-jl1vZzPDinLr9eUt3J/t7V6FgNEw9QjvBPdysz9KfQDD41fQrC2Y4vKQdiaUpFT4bXlb1RHhLpp8wtm6M5TgSw==}
    engines: {node: '>=14.17'}
    hasBin: true

  unbox-primitive@1.1.0:
    resolution: {integrity: sha512-nWJ91DjeOkej/TA8pXQ3myruKpKEYgqvpw9lz4OPHj/NWFNluYrjbz9j01CJ8yKQd2g4jFoOkINCTW2I5LEEyw==}
    engines: {node: '>= 0.4'}

  undici-types@6.21.0:
    resolution: {integrity: sha512-iwDZqg0QAGrg9Rav5H4n0M64c3mkR59cJ6wQp+7C4nI0gsmExaedaYLNO44eT4AtBBwjbTiGPMlt2Md0T9H9JQ==}

  unified@11.0.5:
    resolution: {integrity: sha512-xKvGhPWw3k84Qjh8bI3ZeJjqnyadK+GEFtazSfZv/rKeTkTjOJho6mFqh2SM96iIcZokxiOpg78GazTSg8+KHA==}

  unist-util-is@6.0.1:
    resolution: {integrity: sha512-LsiILbtBETkDz8I9p1dQ0uyRUWuaQzd/cuEeS1hoRSyW5E5XGmTzlwY1OrNzzakGowI9Dr/I8HVaw4hTtnxy8g==}

  unist-util-position@5.0.0:
    resolution: {integrity: sha512-fucsC7HjXvkB5R3kTCO7kUjRdrS0BJt3M/FPxmHMBOm8JQi2BsHAHFsy27E0EolP8rp0NzXsJ+jNPyDWvOJZPA==}

  unist-util-stringify-position@4.0.0:
    resolution: {integrity: sha512-0ASV06AAoKCDkS2+xw5RXJywruurpbC4JZSm7nr7MOt1ojAzvyyaO+UxZf18j8FCF6kmzCZKcAgN/yu2gm2XgQ==}

  unist-util-visit-parents@6.0.2:
    resolution: {integrity: sha512-goh1s1TBrqSqukSc8wrjwWhL0hiJxgA8m4kFxGlQ+8FYQ3C/m11FcTs4YYem7V664AhHVvgoQLk890Ssdsr2IQ==}

  unist-util-visit@5.1.0:
    resolution: {integrity: sha512-m+vIdyeCOpdr/QeQCu2EzxX/ohgS8KbnPDgFni4dQsfSCtpz8UqDyY5GjRru8PDKuYn7Fq19j1CQ+nJSsGKOzg==}

  unrs-resolver@1.12.2:
    resolution: {integrity: sha512-dmlRxBJJayXjqTwC+JtF1HhJmgf3ftQ3YejFcZrf4+KKtJv0qDsK1pjqaaVjG7wJ5NJ6UVP1OqRMQ71Z4C3rxQ==}

  uri-js@4.4.1:
    resolution: {integrity: sha512-7rKUyy33Q1yc98pQ1DAmLtwX109F7TIfWlW1Ydo8Wl1ii1SeHieeh0HHfPeL2fMXK6z0s8ecKs9frCuLJvndBg==}

  use-callback-ref@1.3.3:
    resolution: {integrity: sha512-jQL3lRnocaFtu3V00JToYz/4QkNWswxijDaCVNZRiRTO3HQDLsdu1ZtmIUvV4yPp+rvWm5j0y0TG/S61cuijTg==}
    engines: {node: '>=10'}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8.0 || ^17.0.0 || ^18.0.0 || ^19.0.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  use-sidecar@1.1.3:
    resolution: {integrity: sha512-Fedw0aZvkhynoPYlA5WXrMCAMm+nSWdZt6lzJQ7Ok8S6Q+VsHmHpRWndVRJ8Be0ZbkfPc5LRYH+5XrzXcEeLRQ==}
    engines: {node: '>=10'}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8.0 || ^17.0.0 || ^18.0.0 || ^19.0.0 || ^19.0.0-rc
    peerDependenciesMeta:
      '@types/react':
        optional: true

  vfile-location@5.0.3:
    resolution: {integrity: sha512-5yXvWDEgqeiYiBe1lbxYF7UMAIm/IcopxMHrMQDq3nvKcjPKIhZklUKL+AE7J7uApI4kwe2snsK+eI6UTj9EHg==}

  vfile-message@4.0.3:
    resolution: {integrity: sha512-QTHzsGd1EhbZs4AsQ20JX1rC3cOlt/IWJruk893DfLRr57lcnOeMaWG4K0JrRta4mIJZKth2Au3mM3u03/JWKw==}

  vfile@6.0.3:
    resolution: {integrity: sha512-KzIbH/9tXat2u30jf+smMwFCsno4wHVdNmzFyL+T/L3UGqqk6JKfVqOFOZEpZSHADH1k40ab6NUIXZq422ov3Q==}

  web-namespaces@2.0.1:
    resolution: {integrity: sha512-bKr1DkiNa2krS7qxNtdrtHAmzuYGFQLiQ13TsorsdT6ULTkPLKuu5+GsFpDlg6JFjUTwX2DyhMPG2be8uPrqsQ==}

  which-boxed-primitive@1.1.1:
    resolution: {integrity: sha512-TbX3mj8n0odCBFVlY8AxkqcHASw3L60jIuF8jFP78az3C2YhmGvqbHBpAjTRH2/xqYunrJ9g1jSyjCjpoWzIAA==}
    engines: {node: '>= 0.4'}

  which-builtin-type@1.2.1:
    resolution: {integrity: sha512-6iBczoX+kDQ7a3+YJBnh3T+KZRxM/iYNPXicqk66/Qfm1b93iu+yOImkg0zHbj5LNOcNv1TEADiZ0xa34B4q6Q==}
    engines: {node: '>= 0.4'}

  which-collection@1.0.2:
    resolution: {integrity: sha512-K4jVyjnBdgvc86Y6BkaLZEN933SwYOuBFkdmBu9ZfkcAbdVbpITnDmjvZ/aQjRXQrv5EPkTnD1s39GiiqbngCw==}
```

```yaml
    engines: {node: '>= 0.4'}

  which-typed-array@1.1.22:
    resolution: {integrity: sha512-fvO4ExWMFsqyhG3AiPAObMuY1lxaqgYcxbc49CNdWDDECOJNgQyvsOWVwbZc+qf3rzRtxojBK+CMEv0Ld5CYpw==}
    engines: {node: '>= 0.4'}

  which@2.0.2:
    resolution: {integrity: sha512-BLI3Tl1TW3Pvl70l3yq3Y64i+awpwXqsGBYWkkqMtnbXgrMD+yj7rhW0kuEDxzJaYXGjEW5ogapKNMEKNMjibA==}
    engines: {node: '>= 8'}
    hasBin: true

  word-wrap@1.2.5:
    resolution: {integrity: sha512-BN22B5eaMMI9UMtjrGd5g5eCYPpCPDUy0FJXbYsaT5zYxjFOckS53SQDE3pWkVoWpHXVb3BrYcEN4Twa55B5cA==}
    engines: {node: '>=0.10.0'}

  yocto-queue@0.1.0:
    resolution: {integrity: sha512-rVksvsnNCdJ/ohGc6xgPwyN8eheCxsiLM8mxuE/t/mOVqJewPuO1miLpTHQiRgTKCLexL4MeAFVagts7HmNZ2Q==}
    engines: {node: '>=10'}

  zod@3.25.76:
    resolution: {integrity: sha512-gzUt/qt81nXsFGKIFcC3YnfEAx5NkunCfnDlvuBSSFS02bcXu4Lmea0AFIUwbLWxWPx3d9p8S5QoaujKcNQxcQ==}

  zwitch@2.0.4:
    resolution: {integrity: sha512-bXE4cR/kVZhKZX/RjPEflHaKVhUVl85noU3v6b8apfQEc1x4A+zBxjZ4lN8LqGd6WZ3dl98pY4o717VFmoPp+A==}

snapshots:

  '@alloc/quick-lru@5.2.0': {}

  '@emnapi/core@1.10.0':
    dependencies:
      '@emnapi/wasi-threads': 1.2.1
      tslib: 2.8.1
    optional: true

  '@emnapi/runtime@1.10.0':
    dependencies:
      tslib: 2.8.1
    optional: true

  '@emnapi/runtime@1.11.3':
    dependencies:
      tslib: 2.8.1
    optional: true

  '@emnapi/wasi-threads@1.2.1':
    dependencies:
      tslib: 2.8.1
    optional: true

  '@eslint-community/eslint-utils@4.10.1(eslint@9.39.5(jiti@2.7.0))':
    dependencies:
      eslint: 9.39.5(jiti@2.7.0)
      eslint-visitor-keys: 3.4.3

  '@eslint-community/regexpp@4.12.2': {}

  '@eslint/config-array@0.21.2':
    dependencies:
      '@eslint/object-schema': 2.1.7
      debug: 4.4.3
      minimatch: 3.1.5
    transitivePeerDependencies:
      - supports-color

  '@eslint/config-helpers@0.4.2':
    dependencies:
      '@eslint/core': 0.17.0

  '@eslint/core@0.17.0':
    dependencies:
      '@types/json-schema': 7.0.15

  '@eslint/eslintrc@3.3.6':
    dependencies:
      ajv: 6.15.0
      debug: 4.4.3
      espree: 10.4.0
      globals: 14.0.0
      ignore: 5.3.2
      import-fresh: 3.3.1
      js-yaml: 4.3.1
      minimatch: 3.1.5
      strip-json-comments: 3.1.1
    transitivePeerDependencies:
      - supports-color

  '@eslint/js@9.39.5': {}

  '@eslint/object-schema@2.1.7': {}

  '@eslint/plugin-kit@0.4.1':
    dependencies:
      '@eslint/core': 0.17.0
      levn: 0.4.1

  '@floating-ui/core@1.8.0':
    dependencies:
      '@floating-ui/utils': 0.2.12

  '@floating-ui/dom@1.8.0':
    dependencies:
      '@floating-ui/core': 1.8.0
      '@floating-ui/utils': 0.2.12

```

```yaml
  '@floating-ui/react-dom@2.1.9(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@floating-ui/dom': 1.8.0
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)

  '@floating-ui/utils@0.2.12': {}

  '@humanfs/core@0.19.2':
    dependencies:
      '@humanfs/types': 0.15.0

  '@humanfs/node@0.16.8':
    dependencies:
      '@humanfs/core': 0.19.2
      '@humanfs/types': 0.15.0
      '@humanwhocodes/retry': 0.4.3

  '@humanfs/types@0.15.0': {}

  '@humanwhocodes/module-importer@1.0.1': {}

  '@humanwhocodes/retry@0.4.3': {}

  '@img/colour@1.1.0':
    optional: true

  '@img/sharp-darwin-arm64@0.34.5':
    optionalDependencies:
      '@img/sharp-libvips-darwin-arm64': 1.2.4
    optional: true

  '@img/sharp-darwin-x64@0.34.5':
    optionalDependencies:
      '@img/sharp-libvips-darwin-x64': 1.2.4
    optional: true

  '@img/sharp-libvips-darwin-arm64@1.2.4':
    optional: true

  '@img/sharp-libvips-darwin-x64@1.2.4':
    optional: true

  '@img/sharp-libvips-linux-arm64@1.2.4':
    optional: true

  '@img/sharp-libvips-linux-arm@1.2.4':
    optional: true

  '@img/sharp-libvips-linux-ppc64@1.2.4':
    optional: true

  '@img/sharp-libvips-linux-riscv64@1.2.4':
    optional: true

  '@img/sharp-libvips-linux-s390x@1.2.4':
    optional: true

  '@img/sharp-libvips-linux-x64@1.2.4':
    optional: true

  '@img/sharp-libvips-linuxmusl-arm64@1.2.4':
    optional: true

  '@img/sharp-libvips-linuxmusl-x64@1.2.4':
    optional: true

  '@img/sharp-linux-arm64@0.34.5':
    optionalDependencies:
      '@img/sharp-libvips-linux-arm64': 1.2.4
    optional: true

  '@img/sharp-linux-arm@0.34.5':
    optionalDependencies:
      '@img/sharp-libvips-linux-arm': 1.2.4
    optional: true

  '@img/sharp-linux-ppc64@0.34.5':
    optionalDependencies:
      '@img/sharp-libvips-linux-ppc64': 1.2.4
    optional: true

  '@img/sharp-linux-riscv64@0.34.5':
    optionalDependencies:
      '@img/sharp-libvips-linux-riscv64': 1.2.4
    optional: true

  '@img/sharp-linux-s390x@0.34.5':
    optionalDependencies:
      '@img/sharp-libvips-linux-s390x': 1.2.4
    optional: true

  '@img/sharp-linux-x64@0.34.5':
    optionalDependencies:
      '@img/sharp-libvips-linux-x64': 1.2.4
    optional: true

  '@img/sharp-linuxmusl-arm64@0.34.5':
    optionalDependencies:
      '@img/sharp-libvips-linuxmusl-arm64': 1.2.4
    optional: true

  '@img/sharp-linuxmusl-x64@0.34.5':
    optionalDependencies:
      '@img/sharp-libvips-linuxmusl-x64': 1.2.4
```

```
    optional: true

  '@img/sharp-wasm32@0.34.5':
    dependencies:
      '@emnapi/runtime': 1.11.3
    optional: true

  '@img/sharp-win32-arm64@0.34.5':
    optional: true

  '@img/sharp-win32-ia32@0.34.5':
    optional: true

  '@img/sharp-win32-x64@0.34.5':
    optional: true

  '@jridgewell/gen-mapping@0.3.13':
    dependencies:
      '@jridgewell/sourcemap-codec': 1.5.5
      '@jridgewell/trace-mapping': 0.3.31

  '@jridgewell/remapping@2.3.5':
    dependencies:
      '@jridgewell/gen-mapping': 0.3.13
      '@jridgewell/trace-mapping': 0.3.31

  '@jridgewell/resolve-uri@3.1.2': {}

  '@jridgewell/sourcemap-codec@1.5.5': {}

  '@jridgewell/trace-mapping@0.3.31':
    dependencies:
      '@jridgewell/resolve-uri': 3.1.2
      '@jridgewell/sourcemap-codec': 1.5.5

  '@napi-rs/wasm-runtime@1.2.3(@emnapi/core@1.10.0)(@emnapi/runtime@1.10.0)':
    dependencies:
      '@emnapi/core': 1.10.0
      '@emnapi/runtime': 1.10.0
      '@tybys/wasm-util': 0.10.3
    optional: true

  '@next/env@15.5.23': {}

  '@next/eslint-plugin-next@15.5.23':
    dependencies:
      fast-glob: 3.3.1

  '@next/swc-darwin-arm64@15.5.23':
    optional: true

  '@next/swc-darwin-x64@15.5.23':
    optional: true

  '@next/swc-linux-arm64-gnu@15.5.23':
    optional: true

  '@next/swc-linux-arm64-musl@15.5.23':
    optional: true

  '@next/swc-linux-x64-gnu@15.5.23':
    optional: true

  '@next/swc-linux-x64-musl@15.5.23':
    optional: true

  '@next/swc-win32-arm64-msvc@15.5.23':
    optional: true

  '@next/swc-win32-x64-msvc@15.5.23':
    optional: true

  '@nodelib/fs.scandir@2.1.5':
    dependencies:
      '@nodelib/fs.stat': 2.0.5
      run-parallel: 1.2.0

  '@nodelib/fs.stat@2.0.5': {}

  '@nodelib/fs.walk@1.2.8':
    dependencies:
      '@nodelib/fs.scandir': 2.1.5
      fastq: 1.20.1

  '@nolyfill/is-core-module@1.0.39': {}

  '@radix-ui/number@1.1.3': {}

  '@radix-ui/primitive@1.1.7': {}

  '@radix-ui/react-arrow@1.1.15(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-avatar@1.2.6(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@radix-ui/primitive': 1.1.7
      '@radix-ui/react-context': 1.2.2(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@1
```

```
9.2.8)
      '@radix-ui/react-use-callback-ref': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-is-hydrated': 0.1.3(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-layout-effect': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-collection@1.1.15(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8
)':
    dependencies:
      '@radix-ui/react-compose-refs': 1.1.5(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-context': 1.2.2(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@1
9.2.8)
      '@radix-ui/react-slot': 1.3.3(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-compose-refs@1.1.5(@types/react@19.2.18)(react@19.2.8)':
    dependencies:
      react: 19.2.8
    optionalDependencies:
      '@types/react': 19.2.18

  '@radix-ui/react-context@1.2.2(@types/react@19.2.18)(react@19.2.8)':
    dependencies:
      react: 19.2.8
    optionalDependencies:
      '@types/react': 19.2.18

  '@radix-ui/react-dialog@1.1.23(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@radix-ui/primitive': 1.1.7
      '@radix-ui/react-compose-refs': 1.1.5(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-context': 1.2.2(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-dismissable-layer': 1.1.19(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))
(react@19.2.8)
      '@radix-ui/react-focus-guards': 1.1.6(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-focus-scope': 1.1.16(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react
@19.2.8)
      '@radix-ui/react-id': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-portal': 1.1.17(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2
.8)
      '@radix-ui/react-presence': 1.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19
.2.8)
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@1
9.2.8)
      '@radix-ui/react-slot': 1.3.3(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-controllable-state': 1.2.6(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-layout-effect': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      aria-hidden: 1.2.6
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
      react-remove-scroll: 2.7.2(@types/react@19.2.18)(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-direction@1.1.4(@types/react@19.2.18)(react@19.2.8)':
    dependencies:
      react: 19.2.8
    optionalDependencies:
      '@types/react': 19.2.18

  '@radix-ui/react-dismissable-layer@1.1.19(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react
@19.2.8)':
    dependencies:
      '@radix-ui/primitive': 1.1.7
      '@radix-ui/react-compose-refs': 1.1.5(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@1
9.2.8)
      '@radix-ui/react-use-callback-ref': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-effect-event': 0.0.5(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-dropdown-menu@2.1.24(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.
2.8)':
    dependencies:
      '@radix-ui/primitive': 1.1.7
      '@radix-ui/react-compose-refs': 1.1.5(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-context': 1.2.2(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-id': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-menu': 2.1.24(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8
)
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@1
9.2.8)
      '@radix-ui/react-use-controllable-state': 1.2.6(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-focus-guards@1.1.6(@types/react@19.2.18)(react@19.2.8)':
    dependencies:
```

```
      react: 19.2.8
    optionalDependencies:
      '@types/react': 19.2.18

  '@radix-ui/react-focus-scope@1.1.16(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@radix-ui/react-compose-refs': 1.1.5(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-use-callback-ref': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-id@1.1.4(@types/react@19.2.18)(react@19.2.8)':
    dependencies:
      '@radix-ui/react-use-layout-effect': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
    optionalDependencies:
      '@types/react': 19.2.18

  '@radix-ui/react-menu@2.1.24(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@radix-ui/primitive': 1.1.7
      '@radix-ui/react-collection': 1.1.15(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-compose-refs': 1.1.5(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-context': 1.2.2(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-direction': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-dismissable-layer': 1.1.19(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-focus-guards': 1.1.6(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-focus-scope': 1.1.16(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-id': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-popper': 1.3.7(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-portal': 1.1.17(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-presence': 1.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-roving-focus': 1.1.19(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-slot': 1.3.3(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-callback-ref': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      aria-hidden: 1.2.6
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
      react-remove-scroll: 2.7.2(@types/react@19.2.18)(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-popper@1.3.7(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@floating-ui/react-dom': 2.1.9(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-arrow': 1.1.15(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-compose-refs': 1.1.5(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-context': 1.2.2(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-use-callback-ref': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-layout-effect': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-rect': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-size': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/rect': 1.1.3
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-portal@1.1.17(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-use-layout-effect': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-presence@1.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@radix-ui/react-use-layout-effect': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-primitive@2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@radix-ui/react-slot': 1.3.3(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
```

```
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-roving-focus@1.1.19(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@radix-ui/primitive': 1.1.7
      '@radix-ui/react-collection': 1.1.15(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-compose-refs': 1.1.5(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-context': 1.2.2(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-direction': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-id': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-use-callback-ref': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-controllable-state': 1.2.6(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-is-hydrated': 0.1.3(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-layout-effect': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-scroll-area@1.2.18(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@radix-ui/number': 1.1.3
      '@radix-ui/primitive': 1.1.7
      '@radix-ui/react-compose-refs': 1.1.5(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-context': 1.2.2(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-direction': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-presence': 1.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-use-callback-ref': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-layout-effect': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-slot@1.3.3(@types/react@19.2.18)(react@19.2.8)':
    dependencies:
      '@radix-ui/react-compose-refs': 1.1.5(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
    optionalDependencies:
      '@types/react': 19.2.18

  '@radix-ui/react-tabs@1.1.21(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@radix-ui/primitive': 1.1.7
      '@radix-ui/react-context': 1.2.2(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-direction': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-id': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-presence': 1.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-roving-focus': 1.1.19(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-use-controllable-state': 1.2.6(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-tooltip@1.2.16(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@radix-ui/primitive': 1.1.7
      '@radix-ui/react-compose-refs': 1.1.5(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-context': 1.2.2(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-dismissable-layer': 1.1.19(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-id': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-popper': 1.3.7(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-portal': 1.1.17(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-presence': 1.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      '@radix-ui/react-slot': 1.3.3(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-controllable-state': 1.2.6(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-layout-effect': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-visually-hidden': 1.2.11(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/react-use-callback-ref@1.1.4(@types/react@19.2.18)(react@19.2.8)':
    dependencies:
      react: 19.2.8
    optionalDependencies:
      '@types/react': 19.2.18

```

```yaml
  '@radix-ui/react-use-controllable-state@1.2.6(@types/react@19.2.18)(react@19.2.8)':
    dependencies:
      '@radix-ui/primitive': 1.1.7
      '@radix-ui/react-use-effect-event': 0.0.5(@types/react@19.2.18)(react@19.2.8)
      '@radix-ui/react-use-layout-effect': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
    optionalDependencies:
      '@types/react': 19.2.18

  '@radix-ui/react-use-effect-event@0.0.5(@types/react@19.2.18)(react@19.2.8)':
    dependencies:
      '@radix-ui/react-use-layout-effect': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
    optionalDependencies:
      '@types/react': 19.2.18

  '@radix-ui/react-use-is-hydrated@0.1.3(@types/react@19.2.18)(react@19.2.8)':
    dependencies:
      react: 19.2.8
    optionalDependencies:
      '@types/react': 19.2.18

  '@radix-ui/react-use-layout-effect@1.1.4(@types/react@19.2.18)(react@19.2.8)':
    dependencies:
      react: 19.2.8
    optionalDependencies:
      '@types/react': 19.2.18

  '@radix-ui/react-use-rect@1.1.4(@types/react@19.2.18)(react@19.2.8)':
    dependencies:
      '@radix-ui/rect': 1.1.3
      react: 19.2.8
    optionalDependencies:
      '@types/react': 19.2.18

  '@radix-ui/react-use-size@1.1.4(@types/react@19.2.18)(react@19.2.8)':
    dependencies:
      '@radix-ui/react-use-layout-effect': 1.1.4(@types/react@19.2.18)(react@19.2.8)
      react: 19.2.8
    optionalDependencies:
      '@types/react': 19.2.18

  '@radix-ui/react-visually-hidden@1.2.11(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)':
    dependencies:
      '@radix-ui/react-primitive': 2.1.10(@types/react-dom@19.2.4(@types/react@19.2.18))(@types/react@19.2.18)(react-dom@19.2.8(react@19.2.8))(react@19.2.8)
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18
      '@types/react-dom': 19.2.4(@types/react@19.2.18)

  '@radix-ui/rect@1.1.3': {}

  '@rtsao/scc@1.1.0': {}

  '@rushstack/eslint-patch@1.16.1': {}

  '@swc/helpers@0.5.15':
    dependencies:
      tslib: 2.8.1

  '@tailwindcss/node@4.3.3':
    dependencies:
      '@jridgewell/remapping': 2.3.5
      enhanced-resolve: 5.24.5
      jiti: 2.7.0
      lightningcss: 1.32.0
      magic-string: 0.30.21
      source-map-js: 1.2.1
      tailwindcss: 4.3.3

  '@tailwindcss/oxide-android-arm64@4.3.3':
    optional: true

  '@tailwindcss/oxide-darwin-arm64@4.3.3':
    optional: true

  '@tailwindcss/oxide-darwin-x64@4.3.3':
    optional: true

  '@tailwindcss/oxide-freebsd-x64@4.3.3':
    optional: true

  '@tailwindcss/oxide-linux-arm-gnueabihf@4.3.3':
    optional: true

  '@tailwindcss/oxide-linux-arm64-gnu@4.3.3':
    optional: true

  '@tailwindcss/oxide-linux-arm64-musl@4.3.3':
    optional: true

  '@tailwindcss/oxide-linux-x64-gnu@4.3.3':
    optional: true

  '@tailwindcss/oxide-linux-x64-musl@4.3.3':
    optional: true

  '@tailwindcss/oxide-wasm32-wasi@4.3.3':
    optional: true

  '@tailwindcss/oxide-win32-arm64-msvc@4.3.3':
    optional: true
```

```yaml

  '@tailwindcss/oxide-win32-x64-msvc@4.3.3':
    optional: true

  '@tailwindcss/oxide@4.3.3':
    optionalDependencies:
      '@tailwindcss/oxide-android-arm64': 4.3.3
      '@tailwindcss/oxide-darwin-arm64': 4.3.3
      '@tailwindcss/oxide-darwin-x64': 4.3.3
      '@tailwindcss/oxide-freebsd-x64': 4.3.3
      '@tailwindcss/oxide-linux-arm-gnueabihf': 4.3.3
      '@tailwindcss/oxide-linux-arm64-gnu': 4.3.3
      '@tailwindcss/oxide-linux-arm64-musl': 4.3.3
      '@tailwindcss/oxide-linux-x64-gnu': 4.3.3
      '@tailwindcss/oxide-linux-x64-musl': 4.3.3
      '@tailwindcss/oxide-wasm32-wasi': 4.3.3
      '@tailwindcss/oxide-win32-arm64-msvc': 4.3.3
      '@tailwindcss/oxide-win32-x64-msvc': 4.3.3

  '@tailwindcss/postcss@4.3.3':
    dependencies:
      '@alloc/quick-lru': 5.2.0
      '@tailwindcss/node': 4.3.3
      '@tailwindcss/oxide': 4.3.3
      postcss: 8.5.26
      tailwindcss: 4.3.3

  '@tybys/wasm-util@0.10.3':
    dependencies:
      tslib: 2.8.1
    optional: true

  '@types/debug@4.1.13':
    dependencies:
      '@types/ms': 2.1.0

  '@types/estree-jsx@1.0.5':
    dependencies:
      '@types/estree': 1.0.9

  '@types/estree@1.0.9': {}

  '@types/hast@3.0.5':
    dependencies:
      '@types/unist': 3.0.3

  '@types/json-schema@7.0.15': {}

  '@types/json5@0.0.29': {}

  '@types/mdast@4.0.4':
    dependencies:
      '@types/unist': 3.0.3

  '@types/ms@2.1.0': {}

  '@types/node@22.20.1':
    dependencies:
      undici-types: 6.21.0

  '@types/react-dom@19.2.4(@types/react@19.2.18)':
    dependencies:
      '@types/react': 19.2.18

  '@types/react@19.2.18':
    dependencies:
      csstype: 3.2.3

  '@types/unist@2.0.11': {}

  '@types/unist@3.0.3': {}

  '@typescript-eslint/eslint-plugin@8.67.0(@typescript-eslint/parser@8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3))(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3)':
    dependencies:
      '@eslint-community/regexpp': 4.12.2
      '@typescript-eslint/parser': 8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3)
      '@typescript-eslint/scope-manager': 8.67.0
      '@typescript-eslint/type-utils': 8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3)
      '@typescript-eslint/utils': 8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3)
      '@typescript-eslint/visitor-keys': 8.67.0
      eslint: 9.39.5(jiti@2.7.0)
      ignore: 7.0.6
      natural-compare: 1.4.0
      ts-api-utils: 2.5.0(typescript@5.9.3)
      typescript: 5.9.3
    transitivePeerDependencies:
      - supports-color

  '@typescript-eslint/parser@8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3)':
    dependencies:
      '@typescript-eslint/scope-manager': 8.67.0
      '@typescript-eslint/types': 8.67.0
      '@typescript-eslint/typescript-estree': 8.67.0(typescript@5.9.3)
      '@typescript-eslint/visitor-keys': 8.67.0
      debug: 4.4.3
      eslint: 9.39.5(jiti@2.7.0)
      typescript: 5.9.3
    transitivePeerDependencies:
      - supports-color

  '@typescript-eslint/project-service@8.67.0(typescript@5.9.3)':
    dependencies:
      '@typescript-eslint/tsconfig-utils': 8.67.0(typescript@5.9.3)
      '@typescript-eslint/types': 8.67.0
```

```
      debug: 4.4.3
      typescript: 5.9.3
    transitivePeerDependencies:
      - supports-color

  '@typescript-eslint/scope-manager@8.67.0':
    dependencies:
      '@typescript-eslint/types': 8.67.0
      '@typescript-eslint/visitor-keys': 8.67.0

  '@typescript-eslint/tsconfig-utils@8.67.0(typescript@5.9.3)':
    dependencies:
      typescript: 5.9.3

  '@typescript-eslint/type-utils@8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3)':
    dependencies:
      '@typescript-eslint/types': 8.67.0
      '@typescript-eslint/typescript-estree': 8.67.0(typescript@5.9.3)
      '@typescript-eslint/utils': 8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3)
      debug: 4.4.3
      eslint: 9.39.5(jiti@2.7.0)
      ts-api-utils: 2.5.0(typescript@5.9.3)
      typescript: 5.9.3
    transitivePeerDependencies:
      - supports-color

  '@typescript-eslint/types@8.67.0': {}

  '@typescript-eslint/typescript-estree@8.67.0(typescript@5.9.3)':
    dependencies:
      '@typescript-eslint/project-service': 8.67.0(typescript@5.9.3)
      '@typescript-eslint/tsconfig-utils': 8.67.0(typescript@5.9.3)
      '@typescript-eslint/types': 8.67.0
      '@typescript-eslint/visitor-keys': 8.67.0
      debug: 4.4.3
      minimatch: 10.2.6
      semver: 7.8.5
      tinyglobby: 0.2.17
      ts-api-utils: 2.5.0(typescript@5.9.3)
      typescript: 5.9.3
    transitivePeerDependencies:
      - supports-color

  '@typescript-eslint/utils@8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3)':
    dependencies:
      '@eslint-community/eslint-utils': 4.10.1(eslint@9.39.5(jiti@2.7.0))
      '@typescript-eslint/scope-manager': 8.67.0
      '@typescript-eslint/types': 8.67.0
      '@typescript-eslint/typescript-estree': 8.67.0(typescript@5.9.3)
      eslint: 9.39.5(jiti@2.7.0)
      typescript: 5.9.3
    transitivePeerDependencies:
      - supports-color

  '@typescript-eslint/visitor-keys@8.67.0':
    dependencies:
      '@typescript-eslint/types': 8.67.0
      eslint-visitor-keys: 5.0.1

  '@ungap/structured-clone@1.3.3': {}

  '@unrs/resolver-binding-android-arm-eabi@1.12.2':
    optional: true

  '@unrs/resolver-binding-android-arm64@1.12.2':
    optional: true

  '@unrs/resolver-binding-darwin-arm64@1.12.2':
    optional: true

  '@unrs/resolver-binding-darwin-x64@1.12.2':
    optional: true

  '@unrs/resolver-binding-freebsd-x64@1.12.2':
    optional: true

  '@unrs/resolver-binding-linux-arm-gnueabihf@1.12.2':
    optional: true

  '@unrs/resolver-binding-linux-arm-musleabihf@1.12.2':
    optional: true

  '@unrs/resolver-binding-linux-arm64-gnu@1.12.2':
    optional: true

  '@unrs/resolver-binding-linux-arm64-musl@1.12.2':
    optional: true

  '@unrs/resolver-binding-linux-loong64-gnu@1.12.2':
    optional: true

  '@unrs/resolver-binding-linux-loong64-musl@1.12.2':
    optional: true

  '@unrs/resolver-binding-linux-ppc64-gnu@1.12.2':
    optional: true

  '@unrs/resolver-binding-linux-riscv64-gnu@1.12.2':
    optional: true

  '@unrs/resolver-binding-linux-riscv64-musl@1.12.2':
    optional: true

  '@unrs/resolver-binding-linux-s390x-gnu@1.12.2':
    optional: true
```

```yaml

  '@unrs/resolver-binding-linux-x64-gnu@1.12.2':
    optional: true

  '@unrs/resolver-binding-linux-x64-musl@1.12.2':
    optional: true

  '@unrs/resolver-binding-openharmony-arm64@1.12.2':
    optional: true

  '@unrs/resolver-binding-wasm32-wasi@1.12.2':
    dependencies:
      '@emnapi/core': 1.10.0
      '@emnapi/runtime': 1.10.0
      '@napi-rs/wasm-runtime': 1.2.3(@emnapi/core@1.10.0)(@emnapi/runtime@1.10.0)
    optional: true

  '@unrs/resolver-binding-win32-arm64-msvc@1.12.2':
    optional: true

  '@unrs/resolver-binding-win32-ia32-msvc@1.12.2':
    optional: true

  '@unrs/resolver-binding-win32-x64-msvc@1.12.2':
    optional: true

  acorn-jsx@5.3.2(acorn@8.18.0):
    dependencies:
      acorn: 8.18.0

  acorn@8.18.0: {}

  ajv@6.15.0:
    dependencies:
      fast-deep-equal: 3.1.3
      fast-json-stable-stringify: 2.1.0
      json-schema-traverse: 0.4.1
      uri-js: 4.4.1

  ansi-styles@4.3.0:
    dependencies:
      color-convert: 2.0.1

  argparse@2.0.1: {}

  aria-hidden@1.2.6:
    dependencies:
      tslib: 2.8.1

  aria-query@5.3.2: {}

  array-buffer-byte-length@1.0.2:
    dependencies:
      call-bound: 1.0.4
      is-array-buffer: 3.0.5

  array-includes@3.1.9:
    dependencies:
      call-bind: 1.0.9
      call-bound: 1.0.4
      define-properties: 1.2.1
      es-abstract: 1.24.2
      es-object-atoms: 1.1.2
      get-intrinsic: 1.3.0
      is-string: 1.1.1
      math-intrinsics: 1.1.0

  array.prototype.findlast@1.2.5:
    dependencies:
      call-bind: 1.0.9
      define-properties: 1.2.1
      es-abstract: 1.24.2
      es-errors: 1.3.0
      es-object-atoms: 1.1.2
      es-shim-unscopables: 1.1.0

  array.prototype.findlastindex@1.2.6:
    dependencies:
      call-bind: 1.0.9
      call-bound: 1.0.4
      define-properties: 1.2.1
      es-abstract: 1.24.2
      es-errors: 1.3.0
      es-object-atoms: 1.1.2
      es-shim-unscopables: 1.1.0

  array.prototype.flat@1.3.3:
    dependencies:
      call-bind: 1.0.9
      define-properties: 1.2.1
      es-abstract: 1.24.2
      es-shim-unscopables: 1.1.0

  array.prototype.flatmap@1.3.3:
    dependencies:
      call-bind: 1.0.9
      define-properties: 1.2.1
      es-abstract: 1.24.2
      es-shim-unscopables: 1.1.0

  array.prototype.tosorted@1.1.4:
    dependencies:
      call-bind: 1.0.9
      define-properties: 1.2.1
      es-abstract: 1.24.2
```

```
      es-errors: 1.3.0
      es-shim-unscopables: 1.1.0

  arraybuffer.prototype.slice@1.0.4:
    dependencies:
      array-buffer-byte-length: 1.0.2
      call-bind: 1.0.9
      define-properties: 1.2.1
      es-abstract: 1.24.2
      es-errors: 1.3.0
      get-intrinsic: 1.3.0
      is-array-buffer: 3.0.5

  ast-types-flow@0.0.8: {}

  async-function@1.0.0: {}

  available-typed-arrays@1.0.7:
    dependencies:
      possible-typed-array-names: 1.1.0

  axe-core@4.13.0: {}

  axobject-query@4.1.0: {}

  bail@2.0.2: {}

  balanced-match@1.0.2: {}

  balanced-match@4.0.4: {}

  brace-expansion@1.1.18:
    dependencies:
      balanced-match: 1.0.2
      concat-map: 0.0.1

  brace-expansion@5.0.9:
    dependencies:
      balanced-match: 4.0.4

  braces@3.0.3:
    dependencies:
      fill-range: 7.1.1

  call-bind-apply-helpers@1.0.2:
    dependencies:
      es-errors: 1.3.0
      function-bind: 1.1.2

  call-bind@1.0.9:
    dependencies:
      call-bind-apply-helpers: 1.0.2
      es-define-property: 1.0.1
      get-intrinsic: 1.3.0
      set-function-length: 1.2.2

  call-bound@1.0.4:
    dependencies:
      call-bind-apply-helpers: 1.0.2
      get-intrinsic: 1.3.0

  callsites@3.1.0: {}

  caniuse-lite@1.0.30001809: {}

  ccount@2.0.1: {}

  chalk@4.1.2:
    dependencies:
      ansi-styles: 4.3.0
      supports-color: 7.2.0

  character-entities-html4@2.1.0: {}

  character-entities-legacy@3.0.0: {}

  character-entities@2.0.2: {}

  character-reference-invalid@2.0.1: {}

  class-variance-authority@0.7.1:
    dependencies:
      clsx: 2.1.1

  client-only@0.0.1: {}

  clsx@2.1.1: {}

  color-convert@2.0.1:
    dependencies:
      color-name: 1.1.4

  color-name@1.1.4: {}

  comma-separated-tokens@2.0.3: {}

  concat-map@0.0.1: {}

  cross-spawn@7.0.6:
    dependencies:
      path-key: 3.1.1
      shebang-command: 2.0.0
      which: 2.0.2

  csstype@3.2.3: {}
```

```yaml

  damerau-levenshtein@1.0.8: {}

  data-view-buffer@1.0.2:
    dependencies:
      call-bound: 1.0.4
      es-errors: 1.3.0
      is-data-view: 1.0.2

  data-view-byte-length@1.0.2:
    dependencies:
      call-bound: 1.0.4
      es-errors: 1.3.0
      is-data-view: 1.0.2

  data-view-byte-offset@1.0.1:
    dependencies:
      call-bound: 1.0.4
      es-errors: 1.3.0
      is-data-view: 1.0.2

  debug@3.2.7:
    dependencies:
      ms: 2.1.3

  debug@4.4.3:
    dependencies:
      ms: 2.1.3

  decode-named-character-reference@1.3.0:
    dependencies:
      character-entities: 2.0.2

  deep-is@0.1.4: {}

  define-data-property@1.1.4:
    dependencies:
      es-define-property: 1.0.1
      es-errors: 1.3.0
      gopd: 1.2.0

  define-properties@1.2.1:
    dependencies:
      define-data-property: 1.1.4
      has-property-descriptors: 1.0.2
      object-keys: 1.1.1

  dequal@2.0.3: {}

  detect-libc@2.1.2: {}

  detect-node-es@1.1.0: {}

  devlop@1.1.0:
    dependencies:
      dequal: 2.0.3

  doctrine@2.1.0:
    dependencies:
      esutils: 2.0.3

  dunder-proto@1.0.1:
    dependencies:
      call-bind-apply-helpers: 1.0.2
      es-errors: 1.3.0
      gopd: 1.2.0

  emoji-regex@9.2.2: {}

  enhanced-resolve@5.24.5:
    dependencies:
      graceful-fs: 4.2.11
      tapable: 2.3.3

  entities@6.0.1: {}

  es-abstract-get@1.0.0:
    dependencies:
      es-errors: 1.3.0
      es-object-atoms: 1.1.2
      is-callable: 1.2.7
      object-inspect: 1.13.4

  es-abstract@1.24.2:
    dependencies:
      array-buffer-byte-length: 1.0.2
      arraybuffer.prototype.slice: 1.0.4
      available-typed-arrays: 1.0.7
      call-bind: 1.0.9
      call-bound: 1.0.4
      data-view-buffer: 1.0.2
      data-view-byte-length: 1.0.2
      data-view-byte-offset: 1.0.1
      es-define-property: 1.0.1
      es-errors: 1.3.0
      es-object-atoms: 1.1.2
      es-set-tostringtag: 2.1.0
      es-to-primitive: 1.3.4
      function.prototype.name: 1.2.0
      get-intrinsic: 1.3.0
      get-proto: 1.0.1
      get-symbol-description: 1.1.0
      globalthis: 1.0.4
      gopd: 1.2.0
      has-property-descriptors: 1.0.2
```

```yaml
      has-proto: 1.2.0
      has-symbols: 1.1.0
      hasown: 2.0.4
      internal-slot: 1.1.0
      is-array-buffer: 3.0.5
      is-callable: 1.2.7
      is-data-view: 1.0.2
      is-negative-zero: 2.0.3
      is-regex: 1.2.1
      is-set: 2.0.3
      is-shared-array-buffer: 1.0.4
      is-string: 1.1.1
      is-typed-array: 1.1.15
      is-weakref: 1.1.1
      math-intrinsics: 1.1.0
      object-inspect: 1.13.4
      object-keys: 1.1.1
      object.assign: 4.1.7
      own-keys: 1.0.2
      regexp.prototype.flags: 1.5.4
      safe-array-concat: 1.1.4
      safe-push-apply: 1.0.0
      safe-regex-test: 1.1.0
      set-proto: 1.0.0
      stop-iteration-iterator: 1.1.0
      string.prototype.trim: 1.2.11
      string.prototype.trimend: 1.0.10
      string.prototype.trimstart: 1.0.8
      typed-array-buffer: 1.0.3
      typed-array-byte-length: 1.0.3
      typed-array-byte-offset: 1.0.4
      typed-array-length: 1.0.8
      unbox-primitive: 1.1.0
      which-typed-array: 1.1.22

  es-define-property@1.0.1: {}

  es-errors@1.3.0: {}

  es-iterator-helpers@1.4.0:
    dependencies:
      call-bind: 1.0.9
      call-bound: 1.0.4
      define-properties: 1.2.1
      es-abstract: 1.24.2
      es-errors: 1.3.0
      es-set-tostringtag: 2.1.0
      function-bind: 1.1.2
      get-intrinsic: 1.3.0
      globalthis: 1.0.4
      gopd: 1.2.0
      has-property-descriptors: 1.0.2
      has-proto: 1.2.0
      has-symbols: 1.1.0
      internal-slot: 1.1.0
      iterator.prototype: 1.1.5
      math-intrinsics: 1.1.0

  es-object-atoms@1.1.2:
    dependencies:
      es-errors: 1.3.0

  es-set-tostringtag@2.1.0:
    dependencies:
      es-errors: 1.3.0
      get-intrinsic: 1.3.0
      has-tostringtag: 1.0.2
      hasown: 2.0.4

  es-shim-unscopables@1.1.0:
    dependencies:
      hasown: 2.0.4

  es-to-primitive@1.3.4:
    dependencies:
      es-abstract-get: 1.0.0
      es-define-property: 1.0.1
      es-errors: 1.3.0
      is-callable: 1.2.7
      is-date-object: 1.1.0
      is-symbol: 1.1.1

  escape-string-regexp@4.0.0: {}

  escape-string-regexp@5.0.0: {}

  eslint-config-next@15.5.23(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3):
    dependencies:
      '@next/eslint-plugin-next': 15.5.23
      '@rushstack/eslint-patch': 1.16.1
      '@typescript-eslint/eslint-plugin': 8.67.0(@typescript-eslint/parser@8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3))(eslint@9.39.5(jiti@2.7
.0))(typescript@5.9.3)
      '@typescript-eslint/parser': 8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3)
      eslint: 9.39.5(jiti@2.7.0)
      eslint-import-resolver-node: 0.3.10
      eslint-import-resolver-typescript: 3.10.1(eslint-plugin-import@2.32.0)(eslint@9.39.5(jiti@2.7.0))
      eslint-plugin-import: 2.32.0(@typescript-eslint/parser@8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3))(eslint-import-resolver-typescript@3.
10.1)(eslint@9.39.5(jiti@2.7.0))
      eslint-plugin-jsx-a11y: 6.10.2(eslint@9.39.5(jiti@2.7.0))
      eslint-plugin-react: 7.37.5(eslint@9.39.5(jiti@2.7.0))
      eslint-plugin-react-hooks: 5.2.0(eslint@9.39.5(jiti@2.7.0))
    optionalDependencies:
      typescript: 5.9.3
    transitivePeerDependencies:
      - eslint-import-resolver-webpack
```

```
      - eslint-plugin-import-x
      - supports-color

  eslint-import-resolver-node@0.3.10:
    dependencies:
      debug: 3.2.7
      is-core-module: 2.16.2
      resolve: 2.0.0-next.7
    transitivePeerDependencies:
      - supports-color

  eslint-import-resolver-typescript@3.10.1(eslint-plugin-import@2.32.0)(eslint@9.39.5(jiti@2.7.0)):
    dependencies:
      '@nolyfill/is-core-module': 1.0.39
      debug: 4.4.3
      eslint: 9.39.5(jiti@2.7.0)
      get-tsconfig: 4.14.2
      is-bun-module: 2.0.0
      stable-hash: 0.0.5
      tinyglobby: 0.2.17
      unrs-resolver: 1.12.2
    optionalDependencies:
      eslint-plugin-import: 2.32.0(@typescript-eslint/parser@8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3))(eslint-import-resolver-typescript@3.10.1)(eslint@9.39.5(jiti@2.7.0))
    transitivePeerDependencies:
      - supports-color

  eslint-module-utils@2.14.0(@typescript-eslint/parser@8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3))(eslint-import-resolver-node@0.3.10)(eslint-import-resolver-typescript@3.10.1)(eslint@9.39.5(jiti@2.7.0)):
    dependencies:
      debug: 3.2.7
    optionalDependencies:
      '@typescript-eslint/parser': 8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3)
      eslint: 9.39.5(jiti@2.7.0)
      eslint-import-resolver-node: 0.3.10
      eslint-import-resolver-typescript: 3.10.1(eslint-plugin-import@2.32.0)(eslint@9.39.5(jiti@2.7.0))
    transitivePeerDependencies:
      - supports-color

  eslint-plugin-import@2.32.0(@typescript-eslint/parser@8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3))(eslint-import-resolver-typescript@3.10.1)(eslint@9.39.5(jiti@2.7.0)):
    dependencies:
      '@rtsao/scc': 1.1.0
      array-includes: 3.1.9
      array.prototype.findlastindex: 1.2.6
      array.prototype.flat: 1.3.3
      array.prototype.flatmap: 1.3.3
      debug: 3.2.7
      doctrine: 2.1.0
      eslint: 9.39.5(jiti@2.7.0)
      eslint-import-resolver-node: 0.3.10
      eslint-module-utils: 2.14.0(@typescript-eslint/parser@8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3))(eslint-import-resolver-node@0.3.10)(eslint-import-resolver-typescript@3.10.1)(eslint@9.39.5(jiti@2.7.0))
      hasown: 2.0.4
      is-core-module: 2.16.2
      is-glob: 4.0.3
      minimatch: 3.1.5
      object.fromentries: 2.0.8
      object.groupby: 1.0.3
      object.values: 1.2.1
      semver: 6.3.1
      string.prototype.trimend: 1.0.10
      tsconfig-paths: 3.15.0
    optionalDependencies:
      '@typescript-eslint/parser': 8.67.0(eslint@9.39.5(jiti@2.7.0))(typescript@5.9.3)
    transitivePeerDependencies:
      - eslint-import-resolver-typescript
      - eslint-import-resolver-webpack
      - supports-color

  eslint-plugin-jsx-a11y@6.10.2(eslint@9.39.5(jiti@2.7.0)):
    dependencies:
      aria-query: 5.3.2
      array-includes: 3.1.9
      array.prototype.flatmap: 1.3.3
      ast-types-flow: 0.0.8
      axe-core: 4.13.0
      axobject-query: 4.1.0
      damerau-levenshtein: 1.0.8
      emoji-regex: 9.2.2
      eslint: 9.39.5(jiti@2.7.0)
      hasown: 2.0.4
      jsx-ast-utils: 3.3.5
      language-tags: 1.0.9
      minimatch: 3.1.5
      object.fromentries: 2.0.8
      safe-regex-test: 1.1.0
      string.prototype.includes: 2.0.1

  eslint-plugin-react-hooks@5.2.0(eslint@9.39.5(jiti@2.7.0)):
    dependencies:
      eslint: 9.39.5(jiti@2.7.0)

  eslint-plugin-react@7.37.5(eslint@9.39.5(jiti@2.7.0)):
    dependencies:
      array-includes: 3.1.9
      array.prototype.findlast: 1.2.5
      array.prototype.flatmap: 1.3.3
      array.prototype.tosorted: 1.1.4
      doctrine: 2.1.0
      es-iterator-helpers: 1.4.0
      eslint: 9.39.5(jiti@2.7.0)
      estraverse: 5.3.0
      hasown: 2.0.4
      jsx-ast-utils: 3.3.5
```

```
      minimatch: 3.1.5
      object.entries: 1.1.9
      object.fromentries: 2.0.8
      object.values: 1.2.1
      prop-types: 15.8.1
      resolve: 2.0.0-next.7
      semver: 6.3.1
      string.prototype.matchall: 4.0.12
      string.prototype.repeat: 1.0.0

  eslint-scope@8.4.0:
    dependencies:
      esrecurse: 4.3.0
      estraverse: 5.3.0

  eslint-visitor-keys@3.4.3: {}

  eslint-visitor-keys@4.2.1: {}

  eslint-visitor-keys@5.0.1: {}

  eslint@9.39.5(jiti@2.7.0):
    dependencies:
      '@eslint-community/eslint-utils': 4.10.1(eslint@9.39.5(jiti@2.7.0))
      '@eslint-community/regexpp': 4.12.2
      '@eslint/config-array': 0.21.2
      '@eslint/config-helpers': 0.4.2
      '@eslint/core': 0.17.0
      '@eslint/eslintrc': 3.3.6
      '@eslint/js': 9.39.5
      '@eslint/plugin-kit': 0.4.1
      '@humanfs/node': 0.16.8
      '@humanwhocodes/module-importer': 1.0.1
      '@humanwhocodes/retry': 0.4.3
      '@types/estree': 1.0.9
      ajv: 6.15.0
      chalk: 4.1.2
      cross-spawn: 7.0.6
      debug: 4.4.3
      escape-string-regexp: 4.0.0
      eslint-scope: 8.4.0
      eslint-visitor-keys: 4.2.1
      espree: 10.4.0
      esquery: 1.7.0
      esutils: 2.0.3
      fast-deep-equal: 3.1.3
      file-entry-cache: 8.0.0
      find-up: 5.0.0
      glob-parent: 6.0.2
      ignore: 5.3.2
      imurmurhash: 0.1.4
      is-glob: 4.0.3
      json-stable-stringify-without-jsonify: 1.0.1
      lodash.merge: 4.6.2
      minimatch: 3.1.5
      natural-compare: 1.4.0
      optionator: 0.9.4
    optionalDependencies:
      jiti: 2.7.0
    transitivePeerDependencies:
      - supports-color

  espree@10.4.0:
    dependencies:
      acorn: 8.18.0
      acorn-jsx: 5.3.2(acorn@8.18.0)
      eslint-visitor-keys: 4.2.1

  esquery@1.7.0:
    dependencies:
      estraverse: 5.3.0

  esrecurse@4.3.0:
    dependencies:
      estraverse: 5.3.0

  estraverse@5.3.0: {}

  estree-util-is-identifier-name@3.0.0: {}

  esutils@2.0.3: {}

  extend@3.0.2: {}

  fast-deep-equal@3.1.3: {}

  fast-glob@3.3.1:
    dependencies:
      '@nodelib/fs.stat': 2.0.5
      '@nodelib/fs.walk': 1.2.8
      glob-parent: 5.1.2
      merge2: 1.4.1
      micromatch: 4.0.8

  fast-json-stable-stringify@2.1.0: {}

  fast-levenshtein@2.0.6: {}

  fastq@1.20.1:
    dependencies:
      reusify: 1.1.0

  fdir@6.5.0(picomatch@4.0.5):
    optionalDependencies:
      picomatch: 4.0.5
```

```yaml

  file-entry-cache@8.0.0:
    dependencies:
      flat-cache: 4.0.1

  fill-range@7.1.1:
    dependencies:
      to-regex-range: 5.0.1

  find-up@5.0.0:
    dependencies:
      locate-path: 6.0.0
      path-exists: 4.0.0

  flat-cache@4.0.1:
    dependencies:
      flatted: 3.4.4
      keyv: 4.5.4

  flatted@3.4.4: {}

  for-each@0.3.5:
    dependencies:
      is-callable: 1.2.7

  function-bind@1.1.2: {}

  function.prototype.name@1.2.0:
    dependencies:
      call-bind: 1.0.9
      call-bound: 1.0.4
      es-define-property: 1.0.1
      es-errors: 1.3.0
      functions-have-names: 1.2.3
      has-property-descriptors: 1.0.2
      hasown: 2.0.4
      is-callable: 1.2.7
      is-document.all: 1.0.0

  functions-have-names@1.2.3: {}

  generator-function@2.0.1: {}

  get-intrinsic@1.3.0:
    dependencies:
      call-bind-apply-helpers: 1.0.2
      es-define-property: 1.0.1
      es-errors: 1.3.0
      es-object-atoms: 1.1.2
      function-bind: 1.1.2
      get-proto: 1.0.1
      gopd: 1.2.0
      has-symbols: 1.1.0
      hasown: 2.0.4
      math-intrinsics: 1.1.0

  get-nonce@1.0.1: {}

  get-proto@1.0.1:
    dependencies:
      dunder-proto: 1.0.1
      es-object-atoms: 1.1.2

  get-symbol-description@1.1.0:
    dependencies:
      call-bound: 1.0.4
      es-errors: 1.3.0
      get-intrinsic: 1.3.0

  get-tsconfig@4.14.2:
    dependencies:
      resolve-pkg-maps: 1.0.0

  glob-parent@5.1.2:
    dependencies:
      is-glob: 4.0.3

  glob-parent@6.0.2:
    dependencies:
      is-glob: 4.0.3

  globals@14.0.0: {}

  globalthis@1.0.4:
    dependencies:
      define-properties: 1.2.1
      gopd: 1.2.0

  gopd@1.2.0: {}

  graceful-fs@4.2.11: {}

  has-bigints@1.1.0: {}

  has-flag@4.0.0: {}

  has-property-descriptors@1.0.2:
    dependencies:
      es-define-property: 1.0.1

  has-proto@1.2.0:
    dependencies:
      dunder-proto: 1.0.1

  has-symbols@1.1.0: {}
```

```yaml

  has-tostringtag@1.0.2:
    dependencies:
      has-symbols: 1.1.0

  hasown@2.0.4:
    dependencies:
      function-bind: 1.1.2

  hast-util-from-parse5@8.0.3:
    dependencies:
      '@types/hast': 3.0.5
      '@types/unist': 3.0.3
      devlop: 1.1.0
      hastscript: 9.0.1
      property-information: 7.2.0
      vfile: 6.0.3
      vfile-location: 5.0.3
      web-namespaces: 2.0.1

  hast-util-parse-selector@4.0.0:
    dependencies:
      '@types/hast': 3.0.5

  hast-util-raw@9.1.0:
    dependencies:
      '@types/hast': 3.0.5
      '@types/unist': 3.0.3
      '@ungap/structured-clone': 1.3.3
      hast-util-from-parse5: 8.0.3
      hast-util-to-parse5: 8.0.1
      html-void-elements: 3.0.0
      mdast-util-to-hast: 13.2.1
      parse5: 7.3.0
      unist-util-position: 5.0.0
      unist-util-visit: 5.1.0
      vfile: 6.0.3
      web-namespaces: 2.0.1
      zwitch: 2.0.4

  hast-util-sanitize@5.0.2:
    dependencies:
      '@types/hast': 3.0.5
      '@ungap/structured-clone': 1.3.3
      unist-util-position: 5.0.0

  hast-util-to-jsx-runtime@2.3.6:
    dependencies:
      '@types/estree': 1.0.9
      '@types/hast': 3.0.5
      '@types/unist': 3.0.3
      comma-separated-tokens: 2.0.3
      devlop: 1.1.0
      estree-util-is-identifier-name: 3.0.0
      hast-util-whitespace: 3.0.0
      mdast-util-mdx-expression: 2.0.1
      mdast-util-mdx-jsx: 3.2.0
      mdast-util-mdxjs-esm: 2.0.1
      property-information: 7.2.0
      space-separated-tokens: 2.0.2
      style-to-js: 1.1.21
      unist-util-position: 5.0.0
      vfile-message: 4.0.3
    transitivePeerDependencies:
      - supports-color

  hast-util-to-parse5@8.0.1:
    dependencies:
      '@types/hast': 3.0.5
      comma-separated-tokens: 2.0.3
      devlop: 1.1.0
      property-information: 7.2.0
      space-separated-tokens: 2.0.2
      web-namespaces: 2.0.1
      zwitch: 2.0.4

  hast-util-whitespace@3.0.0:
    dependencies:
      '@types/hast': 3.0.5

  hastscript@9.0.1:
    dependencies:
      '@types/hast': 3.0.5
      comma-separated-tokens: 2.0.3
      hast-util-parse-selector: 4.0.0
      property-information: 7.2.0
      space-separated-tokens: 2.0.2

  html-url-attributes@3.0.1: {}

  html-void-elements@3.0.0: {}

  ignore@5.3.2: {}

  ignore@7.0.6: {}

  import-fresh@3.3.1:
    dependencies:
      parent-module: 1.0.1
      resolve-from: 4.0.0

  imurmurhash@0.1.4: {}

  inline-style-parser@0.2.7: {}

```

```
  internal-slot@1.1.0:
    dependencies:
      es-errors: 1.3.0
      hasown: 2.0.4
      side-channel: 1.1.1

  is-alphabetical@2.0.1: {}

  is-alphanumerical@2.0.1:
    dependencies:
      is-alphabetical: 2.0.1
      is-decimal: 2.0.1

  is-array-buffer@3.0.5:
    dependencies:
      call-bind: 1.0.9
      call-bound: 1.0.4
      get-intrinsic: 1.3.0

  is-async-function@2.1.1:
    dependencies:
      async-function: 1.0.0
      call-bound: 1.0.4
      get-proto: 1.0.1
      has-tostringtag: 1.0.2
      safe-regex-test: 1.1.0

  is-bigint@1.1.0:
    dependencies:
      has-bigints: 1.1.0

  is-boolean-object@1.2.2:
    dependencies:
      call-bound: 1.0.4
      has-tostringtag: 1.0.2

  is-bun-module@2.0.0:
    dependencies:
      semver: 7.8.5

  is-callable@1.2.7: {}

  is-core-module@2.16.2:
    dependencies:
      hasown: 2.0.4

  is-data-view@1.0.2:
    dependencies:
      call-bound: 1.0.4
      get-intrinsic: 1.3.0
      is-typed-array: 1.1.15

  is-date-object@1.1.0:
    dependencies:
      call-bound: 1.0.4
      has-tostringtag: 1.0.2

  is-decimal@2.0.1: {}

  is-document.all@1.0.0:
    dependencies:
      call-bound: 1.0.4

  is-extglob@2.1.1: {}

  is-finalizationregistry@1.1.1:
    dependencies:
      call-bound: 1.0.4

  is-generator-function@1.1.2:
    dependencies:
      call-bound: 1.0.4
      generator-function: 2.0.1
      get-proto: 1.0.1
      has-tostringtag: 1.0.2
      safe-regex-test: 1.1.0

  is-glob@4.0.3:
    dependencies:
      is-extglob: 2.1.1

  is-hexadecimal@2.0.1: {}

  is-map@2.0.3: {}

  is-negative-zero@2.0.3: {}

  is-number-object@1.1.1:
    dependencies:
      call-bound: 1.0.4
      has-tostringtag: 1.0.2

  is-number@7.0.0: {}

  is-plain-obj@4.1.0: {}

  is-regex@1.2.1:
    dependencies:
      call-bound: 1.0.4
      gopd: 1.2.0
      has-tostringtag: 1.0.2
      hasown: 2.0.4

  is-set@2.0.3: {}

```

```
  is-shared-array-buffer@1.0.4:
    dependencies:
      call-bound: 1.0.4

  is-string@1.1.1:
    dependencies:
      call-bound: 1.0.4
      has-tostringtag: 1.0.2

  is-symbol@1.1.1:
    dependencies:
      call-bound: 1.0.4
      has-symbols: 1.1.0
      safe-regex-test: 1.1.0

  is-typed-array@1.1.15:
    dependencies:
      which-typed-array: 1.1.22

  is-weakmap@2.0.2: {}

  is-weakref@1.1.1:
    dependencies:
      call-bound: 1.0.4

  is-weakset@2.0.4:
    dependencies:
      call-bound: 1.0.4
      get-intrinsic: 1.3.0

  isarray@2.0.5: {}

  isexe@2.0.0: {}

  iterator.prototype@1.1.5:
    dependencies:
      define-data-property: 1.1.4
      es-object-atoms: 1.1.2
      get-intrinsic: 1.3.0
      get-proto: 1.0.1
      has-symbols: 1.1.0
      set-function-name: 2.0.2

  jiti@2.7.0: {}

  js-tokens@4.0.0: {}

  js-yaml@4.3.1:
    dependencies:
      argparse: 2.0.1

  json-buffer@3.0.1: {}

  json-schema-traverse@0.4.1: {}

  json-stable-stringify-without-jsonify@1.0.1: {}

  json5@1.0.2:
    dependencies:
      minimist: 1.2.8

  jsx-ast-utils@3.3.5:
    dependencies:
      array-includes: 3.1.9
      array.prototype.flat: 1.3.3
      object.assign: 4.1.7
      object.values: 1.2.1

  keyv@4.5.4:
    dependencies:
      json-buffer: 3.0.1

  language-subtag-registry@0.3.23: {}

  language-tags@1.0.9:
    dependencies:
      language-subtag-registry: 0.3.23

  levn@0.4.1:
    dependencies:
      prelude-ls: 1.2.1
      type-check: 0.4.0

  lightningcss-android-arm64@1.32.0:
    optional: true

  lightningcss-darwin-arm64@1.32.0:
    optional: true

  lightningcss-darwin-x64@1.32.0:
    optional: true

  lightningcss-freebsd-x64@1.32.0:
    optional: true

  lightningcss-linux-arm-gnueabihf@1.32.0:
    optional: true

  lightningcss-linux-arm64-gnu@1.32.0:
    optional: true

  lightningcss-linux-arm64-musl@1.32.0:
    optional: true

  lightningcss-linux-x64-gnu@1.32.0:
```

```yaml
    optional: true

  lightningcss-linux-x64-musl@1.32.0:
    optional: true

  lightningcss-win32-arm64-msvc@1.32.0:
    optional: true

  lightningcss-win32-x64-msvc@1.32.0:
    optional: true

  lightningcss@1.32.0:
    dependencies:
      detect-libc: 2.1.2
    optionalDependencies:
      lightningcss-android-arm64: 1.32.0
      lightningcss-darwin-arm64: 1.32.0
      lightningcss-darwin-x64: 1.32.0
      lightningcss-freebsd-x64: 1.32.0
      lightningcss-linux-arm-gnueabihf: 1.32.0
      lightningcss-linux-arm64-gnu: 1.32.0
      lightningcss-linux-arm64-musl: 1.32.0
      lightningcss-linux-x64-gnu: 1.32.0
      lightningcss-linux-x64-musl: 1.32.0
      lightningcss-win32-arm64-msvc: 1.32.0
      lightningcss-win32-x64-msvc: 1.32.0

  locate-path@6.0.0:
    dependencies:
      p-locate: 5.0.0

  lodash.merge@4.6.2: {}

  longest-streak@3.1.0: {}

  loose-envify@1.4.0:
    dependencies:
      js-tokens: 4.0.0

  lucide-react@0.468.0(react@19.2.8):
    dependencies:
      react: 19.2.8

  magic-string@0.30.21:
    dependencies:
      '@jridgewell/sourcemap-codec': 1.5.5

  markdown-table@3.0.4: {}

  math-intrinsics@1.1.0: {}

  mdast-util-find-and-replace@3.0.2:
    dependencies:
      '@types/mdast': 4.0.4
      escape-string-regexp: 5.0.0
      unist-util-is: 6.0.1
      unist-util-visit-parents: 6.0.2

  mdast-util-from-markdown@2.0.3:
    dependencies:
      '@types/mdast': 4.0.4
      '@types/unist': 3.0.3
      decode-named-character-reference: 1.3.0
      devlop: 1.1.0
      mdast-util-to-string: 4.0.0
      micromark: 4.0.2
      micromark-util-decode-numeric-character-reference: 2.0.2
      micromark-util-decode-string: 2.0.1
      micromark-util-normalize-identifier: 2.0.1
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2
      unist-util-stringify-position: 4.0.0
    transitivePeerDependencies:
      - supports-color

  mdast-util-gfm-autolink-literal@2.0.1:
    dependencies:
      '@types/mdast': 4.0.4
      ccount: 2.0.1
      devlop: 1.1.0
      mdast-util-find-and-replace: 3.0.2
      micromark-util-character: 2.1.1

  mdast-util-gfm-footnote@2.1.0:
    dependencies:
      '@types/mdast': 4.0.4
      devlop: 1.1.0
      mdast-util-from-markdown: 2.0.3
      mdast-util-to-markdown: 2.1.2
      micromark-util-normalize-identifier: 2.0.1
    transitivePeerDependencies:
      - supports-color

  mdast-util-gfm-strikethrough@2.0.0:
    dependencies:
      '@types/mdast': 4.0.4
      mdast-util-from-markdown: 2.0.3
      mdast-util-to-markdown: 2.1.2
    transitivePeerDependencies:
      - supports-color

  mdast-util-gfm-table@2.0.0:
    dependencies:
      '@types/mdast': 4.0.4
      devlop: 1.1.0
```

```
      markdown-table: 3.0.4
      mdast-util-from-markdown: 2.0.3
      mdast-util-to-markdown: 2.1.2
    transitivePeerDependencies:
      - supports-color

  mdast-util-gfm-task-list-item@2.0.0:
    dependencies:
      '@types/mdast': 4.0.4
      devlop: 1.1.0
      mdast-util-from-markdown: 2.0.3
      mdast-util-to-markdown: 2.1.2
    transitivePeerDependencies:
      - supports-color

  mdast-util-gfm@3.1.0:
    dependencies:
      mdast-util-from-markdown: 2.0.3
      mdast-util-gfm-autolink-literal: 2.0.1
      mdast-util-gfm-footnote: 2.1.0
      mdast-util-gfm-strikethrough: 2.0.0
      mdast-util-gfm-table: 2.0.0
      mdast-util-gfm-task-list-item: 2.0.0
      mdast-util-to-markdown: 2.1.2
    transitivePeerDependencies:
      - supports-color

  mdast-util-mdx-expression@2.0.1:
    dependencies:
      '@types/estree-jsx': 1.0.5
      '@types/hast': 3.0.5
      '@types/mdast': 4.0.4
      devlop: 1.1.0
      mdast-util-from-markdown: 2.0.3
      mdast-util-to-markdown: 2.1.2
    transitivePeerDependencies:
      - supports-color

  mdast-util-mdx-jsx@3.2.0:
    dependencies:
      '@types/estree-jsx': 1.0.5
      '@types/hast': 3.0.5
      '@types/mdast': 4.0.4
      '@types/unist': 3.0.3
      ccount: 2.0.1
      devlop: 1.1.0
      mdast-util-from-markdown: 2.0.3
      mdast-util-to-markdown: 2.1.2
      parse-entities: 4.0.2
      stringify-entities: 4.0.4
      unist-util-stringify-position: 4.0.0
      vfile-message: 4.0.3
    transitivePeerDependencies:
      - supports-color

  mdast-util-mdxjs-esm@2.0.1:
    dependencies:
      '@types/estree-jsx': 1.0.5
      '@types/hast': 3.0.5
      '@types/mdast': 4.0.4
      devlop: 1.1.0
      mdast-util-from-markdown: 2.0.3
      mdast-util-to-markdown: 2.1.2
    transitivePeerDependencies:
      - supports-color

  mdast-util-phrasing@4.1.0:
    dependencies:
      '@types/mdast': 4.0.4
      unist-util-is: 6.0.1

  mdast-util-to-hast@13.2.1:
    dependencies:
      '@types/hast': 3.0.5
      '@types/mdast': 4.0.4
      '@ungap/structured-clone': 1.3.3
      devlop: 1.1.0
      micromark-util-sanitize-uri: 2.0.1
      trim-lines: 3.0.1
      unist-util-position: 5.0.0
      unist-util-visit: 5.1.0
      vfile: 6.0.3

  mdast-util-to-markdown@2.1.2:
    dependencies:
      '@types/mdast': 4.0.4
      '@types/unist': 3.0.3
      longest-streak: 3.1.0
      mdast-util-phrasing: 4.1.0
      mdast-util-to-string: 4.0.0
      micromark-util-classify-character: 2.0.1
      micromark-util-decode-string: 2.0.1
      unist-util-visit: 5.1.0
      zwitch: 2.0.4

  mdast-util-to-string@4.0.0:
    dependencies:
      '@types/mdast': 4.0.4

  merge2@1.4.1: {}

  micromark-core-commonmark@2.0.3:
    dependencies:
      decode-named-character-reference: 1.3.0
      devlop: 1.1.0
```

```yaml
      micromark-factory-destination: 2.0.1
      micromark-factory-label: 2.0.1
      micromark-factory-space: 2.0.1
      micromark-factory-title: 2.0.1
      micromark-factory-whitespace: 2.0.1
      micromark-util-character: 2.1.1
      micromark-util-chunked: 2.0.1
      micromark-util-classify-character: 2.0.1
      micromark-util-html-tag-name: 2.0.1
      micromark-util-normalize-identifier: 2.0.1
      micromark-util-resolve-all: 2.0.1
      micromark-util-subtokenize: 2.1.0
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2

  micromark-extension-gfm-autolink-literal@2.1.0:
    dependencies:
      micromark-util-character: 2.1.1
      micromark-util-sanitize-uri: 2.0.1
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2

  micromark-extension-gfm-footnote@2.1.0:
    dependencies:
      devlop: 1.1.0
      micromark-core-commonmark: 2.0.3
      micromark-factory-space: 2.0.1
      micromark-util-character: 2.1.1
      micromark-util-normalize-identifier: 2.0.1
      micromark-util-sanitize-uri: 2.0.1
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2

  micromark-extension-gfm-strikethrough@2.1.0:
    dependencies:
      devlop: 1.1.0
      micromark-util-chunked: 2.0.1
      micromark-util-classify-character: 2.0.1
      micromark-util-resolve-all: 2.0.1
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2

  micromark-extension-gfm-table@2.1.1:
    dependencies:
      devlop: 1.1.0
      micromark-factory-space: 2.0.1
      micromark-util-character: 2.1.1
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2

  micromark-extension-gfm-tagfilter@2.0.0:
    dependencies:
      micromark-util-types: 2.0.2

  micromark-extension-gfm-task-list-item@2.1.0:
    dependencies:
      devlop: 1.1.0
      micromark-factory-space: 2.0.1
      micromark-util-character: 2.1.1
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2

  micromark-extension-gfm@3.0.0:
    dependencies:
      micromark-extension-gfm-autolink-literal: 2.1.0
      micromark-extension-gfm-footnote: 2.1.0
      micromark-extension-gfm-strikethrough: 2.1.0
      micromark-extension-gfm-table: 2.1.1
      micromark-extension-gfm-tagfilter: 2.0.0
      micromark-extension-gfm-task-list-item: 2.1.0
      micromark-util-combine-extensions: 2.0.1
      micromark-util-types: 2.0.2

  micromark-factory-destination@2.0.1:
    dependencies:
      micromark-util-character: 2.1.1
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2

  micromark-factory-label@2.0.1:
    dependencies:
      devlop: 1.1.0
      micromark-util-character: 2.1.1
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2

  micromark-factory-space@2.0.1:
    dependencies:
      micromark-util-character: 2.1.1
      micromark-util-types: 2.0.2

  micromark-factory-title@2.0.1:
    dependencies:
      micromark-factory-space: 2.0.1
      micromark-util-character: 2.1.1
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2

  micromark-factory-whitespace@2.0.1:
    dependencies:
      micromark-factory-space: 2.0.1
      micromark-util-character: 2.1.1
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2
```

```yaml
  micromark-util-character@2.1.1:
    dependencies:
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2

  micromark-util-chunked@2.0.1:
    dependencies:
      micromark-util-symbol: 2.0.1

  micromark-util-classify-character@2.0.1:
    dependencies:
      micromark-util-character: 2.1.1
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2

  micromark-util-combine-extensions@2.0.1:
    dependencies:
      micromark-util-chunked: 2.0.1
      micromark-util-types: 2.0.2

  micromark-util-decode-numeric-character-reference@2.0.2:
    dependencies:
      micromark-util-symbol: 2.0.1

  micromark-util-decode-string@2.0.1:
    dependencies:
      decode-named-character-reference: 1.3.0
      micromark-util-character: 2.1.1
      micromark-util-decode-numeric-character-reference: 2.0.2
      micromark-util-symbol: 2.0.1

  micromark-util-encode@2.0.1: {}

  micromark-util-html-tag-name@2.0.1: {}

  micromark-util-normalize-identifier@2.0.1:
    dependencies:
      micromark-util-symbol: 2.0.1

  micromark-util-resolve-all@2.0.1:
    dependencies:
      micromark-util-types: 2.0.2

  micromark-util-sanitize-uri@2.0.1:
    dependencies:
      micromark-util-character: 2.1.1
      micromark-util-encode: 2.0.1
      micromark-util-symbol: 2.0.1

  micromark-util-subtokenize@2.1.0:
    dependencies:
      devlop: 1.1.0
      micromark-util-chunked: 2.0.1
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2

  micromark-util-symbol@2.0.1: {}

  micromark-util-types@2.0.2: {}

  micromark@4.0.2:
    dependencies:
      '@types/debug': 4.1.13
      debug: 4.4.3
      decode-named-character-reference: 1.3.0
      devlop: 1.1.0
      micromark-core-commonmark: 2.0.3
      micromark-factory-space: 2.0.1
      micromark-util-character: 2.1.1
      micromark-util-chunked: 2.0.1
      micromark-util-combine-extensions: 2.0.1
      micromark-util-decode-numeric-character-reference: 2.0.2
      micromark-util-encode: 2.0.1
      micromark-util-normalize-identifier: 2.0.1
      micromark-util-resolve-all: 2.0.1
      micromark-util-sanitize-uri: 2.0.1
      micromark-util-subtokenize: 2.1.0
      micromark-util-symbol: 2.0.1
      micromark-util-types: 2.0.2
    transitivePeerDependencies:
      - supports-color

  micromatch@4.0.8:
    dependencies:
      braces: 3.0.3
      picomatch: 2.3.2

  minimatch@10.2.6:
    dependencies:
      brace-expansion: 5.0.9

  minimatch@3.1.5:
    dependencies:
      brace-expansion: 1.1.18

  minimist@1.2.8: {}

  ms@2.1.3: {}

  nanoid@3.3.18: {}

  napi-postinstall@0.3.4: {}

  natural-compare@1.4.0: {}
```

```
  next@15.5.23(react-dom@19.2.8(react@19.2.8))(react@19.2.8):
    dependencies:
      '@next/env': 15.5.23
      '@swc/helpers': 0.5.15
      caniuse-lite: 1.0.30001809
      postcss: 8.4.31
      react: 19.2.8
      react-dom: 19.2.8(react@19.2.8)
      styled-jsx: 5.1.6(react@19.2.8)
    optionalDependencies:
      '@next/swc-darwin-arm64': 15.5.23
      '@next/swc-darwin-x64': 15.5.23
      '@next/swc-linux-arm64-gnu': 15.5.23
      '@next/swc-linux-arm64-musl': 15.5.23
      '@next/swc-linux-x64-gnu': 15.5.23
      '@next/swc-linux-x64-musl': 15.5.23
      '@next/swc-win32-arm64-msvc': 15.5.23
      '@next/swc-win32-x64-msvc': 15.5.23
      sharp: 0.34.5
    transitivePeerDependencies:
      - '@babel/core'
      - babel-plugin-macros

  node-exports-info@1.6.2:
    dependencies:
      array.prototype.flatmap: 1.3.3
      es-errors: 1.3.0
      object.entries: 1.1.9
      semver: 6.3.1

  object-assign@4.1.1: {}

  object-inspect@1.13.4: {}

  object-keys@1.1.1: {}

  object.assign@4.1.7:
    dependencies:
      call-bind: 1.0.9
      call-bound: 1.0.4
      define-properties: 1.2.1
      es-object-atoms: 1.1.2
      has-symbols: 1.1.0
      object-keys: 1.1.1

  object.entries@1.1.9:
    dependencies:
      call-bind: 1.0.9
      call-bound: 1.0.4
      define-properties: 1.2.1
      es-object-atoms: 1.1.2

  object.fromentries@2.0.8:
    dependencies:
      call-bind: 1.0.9
      define-properties: 1.2.1
      es-abstract: 1.24.2
      es-object-atoms: 1.1.2

  object.groupby@1.0.3:
    dependencies:
      call-bind: 1.0.9
      define-properties: 1.2.1
      es-abstract: 1.24.2

  object.values@1.2.1:
    dependencies:
      call-bind: 1.0.9
      call-bound: 1.0.4
      define-properties: 1.2.1
      es-object-atoms: 1.1.2

  optionator@0.9.4:
    dependencies:
      deep-is: 0.1.4
      fast-levenshtein: 2.0.6
      levn: 0.4.1
      prelude-ls: 1.2.1
      type-check: 0.4.0
      word-wrap: 1.2.5

  own-keys@1.0.2:
    dependencies:
      call-bound: 1.0.4
      get-intrinsic: 1.3.0
      object-keys: 1.1.1
      safe-push-apply: 1.0.0

  p-limit@3.1.0:
    dependencies:
      yocto-queue: 0.1.0

  p-locate@5.0.0:
    dependencies:
      p-limit: 3.1.0

  parent-module@1.0.1:
    dependencies:
      callsites: 3.1.0

  parse-entities@4.0.2:
    dependencies:
      '@types/unist': 2.0.11
      character-entities-legacy: 3.0.0
      character-reference-invalid: 2.0.1
```

```
      decode-named-character-reference: 1.3.0
      is-alphanumerical: 2.0.1
      is-decimal: 2.0.1
      is-hexadecimal: 2.0.1

  parse5@7.3.0:
    dependencies:
      entities: 6.0.1

  path-exists@4.0.0: {}

  path-key@3.1.1: {}

  path-parse@1.0.7: {}

  picocolors@1.1.1: {}

  picomatch@2.3.2: {}

  picomatch@4.0.5: {}

  possible-typed-array-names@1.1.0: {}

  postcss@8.4.31:
    dependencies:
      nanoid: 3.3.18
      picocolors: 1.1.1
      source-map-js: 1.2.1

  postcss@8.5.26:
    dependencies:
      nanoid: 3.3.18
      picocolors: 1.1.1
      source-map-js: 1.2.1

  prelude-ls@1.2.1: {}

  prettier@3.9.6: {}

  prop-types@15.8.1:
    dependencies:
      loose-envify: 1.4.0
      object-assign: 4.1.1
      react-is: 16.13.1

  property-information@7.2.0: {}

  punycode@2.3.1: {}

  queue-microtask@1.2.3: {}

  react-dom@19.2.8(react@19.2.8):
    dependencies:
      react: 19.2.8
      scheduler: 0.27.0

  react-is@16.13.1: {}

  react-markdown@9.1.0(@types/react@19.2.18)(react@19.2.8):
    dependencies:
      '@types/hast': 3.0.5
      '@types/mdast': 4.0.4
      '@types/react': 19.2.18
      devlop: 1.1.0
      hast-util-to-jsx-runtime: 2.3.6
      html-url-attributes: 3.0.1
      mdast-util-to-hast: 13.2.1
      react: 19.2.8
      remark-parse: 11.0.0
      remark-rehype: 11.1.2
      unified: 11.0.5
      unist-util-visit: 5.1.0
      vfile: 6.0.3
    transitivePeerDependencies:
      - supports-color

  react-remove-scroll-bar@2.3.8(@types/react@19.2.18)(react@19.2.8):
    dependencies:
      react: 19.2.8
      react-style-singleton: 2.2.3(@types/react@19.2.18)(react@19.2.8)
      tslib: 2.8.1
    optionalDependencies:
      '@types/react': 19.2.18

  react-remove-scroll@2.7.2(@types/react@19.2.18)(react@19.2.8):
    dependencies:
      react: 19.2.8
      react-remove-scroll-bar: 2.3.8(@types/react@19.2.18)(react@19.2.8)
      react-style-singleton: 2.2.3(@types/react@19.2.18)(react@19.2.8)
      tslib: 2.8.1
      use-callback-ref: 1.3.3(@types/react@19.2.18)(react@19.2.8)
      use-sidecar: 1.1.3(@types/react@19.2.18)(react@19.2.8)
    optionalDependencies:
      '@types/react': 19.2.18

  react-style-singleton@2.2.3(@types/react@19.2.18)(react@19.2.8):
    dependencies:
      get-nonce: 1.0.1
      react: 19.2.8
      tslib: 2.8.1
    optionalDependencies:
      '@types/react': 19.2.18

  react@19.2.8: {}

```

```
  reflect.getprototypeof@1.0.10:
    dependencies:
      call-bind: 1.0.9
      define-properties: 1.2.1
      es-abstract: 1.24.2
      es-errors: 1.3.0
      es-object-atoms: 1.1.2
      get-intrinsic: 1.3.0
      get-proto: 1.0.1
      which-builtin-type: 1.2.1

  regexp.prototype.flags@1.5.4:
    dependencies:
      call-bind: 1.0.9
      define-properties: 1.2.1
      es-errors: 1.3.0
      get-proto: 1.0.1
      gopd: 1.2.0
      set-function-name: 2.0.2

  rehype-raw@7.0.0:
    dependencies:
      '@types/hast': 3.0.5
      hast-util-raw: 9.1.0
      vfile: 6.0.3

  rehype-sanitize@6.0.0:
    dependencies:
      '@types/hast': 3.0.5
      hast-util-sanitize: 5.0.2

  remark-gfm@4.0.1:
    dependencies:
      '@types/mdast': 4.0.4
      mdast-util-gfm: 3.1.0
      micromark-extension-gfm: 3.0.0
      remark-parse: 11.0.0
      remark-stringify: 11.0.0
      unified: 11.0.5
    transitivePeerDependencies:
      - supports-color

  remark-parse@11.0.0:
    dependencies:
      '@types/mdast': 4.0.4
      mdast-util-from-markdown: 2.0.3
      micromark-util-types: 2.0.2
      unified: 11.0.5
    transitivePeerDependencies:
      - supports-color

  remark-rehype@11.1.2:
    dependencies:
      '@types/hast': 3.0.5
      '@types/mdast': 4.0.4
      mdast-util-to-hast: 13.2.1
      unified: 11.0.5
      vfile: 6.0.3

  remark-stringify@11.0.0:
    dependencies:
      '@types/mdast': 4.0.4
      mdast-util-to-markdown: 2.1.2
      unified: 11.0.5

  resolve-from@4.0.0: {}

  resolve-pkg-maps@1.0.0: {}

  resolve@2.0.0-next.7:
    dependencies:
      es-errors: 1.3.0
      is-core-module: 2.16.2
      node-exports-info: 1.6.2
      object-keys: 1.1.1
      path-parse: 1.0.7
      supports-preserve-symlinks-flag: 1.0.0

  reusify@1.1.0: {}

  run-parallel@1.2.0:
    dependencies:
      queue-microtask: 1.2.3

  safe-array-concat@1.1.4:
    dependencies:
      call-bind: 1.0.9
      call-bound: 1.0.4
      get-intrinsic: 1.3.0
      has-symbols: 1.1.0
      isarray: 2.0.5

  safe-push-apply@1.0.0:
    dependencies:
      es-errors: 1.3.0
      isarray: 2.0.5

  safe-regex-test@1.1.0:
    dependencies:
      call-bound: 1.0.4
      es-errors: 1.3.0
      is-regex: 1.2.1

  scheduler@0.27.0: {}

```

```yaml
  semver@6.3.1: {}

  semver@7.8.5: {}

  set-function-length@1.2.2:
    dependencies:
      define-data-property: 1.1.4
      es-errors: 1.3.0
      function-bind: 1.1.2
      get-intrinsic: 1.3.0
      gopd: 1.2.0
      has-property-descriptors: 1.0.2

  set-function-name@2.0.2:
    dependencies:
      define-data-property: 1.1.4
      es-errors: 1.3.0
      functions-have-names: 1.2.3
      has-property-descriptors: 1.0.2

  set-proto@1.0.0:
    dependencies:
      dunder-proto: 1.0.1
      es-errors: 1.3.0
      es-object-atoms: 1.1.2

  sharp@0.34.5:
    dependencies:
      '@img/colour': 1.1.0
      detect-libc: 2.1.2
      semver: 7.8.5
    optionalDependencies:
      '@img/sharp-darwin-arm64': 0.34.5
      '@img/sharp-darwin-x64': 0.34.5
      '@img/sharp-libvips-darwin-arm64': 1.2.4
      '@img/sharp-libvips-darwin-x64': 1.2.4
      '@img/sharp-libvips-linux-arm': 1.2.4
      '@img/sharp-libvips-linux-arm64': 1.2.4
      '@img/sharp-libvips-linux-ppc64': 1.2.4
      '@img/sharp-libvips-linux-riscv64': 1.2.4
      '@img/sharp-libvips-linux-s390x': 1.2.4
      '@img/sharp-libvips-linux-x64': 1.2.4
      '@img/sharp-libvips-linuxmusl-arm64': 1.2.4
      '@img/sharp-libvips-linuxmusl-x64': 1.2.4
      '@img/sharp-linux-arm': 0.34.5
      '@img/sharp-linux-arm64': 0.34.5
      '@img/sharp-linux-ppc64': 0.34.5
      '@img/sharp-linux-riscv64': 0.34.5
      '@img/sharp-linux-s390x': 0.34.5
      '@img/sharp-linux-x64': 0.34.5
      '@img/sharp-linuxmusl-arm64': 0.34.5
      '@img/sharp-linuxmusl-x64': 0.34.5
      '@img/sharp-wasm32': 0.34.5
      '@img/sharp-win32-arm64': 0.34.5
      '@img/sharp-win32-ia32': 0.34.5
      '@img/sharp-win32-x64': 0.34.5
    optional: true

  shebang-command@2.0.0:
    dependencies:
      shebang-regex: 3.0.0

  shebang-regex@3.0.0: {}

  side-channel-list@1.0.1:
    dependencies:
      es-errors: 1.3.0
      object-inspect: 1.13.4

  side-channel-map@1.0.1:
    dependencies:
      call-bound: 1.0.4
      es-errors: 1.3.0
      get-intrinsic: 1.3.0
      object-inspect: 1.13.4

  side-channel-weakmap@1.0.2:
    dependencies:
      call-bound: 1.0.4
      es-errors: 1.3.0
      get-intrinsic: 1.3.0
      object-inspect: 1.13.4
      side-channel-map: 1.0.1

  side-channel@1.1.1:
    dependencies:
      es-errors: 1.3.0
      object-inspect: 1.13.4
      side-channel-list: 1.0.1
      side-channel-map: 1.0.1
      side-channel-weakmap: 1.0.2

  source-map-js@1.2.1: {}

  space-separated-tokens@2.0.2: {}

  stable-hash@0.0.5: {}

  stop-iteration-iterator@1.1.0:
    dependencies:
      es-errors: 1.3.0
      internal-slot: 1.1.0

  string.prototype.includes@2.0.1:
    dependencies:
```

```yaml
      call-bind: 1.0.9
      define-properties: 1.2.1
      es-abstract: 1.24.2

  string.prototype.matchall@4.0.12:
    dependencies:
      call-bind: 1.0.9
      call-bound: 1.0.4
      define-properties: 1.2.1
      es-abstract: 1.24.2
      es-errors: 1.3.0
      es-object-atoms: 1.1.2
      get-intrinsic: 1.3.0
      gopd: 1.2.0
      has-symbols: 1.1.0
      internal-slot: 1.1.0
      regexp.prototype.flags: 1.5.4
      set-function-name: 2.0.2
      side-channel: 1.1.1

  string.prototype.repeat@1.0.0:
    dependencies:
      define-properties: 1.2.1
      es-abstract: 1.24.2

  string.prototype.trim@1.2.11:
    dependencies:
      call-bind: 1.0.9
      call-bound: 1.0.4
      define-data-property: 1.1.4
      define-properties: 1.2.1
      es-abstract: 1.24.2
      es-object-atoms: 1.1.2
      has-property-descriptors: 1.0.2
      safe-regex-test: 1.1.0

  string.prototype.trimend@1.0.10:
    dependencies:
      call-bind: 1.0.9
      call-bound: 1.0.4
      define-properties: 1.2.1
      es-object-atoms: 1.1.2

  string.prototype.trimstart@1.0.8:
    dependencies:
      call-bind: 1.0.9
      define-properties: 1.2.1
      es-object-atoms: 1.1.2

  stringify-entities@4.0.4:
    dependencies:
      character-entities-html4: 2.1.0
      character-entities-legacy: 3.0.0

  strip-bom@3.0.0: {}

  strip-json-comments@3.1.1: {}

  style-to-js@1.1.21:
    dependencies:
      style-to-object: 1.0.14

  style-to-object@1.0.14:
    dependencies:
      inline-style-parser: 0.2.7

  styled-jsx@5.1.6(react@19.2.8):
    dependencies:
      client-only: 0.0.1
      react: 19.2.8

  supports-color@7.2.0:
    dependencies:
      has-flag: 4.0.0

  supports-preserve-symlinks-flag@1.0.0: {}

  tailwind-merge@2.6.1: {}

  tailwindcss@4.3.3: {}

  tapable@2.3.3: {}

  tinyglobby@0.2.17:
    dependencies:
      fdir: 6.5.0(picomatch@4.0.5)
      picomatch: 4.0.5

  to-regex-range@5.0.1:
    dependencies:
      is-number: 7.0.0

  trim-lines@3.0.1: {}

  trough@2.2.0: {}

  ts-api-utils@2.5.0(typescript@5.9.3):
    dependencies:
      typescript: 5.9.3

  tsconfig-paths@3.15.0:
    dependencies:
      '@types/json5': 0.0.29
      json5: 1.0.2
      minimist: 1.2.8
```

```
      strip-bom: 3.0.0

  tslib@2.8.1: {}

  type-check@0.4.0:
    dependencies:
      prelude-ls: 1.2.1

  typed-array-buffer@1.0.3:
    dependencies:
      call-bound: 1.0.4
      es-errors: 1.3.0
      is-typed-array: 1.1.15

  typed-array-byte-length@1.0.3:
    dependencies:
      call-bind: 1.0.9
      for-each: 0.3.5
      gopd: 1.2.0
      has-proto: 1.2.0
      is-typed-array: 1.1.15

  typed-array-byte-offset@1.0.4:
    dependencies:
      available-typed-arrays: 1.0.7
      call-bind: 1.0.9
      for-each: 0.3.5
      gopd: 1.2.0
      has-proto: 1.2.0
      is-typed-array: 1.1.15
      reflect.getprototypeof: 1.0.10

  typed-array-length@1.0.8:
    dependencies:
      call-bind: 1.0.9
      for-each: 0.3.5
      gopd: 1.2.0
      is-typed-array: 1.1.15
      possible-typed-array-names: 1.1.0
      reflect.getprototypeof: 1.0.10

  typescript@5.9.3: {}

  unbox-primitive@1.1.0:
    dependencies:
      call-bound: 1.0.4
      has-bigints: 1.1.0
      has-symbols: 1.1.0
      which-boxed-primitive: 1.1.1

  undici-types@6.21.0: {}

  unified@11.0.5:
    dependencies:
      '@types/unist': 3.0.3
      bail: 2.0.2
      devlop: 1.1.0
      extend: 3.0.2
      is-plain-obj: 4.1.0
      trough: 2.2.0
      vfile: 6.0.3

  unist-util-is@6.0.1:
    dependencies:
      '@types/unist': 3.0.3

  unist-util-position@5.0.0:
    dependencies:
      '@types/unist': 3.0.3

  unist-util-stringify-position@4.0.0:
    dependencies:
      '@types/unist': 3.0.3

  unist-util-visit-parents@6.0.2:
    dependencies:
      '@types/unist': 3.0.3
      unist-util-is: 6.0.1

  unist-util-visit@5.1.0:
    dependencies:
      '@types/unist': 3.0.3
      unist-util-is: 6.0.1
      unist-util-visit-parents: 6.0.2

  unrs-resolver@1.12.2:
    dependencies:
      napi-postinstall: 0.3.4
    optionalDependencies:
      '@unrs/resolver-binding-android-arm-eabi': 1.12.2
      '@unrs/resolver-binding-android-arm64': 1.12.2
      '@unrs/resolver-binding-darwin-arm64': 1.12.2
      '@unrs/resolver-binding-darwin-x64': 1.12.2
      '@unrs/resolver-binding-freebsd-x64': 1.12.2
      '@unrs/resolver-binding-linux-arm-gnueabihf': 1.12.2
      '@unrs/resolver-binding-linux-arm-musleabihf': 1.12.2
      '@unrs/resolver-binding-linux-arm64-gnu': 1.12.2
      '@unrs/resolver-binding-linux-arm64-musl': 1.12.2
      '@unrs/resolver-binding-linux-loong64-gnu': 1.12.2
      '@unrs/resolver-binding-linux-loong64-musl': 1.12.2
      '@unrs/resolver-binding-linux-ppc64-gnu': 1.12.2
      '@unrs/resolver-binding-linux-riscv64-gnu': 1.12.2
      '@unrs/resolver-binding-linux-riscv64-musl': 1.12.2
      '@unrs/resolver-binding-linux-s390x-gnu': 1.12.2
      '@unrs/resolver-binding-linux-x64-gnu': 1.12.2
```

```
      '@unrs/resolver-binding-linux-x64-musl': 1.12.2
      '@unrs/resolver-binding-openharmony-arm64': 1.12.2
      '@unrs/resolver-binding-wasm32-wasi': 1.12.2
      '@unrs/resolver-binding-win32-arm64-msvc': 1.12.2
      '@unrs/resolver-binding-win32-ia32-msvc': 1.12.2
      '@unrs/resolver-binding-win32-x64-msvc': 1.12.2

  uri-js@4.4.1:
    dependencies:
      punycode: 2.3.1

  use-callback-ref@1.3.3(@types/react@19.2.18)(react@19.2.8):
    dependencies:
      react: 19.2.8
      tslib: 2.8.1
    optionalDependencies:
      '@types/react': 19.2.18

  use-sidecar@1.1.3(@types/react@19.2.18)(react@19.2.8):
    dependencies:
      detect-node-es: 1.1.0
      react: 19.2.8
      tslib: 2.8.1
    optionalDependencies:
      '@types/react': 19.2.18

  vfile-location@5.0.3:
    dependencies:
      '@types/unist': 3.0.3
      vfile: 6.0.3

  vfile-message@4.0.3:
    dependencies:
      '@types/unist': 3.0.3
      unist-util-stringify-position: 4.0.0

  vfile@6.0.3:
    dependencies:
      '@types/unist': 3.0.3
      vfile-message: 4.0.3

  web-namespaces@2.0.1: {}

  which-boxed-primitive@1.1.1:
    dependencies:
      is-bigint: 1.1.0
      is-boolean-object: 1.2.2
      is-number-object: 1.1.1
      is-string: 1.1.1
      is-symbol: 1.1.1

  which-builtin-type@1.2.1:
    dependencies:
      call-bound: 1.0.4
      function.prototype.name: 1.2.0
      has-tostringtag: 1.0.2
      is-async-function: 2.1.1
      is-date-object: 1.1.0
      is-finalizationregistry: 1.1.1
      is-generator-function: 1.1.2
      is-regex: 1.2.1
      is-weakref: 1.1.1
      isarray: 2.0.5
      which-boxed-primitive: 1.1.1
      which-collection: 1.0.2
      which-typed-array: 1.1.22

  which-collection@1.0.2:
    dependencies:
      is-map: 2.0.3
      is-set: 2.0.3
      is-weakmap: 2.0.2
      is-weakset: 2.0.4

  which-typed-array@1.1.22:
    dependencies:
      available-typed-arrays: 1.0.7
      call-bind: 1.0.9
      call-bound: 1.0.4
      for-each: 0.3.5
      get-proto: 1.0.1
      gopd: 1.2.0
      has-tostringtag: 1.0.2

  which@2.0.2:
    dependencies:
      isexe: 2.0.0

  word-wrap@1.2.5: {}

  yocto-queue@0.1.0: {}

  zod@3.25.76: {}

  zwitch@2.0.4: {}
```

frontend/src/app/chat/page.tsx

12 lines · 0.2 KB

```
"use client";

import { AuthGuard } from "@/components/auth/auth-provider";
import { ChatView } from "@/components/chat/chat-view";

export default function ChatPage() {
  return (
    <AuthGuard>
      <ChatView />
    </AuthGuard>
  );
}
```

frontend/src/app/explore/page.tsx

12 lines · 0.2 KB

```
"use client";

import { AuthGuard } from "@/components/auth/auth-provider";
import { ExploreView } from "@/components/explore/explore-view";

export default function ExplorePage() {
  return (
    <AuthGuard>
      <ExploreView />
    </AuthGuard>
  );
}
```

frontend/src/app/globals.css

25 lines · 0.4 KB

```
@import "tailwindcss";

:root {
  --color-brand-600: #4c6ef5;
  --color-brand-700: #4263eb;
}

body {
  font-family:
    "Inter",
    system-ui,
    -apple-system,
    BlinkMacSystemFont,
    "Segoe UI",
    Roboto,
    sans-serif;
}

/* Flint chart embedded SVG styling */
.vis-flint-chart {
  max-width: 100%;
  height: auto;
  border-radius: 0.5rem;
  background: #ffffff;
}
```

frontend/src/app/layout.tsx

25 lines · 0.6 KB

```
import type { Metadata } from "next";
import "./globals.css";
import { AuthProvider } from "@/components/auth/auth-provider";

export const metadata: Metadata = {
  title: "GenBI — Generative Business Intelligence",
  description:
    "Ask questions about your data in natural language. Get SQL, charts, and insights — instantly.",
};

export default function RootLayout({
  children,
}: {
  children: React.ReactNode;
}) {
  return (
    <html lang="en">
      <body className="min-h-screen bg-gray-50 text-gray-900 antialiased">
        <AuthProvider>
          {children}
        </AuthProvider>
      </body>
    </html>
  );
}
```

frontend/src/app/login/page.tsx

28 lines · 0.8 KB

```
"use client";

import { useEffect } from "react";
import { useRouter } from "next/navigation";
import { LoginForm } from "@/components/auth/login-form";
import { useAuth } from "@/components/auth/auth-provider";

export default function LoginPage() {
  const router = useRouter();
  const { isAuthenticated, loading } = useAuth();

  // Already signed in — send them straight to the chat.
  useEffect(() => {
    if (!loading && isAuthenticated) {
      router.replace("/chat");
    }
  }, [loading, isAuthenticated, router]);

  if (loading) {
    return (
      <div className="flex items-center justify-center min-h-screen bg-gray-50">
        <div className="w-3 h-3 bg-brand-600 rounded-full animate-bounce" />
      </div>
    );
  }

  return <LoginForm onSuccess={() => router.replace("/chat")} />;
}
```

frontend/src/app/page.tsx

44 lines · 1.4 KB

```
"use client";

import Link from "next/link";
import { useAuth } from "@/components/auth/auth-provider";

export default function HomePage() {
  const { isAuthenticated, loading } = useAuth();

  return (
    <main className="min-h-screen bg-gray-50 flex flex-col items-center justify-center px-6 text-center">
      <div className="w-12 h-12 rounded-xl bg-brand-600 flex items-center justify-center mb-6">
        <span className="text-white font-bold text-xl">G</span>
      </div>

      <h1 className="text-4xl sm:text-5xl font-bold text-gray-900 tracking-tight">
        GenBI
      </h1>
      <p className="mt-4 max-w-xl text-lg text-gray-600">
        Ask questions about your data in plain English. Get validated SQL, a
        chart, and a written insight — governed, explainable, and multi-tenant.
      </p>

      <div className="mt-8">
        {loading ? (
          <span className="text-sm text-gray-400">Loading…</span>
        ) : isAuthenticated ? (
          <Link
            href="/chat"
            className="px-6 py-3 bg-brand-600 text-white rounded-lg font-medium hover:bg-brand-700 transition-colors"
          >
            Open GenBI
          </Link>
        ) : (
          <Link
            href="/login"
            className="px-6 py-3 bg-brand-600 text-white rounded-lg font-medium hover:bg-brand-700 transition-colors"
          >
            Sign in
          </Link>
        )}
      </div>
    </main>
  );
}
```

frontend/src/app/reports/page.tsx

12 lines · 0.2 KB

```
"use client";

import { AuthGuard } from "@/components/auth/auth-provider";
import { ReportsView } from "@/components/reports/reports-view";

export default function ReportsPage() {
  return (
    <AuthGuard>
      <ReportsView />
    </AuthGuard>
  );
}
```

frontend/src/components/auth/auth-provider.tsx

144 lines · 3.5 KB

```
"use client";

import { createContext, useContext, useState, useEffect, useCallback, type ReactNode } from "react";
import { useRouter } from "next/navigation";
import {
  clearSession,
  getStoredToken,
  getStoredUser,
  storeSession,
} from "@/lib/auth-storage";

interface User {
  id: string;
  email: string;
  name: string;
  tenant_id: string;
  roles: string[];
}

interface AuthState {
  user: User | null;
  token: string | null;
  loading: boolean;
  login: (email: string, password: string) => Promise<void>;
  logout: () => void;
  isAuthenticated: boolean;
}

const AuthContext = createContext<AuthState>({
  user: null,
  token: null,
  loading: true,
  login: async () => {},
  logout: () => {},
  isAuthenticated: false,
});

export function AuthProvider({ children }: { children: ReactNode }) {
  const [user, setUser] = useState<User | null>(null);
  const [token, setToken] = useState<string | null>(null);
  const [loading, setLoading] = useState(true);

  // Restore session from localStorage on mount
  useEffect(() => {
    const savedToken = getStoredToken();
    const savedUser = getStoredUser<User>();

    if (savedToken && savedUser) {
      setToken(savedToken);
      setUser(savedUser);
    } else if (savedToken || savedUser) {
      // Half-written session — drop it rather than half-authenticating.
      clearSession();
    }
    setLoading(false);
  }, []);

  const login = useCallback(async (email: string, password: string) => {
    setLoading(true);
    try {
      const res = await fetch(
        `${process.env.NEXT_PUBLIC_API_URL ?? "http://localhost:8000/api/v1"}/auth/login`,
        {
          method: "POST",
          headers: { "Content-Type": "application/json" },
          body: JSON.stringify({ email, password }),
        },
      );

      if (!res.ok) {
        const err = await res.json().catch(() => ({}));
        throw new Error(err.message ?? "Login failed");
      }

      const data = await res.json();
      const { access_token, user: userData } = data;

      setToken(access_token);
      setUser(userData);
      storeSession(access_token, userData);
    } finally {
      setLoading(false);
    }
  }, []);

  const logout = useCallback(() => {
    setToken(null);
    setUser(null);
    clearSession();
  }, []);

  return (
    <AuthContext.Provider
      value={{
        user,
        token,
        loading,
        login,
        logout,
        isAuthenticated: !!token && !!user,
```

```
      }}
    >
      {children}
    </AuthContext.Provider>
  );
}

export function useAuth() {
  const ctx = useContext(AuthContext);
  if (!ctx) {
    throw new Error("useAuth must be used within an AuthProvider");
  }
  return ctx;
}

/** HOC/guard: redirects to /login if unauthenticated */
export function AuthGuard({ children }: { children: ReactNode }) {
  const { isAuthenticated, loading } = useAuth();
  const router = useRouter();

  useEffect(() => {
    if (!loading && !isAuthenticated) {
      router.replace("/login");
    }
  }, [loading, isAuthenticated, router]);

  if (loading) {
    return (
      <div className="flex items-center justify-center h-screen">
        <div className="flex space-x-2">
          <div className="w-3 h-3 bg-brand-600 rounded-full animate-bounce" />
          <div className="w-3 h-3 bg-brand-600 rounded-full animate-bounce delay-75" />
          <div className="w-3 h-3 bg-brand-600 rounded-full animate-bounce delay-150" />
        </div>
      </div>
    );
  }

  if (!isAuthenticated) {
    return null; // redirect in flight
  }

  return <>{children}</>;
}
```

frontend/src/components/auth/login-form.tsx

86 lines · 2.9 KB

```
"use client";

import { useState } from "react";
import { useAuth } from "@/components/auth/auth-provider";

/** Email/password sign-in form. Calls onSuccess after a successful login. */
export function LoginForm({ onSuccess }: { onSuccess?: () => void }) {
  const { login } = useAuth();
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [error, setError] = useState("");
  const [submitting, setSubmitting] = useState(false);

  const handleSubmit = async (e: React.FormEvent) => {
    e.preventDefault();
    setError("");
    setSubmitting(true);
    try {
      await login(email, password);
      onSuccess?.();
    } catch (err) {
      setError(err instanceof Error ? err.message : "Login failed");
    } finally {
      setSubmitting(false);
    }
  };

  return (
    <div className="flex items-center justify-center min-h-screen bg-gray-50">
      <div className="w-full max-w-sm">
        <div className="text-center mb-8">
          <h1 className="text-2xl font-bold text-gray-900">GenBI</h1>
          <p className="text-sm text-gray-500 mt-1">Sign in to your account</p>
        </div>

        <form
          onSubmit={handleSubmit}
          className="bg-white rounded-xl border border-gray-200 shadow-sm p-6 space-y-4"
        >
          {error && (
            <div className="bg-red-50 border border-red-200 text-red-700 text-sm rounded-lg px-4 py-2">
              {error}
            </div>
          )}

          <div>
            <label className="block text-sm font-medium text-gray-700 mb-1">
              Email
            </label>
            <input
              type="email"
              value={email}
              onChange={(e) => setEmail(e.target.value)}
              placeholder="you@company.com"
              className="w-full px-3 py-2 border border-gray-300 rounded-lg text-sm focus:outline-none focus:ring-2 focus:ring-brand-600 focus:border-transparent"
              required
              autoFocus
            />
          </div>

          <div>
            <label className="block text-sm font-medium text-gray-700 mb-1">
              Password
            </label>
            <input
              type="password"
              value={password}
              onChange={(e) => setPassword(e.target.value)}
              placeholder="••••••••"
              className="w-full px-3 py-2 border border-gray-300 rounded-lg text-sm focus:outline-none focus:ring-2 focus:ring-brand-600 focus:border-transparent"
              required
            />
          </div>

          <button
            type="submit"
            disabled={submitting}
            className="w-full py-2.5 bg-brand-600 text-white rounded-lg text-sm font-medium hover:bg-brand-700 disabled:opacity-50 transition-colors"
          >
            {submitting ? "Signing in..." : "Sign in"}
          </button>
        </form>
      </div>
    </div>
  );
}
```

frontend/src/components/charts/chart-card.tsx

105 lines · 3.0 KB

```
"use client";

import { useState } from "react";

export interface ChartAssemblyInput {
  chartType: string;
  encodings: Record<string, { field: string }>;
  baseSize: { width: number; height: number };
  data: { values: Record<string, unknown>[] };
}

interface ChartCardProps {
  spec: ChartAssemblyInput;
  imageBase64?: string;
  svg?: string;
  title?: string;
  onDownload?: (format: "png" | "svg") => void;
}

export function ChartCard({ spec, imageBase64, svg, title, onDownload }: ChartCardProps) {
  const [format, setFormat] = useState<"png" | "svg">("svg");

  return (
    <div className="border border-gray-200 rounded-xl overflow-hidden bg-white">
      {/* Header */}
      <div className="flex items-center justify-between px-4 py-2 bg-gray-50 border-b border-gray-200">
        <div className="flex items-center gap-2">
          <span className="text-xs font-medium text-gray-600">
            {title ?? spec.chartType}
          </span>
          <span className="text-[10px] text-gray-400 bg-gray-200 px-1.5 py-0.5 rounded">
            Flint
          </span>
        </div>
        <div className="flex items-center gap-1">
          <button
            onClick={() => setFormat("svg")}
            className={`text-[10px] px-2 py-0.5 rounded ${
              format === "svg"
                ? "bg-brand-600 text-white"
                : "text-gray-500 hover:bg-gray-200"
            }`}
          >
            SVG
          </button>
          <button
            onClick={() => setFormat("png")}
            className={`text-[10px] px-2 py-0.5 rounded ${
              format === "png"
                ? "bg-brand-600 text-white"
                : "text-gray-500 hover:bg-gray-200"
            }`}
          >
            PNG
          </button>
          <button
            onClick={() => onDownload?.(format)}
            className="text-[10px] px-2 py-0.5 rounded text-gray-500 hover:bg-gray-200"
          >
            Download
          </button>
        </div>
      </div>

      {/* Chart */}
      <div className="p-4">
        {format === "svg" && svg ? (
          <div
            className="vis-flint-chart w-full"
            dangerouslySetInnerHTML={{ __html: svg }}
          />
        ) : imageBase64 ? (
          <img
            src={`${imageBase64}`}
            alt={spec.chartType}
            className="w-full rounded-lg"
          />
        ) : (
          <div className="flex items-center justify-center h-48 bg-gray-50 text-gray-400 text-sm rounded-lg">
            Chart rendering...
          </div>
        )}
      </div>
    </div>
  );
}

/** Render a grid of chart cards from an array of results. */
export function ChartGrid({
  charts,
}: {
  charts: { spec: ChartAssemblyInput; imageBase64?: string; svg?: string; title?: string }[];
}) {
  if (charts.length === 0) {
    return null;
  }

  return (
    <div className="grid grid-cols-1 md:grid-cols-2 gap-4">
      {charts.map((chart, i) => (
```

```
        <ChartCard key={i} {...chart} />
      ))}
    </div>
  );
}
```

frontend/src/components/chat/chat-view.tsx

575 lines · 20.8 KB

```
"use client";

import { useState, useRef, useEffect, useCallback } from "react";
import { useRouter } from "next/navigation";
import {
  listConversationMessages,
  listConversations,
  type ConversationSummary,
} from "@/lib/api-client";
import { useAuth } from "@/components/auth/auth-provider";
import { ChatRequestSchema, SSEEventSchema } from "@/lib/validators";
import type { SSEEvent } from "@/lib/validators";
import { ChartCard, type ChartAssemblyInput } from "@/components/charts/chart-card";
import { Card, CardContent } from "@/components/ui/card";
import { Button } from "@/components/ui/button";
import { Input } from "@/components/ui/input";
import { Badge } from "@/components/ui/badge";
import { Skeleton } from "@/components/ui/skeleton";
import { TooltipProvider, Tooltip, TooltipTrigger, TooltipContent } from "@/components/ui/tooltip";
import { Send, BarChart3, Database, FileText, Settings, LogOut, Plus, User } from "lucide-react";

interface StreamStage {
  stage: string;
  data: SSEEvent;
}

interface ChatMessage {
  id: string;
  role: "user" | "assistant";
  content: string;
  streaming: boolean;
  stages: StreamStage[];
  sql?: string;
  chartSpec?: Record<string, unknown>;
  chartSvg?: string;
  chartBase64?: string;
  narrative?: string;
  warnings: string[];
  // Large-query confirmation (Phase 13): set when the pipeline stops with
  // status "confirmation_required"; the panel offers Confirm and run.
  needsConfirm?: boolean;
  rowEstimate?: number | null;
  confirmQuery?: string;
}

const API_BASE = process.env.NEXT_PUBLIC_API_URL ?? "http://localhost:8000/api/v1";

export function ChatView() {
  const { user, token, logout } = useAuth();
  const [messages, setMessages] = useState<ChatMessage[]>([]);
  const [input, setInput] = useState("");
  const [loading, setLoading] = useState(false);
  const router = useRouter();
  const chatEndRef = useRef<HTMLDivElement>(null);
  const abortRef = useRef<AbortController | null>(null);

  // Conversations (Phase 14): sidebar list + the active thread. A null id
  // means "new chat" — the backend creates the conversation on first send
  // and returns its id in the SSE start event.
  const [conversations, setConversations] = useState<ConversationSummary[]>([]);
  const [conversationId, setConversationId] = useState<string | null>(null);

  const loadConversations = useCallback(async () => {
    try {
      const res = await listConversations();
      setConversations(res.conversations);
    } catch {
      // Sidebar is best-effort; chat works without it
    }
  }, []);

  useEffect(() => {
    loadConversations();
  }, [loadConversations]);

  const handleSelectConversation = useCallback(async (id: string) => {
    if (loading) return;
    setConversationId(id);
    setMessages([]);
    try {
      const res = await listConversationMessages(id);
      setMessages(
        res.messages.map((m, i) => ({
          id: `${id}-${i}`,
          role: m.role,
          content: m.content,
          streaming: false,
          stages: [],
          warnings: [],
          sql: m.generated_sql ?? undefined,
          narrative: m.role === "assistant" ? m.content : undefined,
        })),
      );
    } catch {
      setMessages([]);
    }
  }, [loading]);

  const handleNewChat = useCallback(() => {
    if (loading) return;
```

```
    setConversationId(null);
    setMessages([]);
    setInput("");
  }, [loading]);

  // Auto-scroll
  useEffect(() => {
    chatEndRef.current?.scrollIntoView({ behavior: "smooth" });
  }, [messages]);

  const updateMessageStage = useCallback(
    (msgId: string, stage: string, data: SSEEvent) => {
      setMessages((prev) =>
        prev.map((m) => {
          if (m.id !== msgId) return m;

          const updated = { ...m, stages: [...m.stages, { stage, data }] };

          // Update state from each stage
          if (data.sql && stage === "sql") updated.sql = data.sql;
          if (data.validated_sql && stage === "validation") updated.sql = data.validated_sql;
          if (data.chart_spec && stage === "chart") updated.chartSpec = data.chart_spec as Record<string, unknown>;
          if (data.svg && stage === "chart") updated.chartSvg = data.svg;
          if (data.image_base64 && stage === "chart") updated.chartBase64 = data.image_base64;
          if (data.narrative && stage === "narrative") updated.narrative = data.narrative;
          if (data.warnings) updated.warnings = data.warnings;

          if (stage === "done") {
            updated.streaming = false;
            updated.content = data.narrative ?? "Here's what I found.";
            if (data.status === "confirmation_required" && data.requires_confirmation) {
              updated.needsConfirm = true;
              updated.rowEstimate = data.row_estimate ?? null;
              updated.content =
                "This query is estimated to scan a large number of rows. " +
                "Please confirm before it runs.";
            }
          }

          return updated;
        }),
      );
    },
    [],
  );

  const handleSend = async (overrideQuery?: string, confirmLarge = false) => {
    const outgoing = overrideQuery ?? input;
    if (!outgoing.trim() || loading) return;

    // Validate input
    const parsed = ChatRequestSchema.safeParse({
      query: outgoing,
      confirm_large_query: confirmLarge || undefined,
    });
    if (!parsed.success) {
      return; // silently reject invalid input
    }

    const msgId = crypto.randomUUID();
    const userMessage: ChatMessage = {
      id: crypto.randomUUID(),
      role: "user",
      content: outgoing,
      streaming: false,
      stages: [],
      warnings: [],
    };
    const assistantMessage: ChatMessage = {
      id: msgId,
      role: "assistant",
      content: "",
      streaming: true,
      stages: [],
      warnings: [],
    };

    setMessages((prev) => [...prev, userMessage, assistantMessage]);
    setInput("");
    setLoading(true);

    try {
      const controller = new AbortController();
      abortRef.current = controller;

      const res = await fetch(`${API_BASE}/chat/stream`, {
        method: "POST",
        headers: {
          "Content-Type": "application/json",
          Authorization: `Bearer ${token}`,
        },
        body: JSON.stringify({
          query: parsed.data.query,
          confirm_large_query: parsed.data.confirm_large_query ?? false,
          conversation_id: conversationId ?? undefined,
        }),
        signal: controller.signal,
      });

      if (!res.ok) {
        throw new Error(`Stream failed: ${res.status}`);
      }

      const reader = res.body?.getReader();
      if (!reader) throw new Error("No response stream");
```

```

      const decoder = new TextDecoder();
      let buffer = "";

      while (true) {
        const { done, value } = await reader.read();
        if (done) break;

        buffer += decoder.decode(value, { stream: true });
        const lines = buffer.split("\n");
        buffer = lines.pop() ?? "";

        for (const line of lines) {
          const trimmed = line.trim();
          if (!trimmed.startsWith("data: ")) continue;

          try {
            const raw = JSON.parse(trimmed.slice(6));
            const event = SSEEventSchema.parse(raw);
            updateMessageStage(msgId, event.event, event);

            // Conversation threading (Phase 14): capture the server-assigned
            // id on start; refresh the sidebar ordering when a turn ends.
            if (event.event === "start" && event.conversation_id) {
              setConversationId(event.conversation_id);
            }
            if (event.event === "done") {
              loadConversations();
            }
          } catch {
            // Skip unparseable chunks
          }
        }
      }
    } catch (error) {
      if (error instanceof DOMException && error.name === "AbortError") return;

      setMessages((prev) =>
        prev.map((m) =>
          m.id === msgId
            ? { ...m, content: "Sorry, something went wrong.", streaming: false }
            : m,
        ),
      );
    } finally {
      setLoading(false);
      abortRef.current = null;
    }
  };

  const handleCancel = () => {
    abortRef.current?.abort();
    setLoading(false);
  };

  // Stage badge labels
  const stageLabels: Record<string, { label: string; color: "default" | "secondary" | "success" | "warning" }> = {
    intent: { label: "Intent", color: "secondary" },
    sql: { label: "SQL", color: "secondary" },
    validation: { label: "Validated", color: "success" },
    data: { label: "Results", color: "success" },
    chart: { label: "Chart", color: "secondary" },
    narrative: { label: "Insight", color: "default" },
    done: { label: "Done", color: "success" },
  };

  return (
    <div className="flex h-screen bg-gray-50">
      {/* Conversations sidebar (Phase 14) */}
      <aside className="hidden md:flex w-64 shrink-0 flex-col border-r border-gray-200 bg-white">
        <div className="flex items-center justify-between px-4 py-3 border-b border-gray-200">
          <span className="text-xs font-semibold uppercase tracking-wide text-gray-500">
            Conversations
          </span>
          <button
            onClick={handleNewChat}
            title="New chat"
            className="p-1.5 rounded-lg hover:bg-gray-100 text-gray-500 transition-colors"
          >
            <Plus className="w-4 h-4" />
          </button>
        </div>
        <div className="flex-1 overflow-y-auto py-2">
          {conversations.length === 0 && (
            <p className="px-4 py-2 text-xs text-gray-400">No conversations yet</p>
          )}
          {conversations.map((c) => (
            <button
              key={c.id}
              onClick={() => handleSelectConversation(c.id)}
              className={`w-full text-left px-4 py-2 text-sm truncate transition-colors ${
                c.id === conversationId
                  ? "bg-brand-50 text-brand-700 border-l-2 border-brand-600"
                  : "text-gray-600 hover:bg-gray-50 border-l-2 border-transparent"
              }`}
            >
              {c.title}
            </button>
          ))}
        </div>
      </aside>

      <div className="flex flex-col flex-1 min-w-0">
      {/* Top navbar */}
      <header className="flex items-center justify-between px-6 py-3 bg-white border-b border-gray-200 shrink-0">
```

```
        <div className="flex items-center gap-3">
          <div className="w-8 h-8 bg-brand-600 rounded-lg flex items-center justify-center">
            <BarChart3 className="w-5 h-5 text-white" />
          </div>
          <h1 className="text-lg font-semibold text-gray-900">GenBI</h1>
        </div>

        <div className="flex items-center gap-2">
          <TooltipProvider>
            <Tooltip>
              <TooltipTrigger asChild>
                <button
                  onClick={() => router.push("/explore")}
                  className="p-2 rounded-lg hover:bg-gray-100 text-gray-500 transition-colors"
                >
                  <Database className="w-5 h-5" />
                </button>
              </TooltipTrigger>
              <TooltipContent>Explore metrics</TooltipContent>
            </Tooltip>
          </TooltipProvider>

          <TooltipProvider>
            <Tooltip>
              <TooltipTrigger asChild>
                <button
                  onClick={() => router.push("/reports")}
                  className="p-2 rounded-lg hover:bg-gray-100 text-gray-500 transition-colors"
                >
                  <FileText className="w-5 h-5" />
                </button>
              </TooltipTrigger>
              <TooltipContent>Reports</TooltipContent>
            </Tooltip>
          </TooltipProvider>

          <TooltipProvider>
            <Tooltip>
              <TooltipTrigger asChild>
                <button className="p-2 rounded-lg hover:bg-gray-100 text-gray-500 transition-colors">
                  <Settings className="w-5 h-5" />
                </button>
              </TooltipTrigger>
              <TooltipContent>Settings</TooltipContent>
            </Tooltip>
          </TooltipProvider>

          <div className="flex items-center gap-2 pl-3 border-l border-gray-200">
            <div className="w-8 h-8 rounded-full bg-brand-100 flex items-center justify-center">
              <User className="w-4 h-4 text-brand-600" />
            </div>
            <span className="text-sm text-gray-600 hidden sm:inline">
              {user?.name ?? "User"}
            </span>
            <button
              onClick={logout}
              className="p-1.5 rounded-lg hover:bg-gray-100 text-gray-400 transition-colors"
            >
              <LogOut className="w-4 h-4" />
            </button>
          </div>
        </div>
      </header>

      {/* Messages area */}
      <div className="flex-1 overflow-y-auto">
        <div className="max-w-4xl mx-auto px-4 py-6 space-y-6">
          {messages.length === 0 && (
            <div className="text-center mt-32">
              <div className="w-16 h-16 bg-brand-100 rounded-2xl flex items-center justify-center mx-auto mb-4">
                <BarChart3 className="w-8 h-8 text-brand-600" />
              </div>
              <h2 className="text-2xl font-light text-gray-600 mb-2">
                Ask anything about your data
              </h2>
              <p className="text-sm text-gray-400">
                Natural language to SQL, charts, and insights — instantly
              </p>
              <div className="flex gap-2 justify-center mt-6 flex-wrap">
                {[
                  "Show revenue by region",
                  "Monthly active users trend",
                  "Top 10 customers by value",
                ].map((suggestion) => (
                  <button
                    key={suggestion}
                    onClick={() => {
                      setInput(suggestion);
                    }}
                    className="px-3 py-1.5 text-xs bg-white border border-gray-200 rounded-full text-gray-600 hover:border-brand-300 hover:text-brand-600 transition-colors"
                  >
                    {suggestion}
                  </button>
                ))}
              </div>
            </div>
          )}

          {messages.map((msg) => (
            <div
              key={msg.id}
              className={`flex ${msg.role === "user" ? "justify-end" : "justify-start"}`}
            >
              <div className={`max-w-[85%] ${msg.role === "user" ? "" : "w-full"}`}>
```

```
                {/* User bubble */}
                {msg.role === "user" && (
                  <div className="bg-brand-600 text-white rounded-2xl rounded-br-md px-5 py-3 shadow-sm">
                    <p className="text-sm whitespace-pre-wrap">{msg.content}</p>
                  </div>
                )}

                {/* Assistant card */}
                {msg.role === "assistant" && (
                  <Card>
                    <CardContent className="pt-6 space-y-4">
                      {/* Stage badges */}
                      {msg.stages.length > 0 && (
                        <div className="flex flex-wrap gap-1.5">
                          {msg.stages.map((s, i) => {
                            const info = stageLabels[s.stage] ?? { label: s.stage, color: "secondary" as const };
                            return (
                              <Badge key={i} variant={info.color} className="text-[10px]">
                                {info.label}
                              </Badge>
                            );
                          })}
                          {msg.streaming && (
                            <Badge variant="secondary" className="text-[10px] animate-pulse">
                              ...
                            </Badge>
                          )}
                        </div>
                      )}

                      {/* Streaming skeleton */}
                      {msg.streaming && msg.stages.length === 0 && (
                        <div className="space-y-2">
                          <Skeleton className="h-4 w-3/4" />
                          <Skeleton className="h-4 w-1/2" />
                          <Skeleton className="h-4 w-5/6" />
                        </div>
                      )}

                      {/* SQL block */}
                      {msg.sql && (
                        <div className="bg-gray-900 rounded-lg overflow-hidden">
                          <div className="flex items-center justify-between px-4 py-2 bg-gray-800">
                            <span className="text-[10px] font-medium text-gray-400 uppercase tracking-wider">
                              Generated SQL
                            </span>
                            <button
                              onClick={() => navigator.clipboard.writeText(msg.sql!)}
                              className="text-[10px] text-gray-500 hover:text-gray-300 transition-colors"
                            >
                              Copy
                            </button>
                          </div>
                          <pre className="p-4 text-xs text-green-400 font-mono overflow-x-auto">
                            {msg.sql}
                          </pre>
                        </div>
                      )}

                      {/* Chart */}
                      {(msg.chartSvg || msg.chartBase64 || msg.chartSpec) && (
                        <ChartCard
                          spec={msg.chartSpec as unknown as ChartAssemblyInput}
                          svg={msg.chartSvg}
                          imageBase64={msg.chartBase64}
                        />
                      )}

                      {/* Narrative */}
                      {msg.narrative && !msg.streaming && (
                        <p className="text-sm text-gray-700 leading-relaxed">
                          {msg.narrative}
                        </p>
                      )}

                      {/* Warnings */}
                      {msg.warnings.length > 0 && !msg.streaming && (
                        <div className="text-xs text-amber-600 bg-amber-50 rounded-lg px-3 py-2">
                          {msg.warnings.map((w, i) => (
                            <div key={i}>{w}</div>
                          ))}
                        </div>
                      )}

                      {/* Large-query confirmation (Phase 13) */}
                      {msg.needsConfirm && !msg.streaming && (
                        <div className="flex items-center justify-between gap-3 bg-amber-50 border border-amber-200 rounded-lg px-3 py-2">
                          <span className="text-xs text-amber-700">
                            Estimated to scan{" "}
                            {msg.rowEstimate ? `~${msg.rowEstimate.toLocaleString()} rows` : "many rows"}.
                            Run anyway?
                          </span>
                          <Button
                            size="sm"
                            onClick={() => {
                              const idx = messages.findIndex((m) => m.id === msg.id);
                              const original = idx > 0 ? messages[idx - 1].content : "";
                              if (original) handleSend(original, true);
                            }}
                          >
                            Confirm and run
                          </Button>
                        </div>
                      )}
                    </CardContent>
```

```
                  </Card>
                )}
              </div>
            </div>
          ))}

          <div ref={chatEndRef} />
        </div>
      </div>

      {/* Input bar */}
      <div className="border-t border-gray-200 bg-white shrink-0">
        <div className="max-w-4xl mx-auto px-4 py-4">
          <div className="flex gap-3">
            <Input
              value={input}
              onChange={(e) => setInput(e.target.value)}
              onKeyDown={(e) => {
                if (e.key === "Enter" && !e.shiftKey) {
                  e.preventDefault();
                  handleSend();
                }
              }}              placeholder="Ask a question about your data..."
              className="flex-1 rounded-xl"
              disabled={loading}
            />
            {loading ? (
              <Button
                onClick={handleCancel}
                variant="destructive"
                size="lg"
                className="rounded-xl"
              >
                Cancel
              </Button>
            ) : (
              <Button
                onClick={() => handleSend()}
                disabled={!input.trim()}
                size="lg"
                className="rounded-xl"
              >
                <Send className="w-4 h-4 mr-1.5" />
                Send
              </Button>
            )}
          </div>
          <p className="text-[10px] text-gray-400 mt-2 text-center">
            GenBI may produce inaccurate results. Always verify data with source systems.
          </p>
        </div>
      </div>
      </div>
    </div>
  );
}
```

frontend/src/components/explore/explore-view.tsx

305 lines · 11.8 KB

```
"use client";

import { useCallback, useEffect, useMemo, useState } from "react";
import Link from "next/link";
import {
  listMetrics,
  queryMetrics,
  renderChart,
  type MetricQueryResponse,
  type MetricSummary,
} from "@/lib/api-client";
import { ChartCard, type ChartAssemblyInput } from "@/components/charts/chart-card";
import { Badge } from "@/components/ui/badge";
import { Button } from "@/components/ui/button";
import { ArrowLeft, BarChart3, Play } from "lucide-react";

const selectClass =
  "w-full px-3 py-2 border border-gray-300 rounded-lg text-sm bg-white " +
  "focus:outline-none focus:ring-2 focus:ring-brand-600 focus:border-transparent";

type Granularity = "none" | "day" | "month";

export function ExploreView() {
  const [metrics, setMetrics] = useState<MetricSummary[]>([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState("");

  const [measure, setMeasure] = useState("");
  const [dimension, setDimension] = useState("");
  const [granularity, setGranularity] = useState<Granularity>("none");
  const [limit, setLimit] = useState(50);

  const [running, setRunning] = useState(false);
  const [result, setResult] = useState<MetricQueryResponse | null>(null);
  const [chartSpec, setChartSpec] = useState<ChartAssemblyInput | null>(null);
  const [chartSvg, setChartSvg] = useState<string | undefined>();

  useEffect(() => {
    let cancelled = false;
    (async () => {
      try {
        const res = await listMetrics();
        if (cancelled) return;
        setMetrics(res.metrics);
        if (res.metrics.length > 0) setMeasure(res.metrics[0].name);
      } catch (e) {
        if (!cancelled) setError(e instanceof Error ? e.message : "Failed to load catalog");
      } finally {
        if (!cancelled) setLoading(false);
      }
    })();
    return () => {
      cancelled = true;
    };
  }, []);

  const selected = useMemo(
    () => metrics.find((m) => m.name === measure) ?? null,
    [metrics, measure],
  );

  // Reset dependent picks when the measure changes.
  useEffect(() => {
    setDimension("");
    setGranularity("none");
  }, [measure]);

  const runQuery = useCallback(async () => {
    if (!selected) return;
    setRunning(true);
    setError("");
    setResult(null);
    setChartSpec(null);
    setChartSvg(undefined);
    try {
      const res = await queryMetrics({
        measures: [selected.name],
        ...(dimension ? { dimensions: [dimension] } : {}),
        ...(granularity !== "none" && selected.time_dimensions.length > 0
          ? {
              time_dimensions: [
                {
                  dimension: selected.time_dimensions[0],
                  granularity,
                },
              ],
            }
          : {}),
        limit,
      });
      setResult(res);

      // Render a bar chart when there is a dimension to slice by.
      const xField = dimension ? dimension.split(".").pop() : granularity !== "none" ? granularity : null;
      const yField = selected.measure_name;
      if (xField && res.data.length > 0 && res.data.length <= 50) {
        const spec: ChartAssemblyInput = {
          chartType: "Bar Chart",
          encodings: { x: { field: xField }, y: { field: yField } },
          baseSize: { width: 600, height: 400 },
```

```
          data: { values: res.data },
        };
        setChartSpec(spec);
        try {
          const rendered = await renderChart({ spec, backend: "vegalite", format: "svg" });
          setChartSvg(rendered.svg ?? undefined);
        } catch {
          setChartSvg(undefined); // table still renders; chart is best-effort
        }
      }
    } catch (e) {
      setError(e instanceof Error ? e.message : "Query failed");
    } finally {
      setRunning(false);
    }
  }, [selected, dimension, granularity, limit]);

  const columns = useMemo(() => {
    if (!result || result.data.length === 0) return [];
    return Object.keys(result.data[0]);
  }, [result]);

  return (
    <div className="flex flex-col h-screen bg-gray-50">
      <header className="shrink-0 flex items-center justify-between px-4 py-3 bg-white border-b border-gray-200">
        <div className="flex items-center gap-3">
          <div className="w-8 h-8 rounded-lg bg-brand-600 flex items-center justify-center">
            <BarChart3 className="w-5 h-5 text-white" />
          </div>
          <div>
            <h1 className="text-sm font-semibold text-gray-900">Explore</h1>
            <p className="text-[11px] text-gray-500">Semantic layer metrics</p>
          </div>
        </div>
        <Link
          href="/chat"
          className="flex items-center gap-1.5 px-3 py-1.5 text-sm text-gray-600 rounded-lg hover:bg-gray-100 transition-colors"
        >
          <ArrowLeft className="w-4 h-4" />
          Back to chat
        </Link>
      </header>

      <main className="flex-1 overflow-y-auto">
        <div className="max-w-5xl mx-auto px-4 py-6 space-y-6">
          {error && (
            <div className="bg-red-50 border border-red-200 text-red-700 text-sm rounded-lg px-4 py-2">
              {error}
            </div>
          )}

          {loading ? (
            <div className="flex items-center justify-center py-16">
              <div className="w-3 h-3 bg-brand-600 rounded-full animate-bounce" />
            </div>
          ) : (
            <>
              {/* Catalog */}
              <section>
                <h2 className="text-xs font-semibold uppercase tracking-wide text-gray-500 mb-2">
                  Catalog ({metrics.length} metrics)
                </h2>
                <div className="grid grid-cols-2 md:grid-cols-3 gap-2">
                  {metrics.map((m) => (
                    <button
                      key={m.name}
                      onClick={() => setMeasure(m.name)}
                      className={`text-left p-3 bg-white border rounded-xl transition-colors ${
                        m.name === measure
                          ? "border-brand-500 ring-1 ring-brand-500"
                          : "border-gray-200 hover:border-brand-300"
                      }`}
                    >
                      <div className="text-sm font-medium text-gray-900 truncate">{m.title}</div>
                      <div className="text-[11px] text-gray-400 truncate">{m.cube_name}</div>
                      <Badge variant="secondary" className="mt-1.5">{m.metric_type}</Badge>
                    </button>
                  ))}
                </div>
              </section>

              {/* Query builder */}
              <section className="bg-white border border-gray-200 rounded-xl p-4">
                <h2 className="text-xs font-semibold uppercase tracking-wide text-gray-500 mb-3">
                  Query
                </h2>
                <div className="grid grid-cols-2 md:grid-cols-4 gap-3">
                  <div>
                    <label className="block text-xs font-medium text-gray-600 mb-1">Measure</label>
                    <select value={measure} onChange={(e) => setMeasure(e.target.value)} className={selectClass}>
                      {metrics.map((m) => (
                        <option key={m.name} value={m.name}>
                          {m.title} ({m.cube_name})
                        </option>
                      ))}
                    </select>
                  </div>
                  <div>
                    <label className="block text-xs font-medium text-gray-600 mb-1">Group by</label>
                    <select
                      value={dimension}
                      onChange={(e) => setDimension(e.target.value)}
                      className={selectClass}
                      disabled={!selected}
                    >
```

```
                      <option value="">— none —</option>
                      {selected?.dimensions.map((d) => (
                        <option key={d} value={d}>
                          {d.split(".").pop()}
                        </option>
                      ))}
                    </select>
                  </div>
                  <div>
                    <label className="block text-xs font-medium text-gray-600 mb-1">Time</label>
                    <select
                      value={granularity}
                      onChange={(e) => setGranularity(e.target.value as Granularity)}
                      className={selectClass}
                      disabled={!selected || selected.time_dimensions.length === 0}
                    >
                      <option value="none">— none —</option>
                      <option value="day">by day</option>
                      <option value="month">by month</option>
                    </select>
                  </div>
                  <div>
                    <label className="block text-xs font-medium text-gray-600 mb-1">Row limit</label>
                    <select value={limit} onChange={(e) => setLimit(Number(e.target.value))} className={selectClass}>
                      {[10, 25, 50, 100].map((n) => (
                        <option key={n} value={n}>
                          {n}
                        </option>
                      ))}
                    </select>
                  </div>
                </div>
                <div className="mt-3 flex items-center gap-3">
                  <Button onClick={runQuery} disabled={running || !selected}>
                    <Play className="w-4 h-4 mr-1" />
                    {running ? "Running…" : "Run query"}
                  </Button>
                  {result && (
                    <span className="text-xs text-gray-400">
                      {result.data.length} rows · {result.latency_ms} ms
                      {result.cached ? " · cached" : ""}
                    </span>
                  )}
                </div>
              </section>

              {/* Results */}
              {result && result.data.length > 0 && (
                <section className="space-y-4">
                  {chartSpec && (
                    <ChartCard spec={chartSpec} svg={chartSvg} title={selected?.title} />
                  )}
                  <div className="bg-white border border-gray-200 rounded-xl overflow-hidden">
                    <table className="w-full text-sm">
                      <thead>
                        <tr className="bg-gray-50 border-b border-gray-200">
                          {columns.map((c) => (
                            <th key={c} className="text-left px-4 py-2 text-xs font-medium text-gray-600">
                              {c}
                            </th>
                          ))}
                        </tr>
                      </thead>
                      <tbody>
                        {result.data.map((row, i) => (
                          <tr key={i} className="border-b border-gray-100 last:border-0">
                            {columns.map((c) => (
                              <td key={c} className="px-4 py-2 text-gray-700">
                                {formatCell(row[c])}
                              </td>
                            ))}
                          </tr>
                        ))}
                      </tbody>
                    </table>
                  </div>
                </section>
              )}

              {result && result.data.length === 0 && (
                <div className="bg-white border border-gray-200 rounded-xl px-4 py-8 text-center text-sm text-gray-500">
                  No rows. Under tenant RLS this usually means the tenant has no data
                  for this metric — try seeding (`make seed`).
                </div>
              )}
            </>
          )}
        </div>
      </main>
    </div>
  );
}

function formatCell(value: unknown): string {
  if (value === null || value === undefined) return "—";
  if (typeof value === "number") {
    return Number.isInteger(value) ? value.toLocaleString() : value.toLocaleString(undefined, { maximumFractionDigits: 2 });
  }
  return String(value);
}
```

frontend/src/components/reports/reports-view.tsx

223 lines · 8.3 KB

```
"use client";

import { useCallback, useEffect, useState } from "react";
import Link from "next/link";
import {
  generateReport,
  getReport,
  listReports,
  type Report,
  type ReportSummary,
} from "@/lib/api-client";
import { ChartCard, type ChartAssemblyInput } from "@/components/charts/chart-card";
import { Badge } from "@/components/ui/badge";
import { Button } from "@/components/ui/button";
import { Input } from "@/components/ui/input";
import { ArrowLeft, FileText, Play } from "lucide-react";

export function ReportsView() {
  const [reports, setReports] = useState<ReportSummary[]>([]);
  const [prompt, setPrompt] = useState("");
  const [sectionCount, setSectionCount] = useState(3);
  const [generating, setGenerating] = useState(false);
  const [error, setError] = useState("");
  const [active, setActive] = useState<Report | null>(null);
  const [loadingReport, setLoadingReport] = useState(false);

  const loadReports = useCallback(async () => {
    try {
      const res = await listReports();
      setReports(res.reports);
    } catch {
      // Sidebar is best-effort
    }
  }, []);

  useEffect(() => {
    loadReports();
  }, [loadReports]);

  const handleGenerate = async () => {
    if (!prompt.trim() || generating) return;
    setGenerating(true);
    setError("");
    setActive(null);
    try {
      const report = await generateReport(prompt, sectionCount);
      setActive(report);
      loadReports();
    } catch (e) {
      setError(e instanceof Error ? e.message : "Report generation failed");
    } finally {
      setGenerating(false);
    }
  };

  const handleSelect = async (id: string) => {
    if (generating) return;
    setLoadingReport(true);
    setError("");
    try {
      setActive(await getReport(id));
    } catch (e) {
      setError(e instanceof Error ? e.message : "Could not load report");
    } finally {
      setLoadingReport(false);
    }
  };

  return (
    <div className="flex h-screen bg-gray-50">
      {/* Reports sidebar */}
      <aside className="hidden md:flex w-64 shrink-0 flex-col border-r border-gray-200 bg-white">
        <div className="flex items-center justify-between px-4 py-3 border-b border-gray-200">
          <span className="text-xs font-semibold uppercase tracking-wide text-gray-500">
            Reports
          </span>
        </div>
        <div className="flex-1 overflow-y-auto py-2">
          {reports.length === 0 && (
            <p className="px-4 py-2 text-xs text-gray-400">No reports yet</p>
          )}
          {reports.map((r) => (
            <button
              key={r.id}
              onClick={() => handleSelect(r.id)}
              className={`w-full text-left px-4 py-2 transition-colors ${
                active?.report_id === r.id
                  ? "bg-brand-50 text-brand-700 border-l-2 border-brand-600"
                  : "text-gray-600 hover:bg-gray-50 border-l-2 border-transparent"
              }`}
            >
              <div className="text-sm truncate">{r.title}</div>
              <div className="text-[11px] text-gray-400">
                {r.section_count} section{r.section_count === 1 ? "" : "s"}
              </div>
            </button>
          ))}
        </div>
      </aside>

```

```
      {/* Main */}
      <div className="flex flex-col flex-1 min-w-0">
        <header className="shrink-0 flex items-center justify-between px-4 py-3 bg-white border-b border-gray-200">
          <div className="flex items-center gap-3">
            <div className="w-8 h-8 rounded-lg bg-brand-600 flex items-center justify-center">
              <FileText className="w-5 h-5 text-white" />
            </div>
            <div>
              <h1 className="text-sm font-semibold text-gray-900">Reports</h1>
              <p className="text-[11px] text-gray-500">Multi-chart reports from a prompt</p>
            </div>
          </div>
          <Link
            href="/chat"
            className="flex items-center gap-1.5 px-3 py-1.5 text-sm text-gray-600 rounded-lg hover:bg-gray-100 transition-colors"
          >
            <ArrowLeft className="w-4 h-4" />
            Back to chat
          </Link>
        </header>

        <main className="flex-1 overflow-y-auto">
          <div className="max-w-4xl mx-auto px-4 py-6 space-y-6">
            {error && (
              <div className="bg-red-50 border border-red-200 text-red-700 text-sm rounded-lg px-4 py-2">
                {error}
              </div>
            )}

            {/* Generator */}
            <section className="bg-white border border-gray-200 rounded-xl p-4">
              <div className="flex gap-3">
                <Input
                  value={prompt}
                  onChange={(e) => setPrompt(e.target.value)}
                  onKeyDown={(e) => {
                    if (e.key === "Enter") handleGenerate();
                  }}
                  placeholder='e.g. "Q3 performance: revenue, pipeline, and active users"'
                  disabled={generating}
                  className="flex-1 rounded-xl"
                />
                <select
                  value={sectionCount}
                  onChange={(e) => setSectionCount(Number(e.target.value))}
                  className="px-3 py-2 border border-gray-300 rounded-lg text-sm bg-white focus:outline-none focus:ring-2 focus:ring-brand-600"
                >
                  {[2, 3, 4].map((n) => (
                    <option key={n} value={n}>
                      {n} charts
                    </option>
                  ))}
                </select>
                <Button onClick={handleGenerate} disabled={generating || !prompt.trim()}>
                  <Play className="w-4 h-4 mr-1" />
                  {generating ? "Generating…" : "Generate"}
                </Button>
              </div>
              <p className="text-[10px] text-gray-400 mt-2">
                Picks metrics from the semantic layer, runs tenant-scoped queries, renders charts.
              </p>
            </section>

            {loadingReport && (
              <div className="flex items-center justify-center py-16">
                <div className="w-3 h-3 bg-brand-600 rounded-full animate-bounce" />
              </div>
            )}

            {/* Report display */}
            {active && !loadingReport && (
              <section className="space-y-6">
                <div>
                  <h2 className="text-xl font-semibold text-gray-900">{active.title}</h2>
                  {active.summary && (
                    <p className="text-sm text-gray-600 mt-2 leading-relaxed">{active.summary}</p>
                  )}
                  <div className="flex gap-2 mt-2">
                    <Badge variant="secondary">{active.sections.length} sections</Badge>
                    <Badge variant="outline">{new Date(active.created_at).toLocaleString()}</Badge>
                  </div>
                </div>

                {active.warnings.length > 0 && (
                  <div className="text-xs text-amber-600 bg-amber-50 rounded-lg px-3 py-2">
                    {active.warnings.map((w, i) => (
                      <div key={i}>{w}</div>
                    ))}
                  </div>
                )}

                {active.sections.map((s) => (
                  <div
                    key={s.position}
                    className="bg-white border border-gray-200 rounded-xl p-4 space-y-3"
                  >
                    <div className="flex items-center justify-between">
                      <h3 className="text-sm font-semibold text-gray-900">{s.section_title}</h3>
                      <span className="text-xs text-gray-400">
                        {s.data_total != null ? `Total: ${s.data_total.toLocaleString()}` : ""}
                        {` · ${s.row_count} rows`}
                      </span>
                    </div>
                    {s.chart_svg && (
                      <ChartCard
```

```
                        spec={s.chart_spec as unknown as ChartAssemblyInput}
                        svg={s.chart_svg}
                        title={s.section_title}
                      />
                    )}
                    {s.narrative && (
                      <p className="text-sm text-gray-600 leading-relaxed">{s.narrative}</p>
                    )}
                  </div>
                ))}
              </section>
            )}
          </div>
        </main>
      </div>
    </div>
  );
}
```

frontend/src/components/ui/avatar.tsx

49 lines · 1.4 KB

```
"use client";

import * as React from "react";
import * as AvatarPrimitive from "@radix-ui/react-avatar";
import { cn } from "@/lib/shadcn";

const Avatar = React.forwardRef<
  React.ElementRef<typeof AvatarPrimitive.Root>,
  React.ComponentPropsWithoutRef<typeof AvatarPrimitive.Root>
>(({ className, ...props }, ref) => (
  <AvatarPrimitive.Root
    ref={ref}
    className={cn(
      "relative flex h-10 w-10 shrink-0 overflow-hidden rounded-full",
      className,
    )}
    {...props}
  />
));
Avatar.displayName = AvatarPrimitive.Root.displayName;

const AvatarImage = React.forwardRef<
  React.ElementRef<typeof AvatarPrimitive.Image>,
  React.ComponentPropsWithoutRef<typeof AvatarPrimitive.Image>
>(({ className, ...props }, ref) => (
  <AvatarPrimitive.Image
    ref={ref}
    className={cn("aspect-square h-full w-full", className)}
    {...props}
  />
));
AvatarImage.displayName = AvatarPrimitive.Image.displayName;

const AvatarFallback = React.forwardRef<
  React.ElementRef<typeof AvatarPrimitive.Fallback>,
  React.ComponentPropsWithoutRef<typeof AvatarPrimitive.Fallback>
>(({ className, ...props }, ref) => (
  <AvatarPrimitive.Fallback
    ref={ref}
    className={cn(
      "flex h-full w-full items-center justify-center rounded-full bg-gray-100 text-sm font-medium text-gray-600",
      className,
    )}
    {...props}
  />
));
AvatarFallback.displayName = AvatarPrimitive.Fallback.displayName;

export { Avatar, AvatarImage, AvatarFallback };
```

frontend/src/components/ui/badge.tsx

34 lines · 1.0 KB

```
import * as React from "react";
import { cn } from "@/lib/shadcn";
import { cva, type VariantProps } from "class-variance-authority";

const badgeVariants = cva(
  "inline-flex items-center rounded-full px-2.5 py-0.5 text-xs font-medium transition-colors focus:outline-none focus:ring-2 focus:ring-brand-600 focu
s:ring-offset-2",
  {
    variants: {
      variant: {
        default: "bg-brand-600 text-white",
        secondary: "bg-gray-100 text-gray-800",
        destructive: "bg-red-100 text-red-700",
        outline: "text-gray-600 border border-gray-300",
        success: "bg-green-100 text-green-700",
        warning: "bg-amber-100 text-amber-700",
      },
    },
    defaultVariants: {
      variant: "default",
    },
  },
);

export interface BadgeProps
  extends React.HTMLAttributes<HTMLDivElement>,
    VariantProps<typeof badgeVariants> {}

function Badge({ className, variant, ...props }: BadgeProps) {
  return (
    <div className={cn(badgeVariants({ variant }), className)} {...props} />
  );
}

export { Badge, badgeVariants };
```

frontend/src/components/ui/button.tsx

53 lines · 1.7 KB

```
import * as React from "react";
import { Slot } from "@radix-ui/react-slot";
import { cva, type VariantProps } from "class-variance-authority";
import { cn } from "@/lib/shadcn";

const buttonVariants = cva(
  "inline-flex items-center justify-center whitespace-nowrap rounded-lg text-sm font-medium transition-colors focus-visible:outline-none focus-visible
:ring-2 focus-visible:ring-brand-600 focus-visible:ring-offset-2 disabled:pointer-events-none disabled:opacity-50",
  {
    variants: {
      variant: {
        default: "bg-brand-600 text-white hover:bg-brand-700 shadow-sm",
        destructive: "bg-red-500 text-white hover:bg-red-600 shadow-sm",
        outline:
          "border border-gray-300 bg-white hover:bg-gray-50 text-gray-700",
        secondary: "bg-gray-100 text-gray-900 hover:bg-gray-200",
        ghost: "hover:bg-gray-100 text-gray-700",
        link: "text-brand-600 underline-offset-4 hover:underline",
      },
      size: {
        default: "h-10 px-4 py-2",
        sm: "h-8 rounded-md px-3 text-xs",
        lg: "h-12 rounded-xl px-8 text-base",
        icon: "h-10 w-10",
      },
    },
    defaultVariants: {
      variant: "default",
      size: "default",
    },
  },
);

export interface ButtonProps
  extends React.ButtonHTMLAttributes<HTMLButtonElement>,
    VariantProps<typeof buttonVariants> {
  asChild?: boolean;
}

const Button = React.forwardRef<HTMLButtonElement, ButtonProps>(
  ({ className, variant, size, asChild = false, ...props }, ref) => {
    const Comp = asChild ? Slot : "button";
    return (
      <Comp
        className={cn(buttonVariants({ variant, size, className }))}
        ref={ref}
        {...props}
      />
    );
  },
);
Button.displayName = "Button";

export { Button, buttonVariants };
```

frontend/src/components/ui/card.tsx

76 lines · 1.9 KB

```
import * as React from "react";
import { cn } from "@/lib/shadcn";

const Card = React.forwardRef<HTMLDivElement, React.HTMLAttributes<HTMLDivElement>>(
  ({ className, ...props }, ref) => (
    <div
      ref={ref}
      className={cn(
        "rounded-xl border border-gray-200 bg-white text-gray-900 shadow-sm",
        className,
      )}
      {...props}
    />
  ),
);
Card.displayName = "Card";

const CardHeader = React.forwardRef<HTMLDivElement, React.HTMLAttributes<HTMLDivElement>>(
  ({ className, ...props }, ref) => (
    <div
      ref={ref}
      className={cn("flex flex-col space-y-1.5 p-6", className)}
      {...props}
    />
  ),
);
CardHeader.displayName = "CardHeader";

const CardTitle = React.forwardRef<HTMLParagraphElement, React.HTMLAttributes<HTMLHeadingElement>>(
  ({ className, ...props }, ref) => (
    <h3
      ref={ref}
      className={cn("text-lg font-semibold leading-none tracking-tight", className)}
      {...props}
    />
  ),
);
CardTitle.displayName = "CardTitle";

const CardDescription = React.forwardRef<HTMLParagraphElement, React.HTMLAttributes<HTMLParagraphElement>>(
  ({ className, ...props }, ref) => (
    <p
      ref={ref}
      className={cn("text-sm text-gray-500", className)}
      {...props}
    />
  ),
);
CardDescription.displayName = "CardDescription";

const CardContent = React.forwardRef<HTMLDivElement, React.HTMLAttributes<HTMLDivElement>>(
  ({ className, ...props }, ref) => (
    <div ref={ref} className={cn("p-6 pt-0", className)} {...props} />
  ),
);
CardContent.displayName = "CardContent";

const CardFooter = React.forwardRef<HTMLDivElement, React.HTMLAttributes<HTMLDivElement>>(
  ({ className, ...props }, ref) => (
    <div
      ref={ref}
      className={cn("flex items-center p-6 pt-0", className)}
      {...props}
    />
  ),
);
CardFooter.displayName = "CardFooter";

export {
  Card,
  CardHeader,
  CardFooter,
  CardTitle,
  CardDescription,
  CardContent,
};
```

frontend/src/components/ui/dialog.tsx

109 lines · 3.4 KB

```
"use client";

import * as React from "react";
import * as DialogPrimitive from "@radix-ui/react-dialog";
import { X } from "lucide-react";
import { cn } from "@/lib/shadcn";

const Dialog = DialogPrimitive.Root;
const DialogTrigger = DialogPrimitive.Trigger;
const DialogPortal = DialogPrimitive.Portal;
const DialogClose = DialogPrimitive.Close;

const DialogOverlay = React.forwardRef<
  React.ElementRef<typeof DialogPrimitive.Overlay>,
  React.ComponentPropsWithoutRef<typeof DialogPrimitive.Overlay>
>(({ className, ...props }, ref) => (
  <DialogPrimitive.Overlay
    ref={ref}
    className={cn(
      "fixed inset-0 z-50 bg-black/50 data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0",
      className,
    )}
    {...props}
  />
));
DialogOverlay.displayName = DialogPrimitive.Overlay.displayName;

const DialogContent = React.forwardRef<
  React.ElementRef<typeof DialogPrimitive.Content>,
  React.ComponentPropsWithoutRef<typeof DialogPrimitive.Content>
>(({ className, children, ...props }, ref) => (
  <DialogPortal>
    <DialogOverlay />
    <DialogPrimitive.Content
      ref={ref}
      className={cn(
        "fixed left-[50%] top-[50%] z-50 grid w-full max-w-lg translate-x-[-50%] translate-y-[-50%] gap-4 border border-gray-200 bg-white p-6 shadow-lg rounded-xl duration-200 data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0 data-[state=closed]:zoom-out-95 data-[state=open]:zoom-in-95",
        className,
      )}
      {...props}
    >
      {children}
      <DialogPrimitive.Close className="absolute right-4 top-4 rounded-sm opacity-70 transition-opacity hover:opacity-100 focus:outline-none focus:ring-2 focus:ring-brand-600">
        <X className="h-4 w-4" />
        <span className="sr-only">Close</span>
      </DialogPrimitive.Close>
    </DialogPrimitive.Content>
  </DialogPortal>
));
DialogContent.displayName = DialogPrimitive.Content.displayName;

const DialogHeader = ({
  className,
  ...props
}: React.HTMLAttributes<HTMLDivElement>) => (
  <div
    className={cn("flex flex-col space-y-1.5 text-center sm:text-left", className)}
    {...props}
  />
);
DialogHeader.displayName = "DialogHeader";

const DialogFooter = ({
  className,
  ...props
}: React.HTMLAttributes<HTMLDivElement>) => (
  <div
    className={cn("flex flex-col-reverse sm:flex-row sm:justify-end sm:space-x-2", className)}
    {...props}
  />
);
DialogFooter.displayName = "DialogFooter";

const DialogTitle = React.forwardRef<
  React.ElementRef<typeof DialogPrimitive.Title>,
  React.ComponentPropsWithoutRef<typeof DialogPrimitive.Title>
>(({ className, ...props }, ref) => (
  <DialogPrimitive.Title
    ref={ref}
    className={cn("text-lg font-semibold leading-none tracking-tight", className)}
    {...props}
  />
));
DialogTitle.displayName = DialogPrimitive.Title.displayName;

const DialogDescription = React.forwardRef<
  React.ElementRef<typeof DialogPrimitive.Description>,
  React.ComponentPropsWithoutRef<typeof DialogPrimitive.Description>
>(({ className, ...props }, ref) => (
  <DialogPrimitive.Description
    ref={ref}
    className={cn("text-sm text-gray-500", className)}
    {...props}
  />
));
DialogDescription.displayName = DialogPrimitive.Description.displayName;
```

```
 97|
 98| export {
 99|   Dialog,
100|   DialogPortal,
101|   DialogOverlay,
102|   DialogClose,
103|   DialogTrigger,
104|   DialogContent,
105|   DialogHeader,
106|   DialogFooter,
107|   DialogTitle,
108|   DialogDescription,
109| };
```

frontend/src/components/ui/dropdown-menu.tsx

94 lines · 3.1 KB

```
"use client";

import * as React from "react";
import * as DropdownMenuPrimitive from "@radix-ui/react-dropdown-menu";
import { cn } from "@/lib/shadcn";

const DropdownMenu = DropdownMenuPrimitive.Root;
const DropdownMenuTrigger = DropdownMenuPrimitive.Trigger;
const DropdownMenuGroup = DropdownMenuPrimitive.Group;
const DropdownMenuPortal = DropdownMenuPrimitive.Portal;
const DropdownMenuSub = DropdownMenuPrimitive.Sub;
const DropdownMenuRadioGroup = DropdownMenuPrimitive.RadioGroup;

const DropdownMenuContent = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.Content>,
  React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Content>
>(({ className, sideOffset = 4, ...props }, ref) => (
  <DropdownMenuPrimitive.Portal>
    <DropdownMenuPrimitive.Content
      ref={ref}
      sideOffset={sideOffset}
      className={cn(
        "z-50 min-w-[8rem] overflow-hidden rounded-lg border border-gray-200 bg-white p-1 shadow-md",
        className,
      )}
      {...props}
    />
  </DropdownMenuPrimitive.Portal>
));
DropdownMenuContent.displayName = DropdownMenuPrimitive.Content.displayName;

const DropdownMenuItem = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.Item>,
  React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Item>
>(({ className, ...props }, ref) => (
  <DropdownMenuPrimitive.Item
    ref={ref}
    className={cn(
      "relative flex cursor-pointer select-none items-center rounded-md px-2 py-1.5 text-sm outline-none transition-colors focus:bg-gray-100 focus:tex
t-gray-900 data-[disabled]:pointer-events-none data-[disabled]:opacity-50",
      className,
    )}
    {...props}
  />
));
DropdownMenuItem.displayName = DropdownMenuPrimitive.Item.displayName;

const DropdownMenuLabel = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.Label>,
  React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Label>
>(({ className, ...props }, ref) => (
  <DropdownMenuPrimitive.Label
    ref={ref}
    className={cn("px-2 py-1.5 text-sm font-semibold text-gray-900", className)}
    {...props}
  />
));
DropdownMenuLabel.displayName = DropdownMenuPrimitive.Label.displayName;

const DropdownMenuSeparator = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.Separator>,
  React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Separator>
>(({ className, ...props }, ref) => (
  <DropdownMenuPrimitive.Separator
    ref={ref}
    className={cn("-mx-1 my-1 h-px bg-gray-200", className)}
    {...props}
  />
));
DropdownMenuSeparator.displayName = DropdownMenuPrimitive.Separator.displayName;

const DropdownMenuShortcut = ({
  className,
  ...props
}: React.HTMLAttributes<HTMLSpanElement>) => (
  <span
    className={cn("ml-auto text-xs tracking-widest text-gray-400", className)}
    {...props}
  />
);
DropdownMenuShortcut.displayName = "DropdownMenuShortcut";

export {
  DropdownMenu,
  DropdownMenuTrigger,
  DropdownMenuContent,
  DropdownMenuItem,
  DropdownMenuLabel,
  DropdownMenuSeparator,
  DropdownMenuShortcut,
  DropdownMenuGroup,
  DropdownMenuPortal,
  DropdownMenuSub,
  DropdownMenuRadioGroup,
};
```

frontend/src/components/ui/input.tsx

23 lines · 0.8 KB

```
import * as React from "react";
import { cn } from "@/lib/shadcn";

export type InputProps = React.InputHTMLAttributes<HTMLInputElement>;

const Input = React.forwardRef<HTMLInputElement, InputProps>(
  ({ className, type, ...props }, ref) => {
    return (
      <input
        type={type}
        className={cn(
          "flex h-10 w-full rounded-lg border border-gray-300 bg-white px-3 py-2 text-sm ring-offset-white file:border-0 file:bg-transparent file:text
-sm file:font-medium placeholder:text-gray-400 focus-visible:outline-none focus-visible:ring-2 focus-visible:ring-brand-600 focus-visible:border-trans
parent disabled:cursor-not-allowed disabled:opacity-50",
          className,
        )}
        ref={ref}
        {...props}
      />
    );
  },
);
Input.displayName = "Input";

export { Input };
```

frontend/src/components/ui/separator.tsx

26 lines · 0.5 KB

```
"use client";

import { cn } from "@/lib/shadcn";

interface SeparatorProps extends React.HTMLAttributes<HTMLDivElement> {
  orientation?: "horizontal" | "vertical";
}

function Separator({
  className,
  orientation = "horizontal",
  ...props
}: SeparatorProps) {
  return (
    <div
      className={cn(
        "shrink-0 bg-gray-200",
        orientation === "horizontal" ? "h-[1px] w-full" : "h-full w-[1px]",
        className,
      )}
      {...props}
    />
  );
}

export { Separator };
```

frontend/src/components/ui/skeleton.tsx

15 lines · 0.3 KB

```
import { cn } from "@/lib/shadcn";

function Skeleton({
  className,
  ...props
}: React.HTMLAttributes<HTMLDivElement>) {
  return (
    <div
      className={cn("animate-pulse rounded-md bg-gray-200", className)}
      {...props}
    />
  );
}

export { Skeleton };
```

frontend/src/components/ui/tabs.tsx

54 lines · 1.8 KB

```
"use client";

import * as React from "react";
import * as TabsPrimitive from "@radix-ui/react-tabs";
import { cn } from "@/lib/shadcn";

const Tabs = TabsPrimitive.Root;

const TabsList = React.forwardRef<
  React.ElementRef<typeof TabsPrimitive.List>,
  React.ComponentPropsWithoutRef<typeof TabsPrimitive.List>
>(({ className, ...props }, ref) => (
  <TabsPrimitive.List
    ref={ref}
    className={cn(
      "inline-flex h-10 items-center justify-center rounded-lg bg-gray-100 p-1 text-gray-500",
      className,
    )}
    {...props}
  />
));
TabsList.displayName = TabsPrimitive.List.displayName;

const TabsTrigger = React.forwardRef<
  React.ElementRef<typeof TabsPrimitive.Trigger>,
  React.ComponentPropsWithoutRef<typeof TabsPrimitive.Trigger>
>(({ className, ...props }, ref) => (
  <TabsPrimitive.Trigger
    ref={ref}
    className={cn(
      "inline-flex items-center justify-center whitespace-nowrap rounded-md px-3 py-1.5 text-sm font-medium transition-all focus-visible:outline-none
focus-visible:ring-2 focus-visible:ring-brand-600 disabled:pointer-events-none disabled:opacity-50 data-[state=active]:bg-white data-[state=active]:te
xt-gray-900 data-[state=active]:shadow-sm",
      className,
    )}
    {...props}
  />
));
TabsTrigger.displayName = TabsPrimitive.Trigger.displayName;

const TabsContent = React.forwardRef<
  React.ElementRef<typeof TabsPrimitive.Content>,
  React.ComponentPropsWithoutRef<typeof TabsPrimitive.Content>
>(({ className, ...props }, ref) => (
  <TabsPrimitive.Content
    ref={ref}
    className={cn(
      "mt-2 focus-visible:outline-none focus-visible:ring-2 focus-visible:ring-brand-600",
      className,
    )}
    {...props}
  />
));
TabsContent.displayName = TabsPrimitive.Content.displayName;

export { Tabs, TabsList, TabsTrigger, TabsContent };
```

frontend/src/components/ui/tooltip.tsx

27 lines · 0.9 KB

```
"use client";

import * as React from "react";
import * as TooltipPrimitive from "@radix-ui/react-tooltip";
import { cn } from "@/lib/shadcn";

const TooltipProvider = TooltipPrimitive.Provider;
const Tooltip = TooltipPrimitive.Root;
const TooltipTrigger = TooltipPrimitive.Trigger;

const TooltipContent = React.forwardRef<
  React.ElementRef<typeof TooltipPrimitive.Content>,
  React.ComponentPropsWithoutRef<typeof TooltipPrimitive.Content>
>(({ className, sideOffset = 4, ...props }, ref) => (
  <TooltipPrimitive.Content
    ref={ref}
    sideOffset={sideOffset}
    className={cn(
      "z-50 overflow-hidden rounded-md bg-gray-900 px-3 py-1.5 text-xs text-white shadow-md animate-in fade-in-0 zoom-in-95",
      className,
    )}
    {...props}
  />
));
TooltipContent.displayName = TooltipPrimitive.Content.displayName;

export { Tooltip, TooltipTrigger, TooltipContent, TooltipProvider };
```

frontend/src/lib/api-client.ts

289 lines · 6.8 KB

```
/** API client — centralized HTTP layer for all backend calls.
 *
 * Every frontend data-fetch goes through this module.
 * Never call fetch() directly in components.
 */

import type { ChartAssemblyInput } from "@/types/chart";
import { getStoredToken } from "@/lib/auth-storage";

const API_BASE = process.env.NEXT_PUBLIC_API_URL ?? "http://localhost:8000/api/v1";

interface RequestOptions {
  method?: string;
  body?: unknown;
}

async function request<T>(path: string, options: RequestOptions = {}): Promise<T> {
  const headers: Record<string, string> = {
    "Content-Type": "application/json",
  };

  const token = getStoredToken();
  if (token) {
    headers["Authorization"] = `Bearer ${token}`;
  }

  const res = await fetch(`${API_BASE}${path}`, {
    method: options.method ?? "GET",
    headers,
    body: options.body ? JSON.stringify(options.body) : undefined,
  });

  if (!res.ok) {
    const error = await res.json().catch(() => ({ message: res.statusText }));
    throw new ApiError(res.status, error.code ?? "UNKNOWN", error.message ?? "Request failed");
  }

  return res.json();
}

export class ApiError extends Error {
  constructor(
    public status: number,
    public code: string,
    message: string,
  ) {
    super(message);
    this.name = "ApiError";
  }
}

// --- Chat ---

export interface ChatRequest {
  query: string;
  conversation_id?: string;
  confirm_large_query?: boolean;
}

export interface ChatResponse {
  conversation_id: string;
  query: string;
  sql?: string;
  sql_explanation?: string;
  chart_spec?: Record<string, unknown>;
  narrative?: string;
  chart_image_base64?: string;
  chart_svg?: string;
  warnings: string[];
  requires_confirmation?: boolean;
  row_estimate?: number | null;
}

export function sendChat(req: ChatRequest): Promise<ChatResponse> {
  return request<ChatResponse>("/chat", { method: "POST", body: req });
}

export function streamChat(
  req: ChatRequest,
  onEvent: (event: Record<string, unknown>) => void,
  onError: (error: Error) => void,
): AbortController {
  const controller = new AbortController();
  const token = getStoredToken();

  fetch(`${API_BASE}/chat/stream`, {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      ...(token ? { Authorization: `Bearer ${token}` } : {}),
    },
    body: JSON.stringify(req),
    signal: controller.signal,
  })
    .then(async (res) => {
      const reader = res.body?.getReader();
      if (!reader) return;

      const decoder = new TextDecoder();
      let buffer = "";
```

```

      while (true) {
        const { done, value } = await reader.read();
        if (done) break;

        buffer += decoder.decode(value, { stream: true });
        const lines = buffer.split("\n");
        buffer = lines.pop() ?? "";

        for (const line of lines) {
          if (line.startsWith("data: ")) {
            try {
              const event = JSON.parse(line.slice(6));
              onEvent(event);
            } catch {
              // Skip unparseable chunks
            }
          }
        }
      }
    })
    .catch(onError);

  return controller;
}

// --- Charts ---

export interface ChartRenderRequest {
  spec: ChartAssemblyInput;
  backend?: string;
  format?: string;
}

export interface ChartRenderResponse {
  success: boolean;
  format: string;
  image_base64?: string;
  svg?: string;
  warnings: string[];
  errors: string[];
}

export function renderChart(req: ChartRenderRequest): Promise<ChartRenderResponse> {
  return request<ChartRenderResponse>("/charts/render", { method: "POST", body: req });
}

// --- Metrics ---

export interface MetricSummary {
  name: string;
  title: string;
  description: string;
  metric_type: string;
  cube_name: string;
  measure_name: string;
  dimensions: string[];
  time_dimensions: string[];
}

export interface MetricListResponse {
  metrics: MetricSummary[];
  count: number;
}

export function listMetrics(): Promise<MetricListResponse> {
  return request<MetricListResponse>("/metrics/list");
}

export interface MetricQueryRequest {
  measures: string[];
  dimensions?: string[];
  time_dimensions?: { dimension: string; granularity: string }[];
  filters?: { member: string; operator: string; values?: string[] }[];
  order?: string[][];
  limit?: number;
  offset?: number;
  timezone?: string;
}

export interface MetricQueryResponse {
  data: Record<string, unknown>[];
  annotation: Record<string, unknown>;
  total: number | null;
  query: Record<string, unknown>;
  latency_ms: number;
  cached: boolean;
}

export function queryMetrics(req: MetricQueryRequest): Promise<MetricQueryResponse> {
  return request<MetricQueryResponse>("/metrics/query", { method: "POST", body: req });
}

export interface DatasourceSummary {
  name: string;
  title: string;
  measures: number;
  dimensions: number;
}

export function listDatasources(): Promise<{ datasources: DatasourceSummary[]; count: number }> {
  return request<{ datasources: DatasourceSummary[]; count: number }>("/datasources");
}

// --- Conversations ---
```

```

export interface ConversationSummary {
  id: string;
  title: string;
  created_at: string;
  updated_at: string;
}

export interface ConversationMessage {
  role: "user" | "assistant";
  content: string;
  generated_sql?: string | null;
  created_at: string;
}

export function listConversations(): Promise<{
  conversations: ConversationSummary[];
  count: number;
}> {
  return request<{ conversations: ConversationSummary[]; count: number }>("/conversations");
}

export function listConversationMessages(
  conversationId: string,
): Promise<{ messages: ConversationMessage[]; count: number }> {
  return request<{ messages: ConversationMessage[]; count: number }>(
    `/conversations/${conversationId}/messages`,
  );
}

// --- Reports ---

export interface ReportSection {
  position: number;
  metric_name: string;
  section_title: string;
  chart_spec: Record<string, unknown>;
  chart_svg?: string | null;
  data_total?: number | null;
  row_count: number;
  narrative?: string | null;
}

export interface Report {
  report_id: string;
  title: string;
  prompt: string;
  summary?: string | null;
  status: string;
  created_at: string;
  sections: ReportSection[];
  warnings: string[];
}

export interface ReportSummary {
  id: string;
  title: string;
  created_at: string;
  section_count: number;
}

export function generateReport(
  prompt: string,
  maxSections = 3,
): Promise<Report> {
  return request<Report>("/reports/generate", {
    method: "POST",
    body: { prompt, max_sections: maxSections },
  });
}

export function listReports(): Promise<{ reports: ReportSummary[]; count: number }> {
  return request<{ reports: ReportSummary[]; count: number }>("/reports");
}

export function getReport(reportId: string): Promise<Report> {
  return request<Report>(`/reports/${reportId}`);
}

// --- Health ---

export function healthCheck(): Promise<{ status: string; version: string }> {
  return request("/health");
}
```

frontend/src/lib/auth-storage.ts

35 lines · 1.0 KB

```
/** Centralized auth storage — single source of truth for session keys.
 *
 * The API client and the auth provider must agree on where the JWT lives;
 * they historically didn't ("token" vs "genbi_token"), which silently broke
 * authenticated API calls after login.
 */

const TOKEN_KEY = "genbi_token";
const USER_KEY = "genbi_user";

export function getStoredToken(): string | null {
  if (typeof window === "undefined") return null;
  return localStorage.getItem(TOKEN_KEY);
}

export function getStoredUser<T>(): T | null {
  if (typeof window === "undefined") return null;
  const raw = localStorage.getItem(USER_KEY);
  if (!raw) return null;
  try {
    return JSON.parse(raw) as T;
  } catch {
    return null;
  }
}

export function storeSession(token: string, user: unknown): void {
  localStorage.setItem(TOKEN_KEY, token);
  localStorage.setItem(USER_KEY, JSON.stringify(user));
}

export function clearSession(): void {
  localStorage.removeItem(TOKEN_KEY);
  localStorage.removeItem(USER_KEY);
}
```

frontend/src/lib/shadcn.ts

86 lines · 1.9 KB

```
/** shadcn/ui component library — generated by shadcn/ui CLI.
 *
 * All shadcn/ui primitives are re-exported here for consistent imports.
 * shadcn/ui components use Radix UI primitives + Tailwind CSS styling.
 */

import { clsx, type ClassValue } from "clsx";
import { twMerge } from "tailwind-merge";

export function cn(...inputs: ClassValue[]) {
  return twMerge(clsx(inputs));
}

// Re-export Radix primitives
export { Slot } from "@radix-ui/react-slot";
export type { SlotProps } from "@radix-ui/react-slot";

// Button
export { Button, buttonVariants } from "@/components/ui/button";
export type { ButtonProps } from "@/components/ui/button";

// Input
export { Input } from "@/components/ui/input";

// Card
export {
  Card,
  CardHeader,
  CardFooter,
  CardTitle,
  CardDescription,
  CardContent,
} from "@/components/ui/card";

// Dialog
export {
  Dialog,
  DialogPortal,
  DialogOverlay,
  DialogClose,
  DialogTrigger,
  DialogContent,
  DialogHeader,
  DialogFooter,
  DialogTitle,
  DialogDescription,
} from "@/components/ui/dialog";

// Dropdown
export {
  DropdownMenu,
  DropdownMenuTrigger,
  DropdownMenuContent,
  DropdownMenuItem,
  DropdownMenuLabel,
  DropdownMenuSeparator,
  DropdownMenuShortcut,
  DropdownMenuGroup,
  DropdownMenuPortal,
  DropdownMenuSub,
  DropdownMenuRadioGroup,
} from "@/components/ui/dropdown-menu";

// Tabs
export { Tabs, TabsList, TabsTrigger, TabsContent } from "@/components/ui/tabs";

// Tooltip
export {
  Tooltip,
  TooltipTrigger,
  TooltipContent,
  TooltipProvider,
} from "@/components/ui/tooltip";

// Avatar
export { Avatar, AvatarImage, AvatarFallback } from "@/components/ui/avatar";

// Separator
export { Separator } from "@/components/ui/separator";

// Badge
export { Badge, badgeVariants } from "@/components/ui/badge";
export type { BadgeProps } from "@/components/ui/badge";

// Skeleton
export { Skeleton } from "@/components/ui/skeleton";
```

frontend/src/lib/validators.ts

180 lines · 5.2 KB

```
import { z } from "zod";

/** Chat request validation */
export const ChatRequestSchema = z.object({
  query: z.string().min(1, "Query cannot be empty").max(2000, "Query too long"),
  conversation_id: z.string().uuid().optional(),
  confirm_large_query: z.boolean().optional(),
});

export type ChatRequest = z.infer<typeof ChatRequestSchema>;

/** Chat response validation */
export const ChatResponseSchema = z.object({
  conversation_id: z.string(),
  query: z.string(),
  sql: z.string().optional(),
  sql_explanation: z.string().optional(),
  chart_spec: z.record(z.unknown()).optional(),
  narrative: z.string().optional(),
  chart_image_base64: z.string().optional(),
  chart_svg: z.string().optional(),
  warnings: z.array(z.string()),
});

export type ChatResponse = z.infer<typeof ChatResponseSchema>;

/** SSE event validation */
export const SSEEventSchema = z.object({
  event: z.enum([
    "start", "intent", "sql", "validation", "data",
    "chart", "narrative", "done",
  ]),
  conversation_id: z.string().optional(),
  intent: z.string().optional(),
  plan: z.array(z.record(z.unknown())).optional(),
  sql: z.string().nullable().optional(),
  valid: z.boolean().optional(),
  validated_sql: z.string().nullable().optional(),
  row_count: z.number().optional(),
  preview: z.array(z.record(z.unknown())).optional(),
  chart_spec: z.record(z.unknown()).optional(),
  image_base64: z.string().optional(),
  svg: z.string().optional(),
  narrative: z.string().nullable().optional(),
  status: z.string().optional(),
  error: z.string().optional(),
  warnings: z.array(z.string()).optional(),
  session_id: z.string().optional(),
  requires_confirmation: z.boolean().optional(),
  row_estimate: z.number().nullable().optional(),
});

export type SSEEvent = z.infer<typeof SSEEventSchema>;

/** Metric list response (GET /metrics/list) */
export const MetricSummarySchema = z.object({
  name: z.string(),
  title: z.string(),
  description: z.string(),
  metric_type: z.string(),
  cube_name: z.string(),
  measure_name: z.string(),
  dimensions: z.array(z.string()),
  time_dimensions: z.array(z.string()),
});

export type MetricSummary = z.infer<typeof MetricSummarySchema>;

export const MetricListResponseSchema = z.object({
  metrics: z.array(MetricSummarySchema),
  count: z.number(),
});

export type MetricListResponse = z.infer<typeof MetricListResponseSchema>;

/** Metric query (POST /metrics/query) */
export const MetricQueryRequestSchema = z.object({
  measures: z.array(z.string()).min(1).max(5),
  dimensions: z.array(z.string()).max(5).optional(),
  time_dimensions: z
    .array(
      z.object({
        dimension: z.string(),
        granularity: z.enum(["day", "week", "month", "quarter", "year", "hour"]),
      }),
    )
    .max(3)
    .optional(),
  filters: z
    .array(
      z.object({
        member: z.string(),
        operator: z.string(),
        values: z.array(z.string()).optional(),
      }),
    )
    .optional(),
  order: z.array(z.array(z.string())).optional(),
  limit: z.number().int().min(1).max(1000).optional(),
  offset: z.number().int().min(0).optional(),
```

```
  timezone: z.string().optional(),
});

export type MetricQueryRequest = z.infer<typeof MetricQueryRequestSchema>;

export const MetricQueryResponseSchema = z.object({
  data: z.array(z.record(z.unknown())),
  annotation: z.record(z.unknown()),
  total: z.number().nullable(),
  query: z.record(z.unknown()),
  latency_ms: z.number(),
  cached: z.boolean(),
});

export type MetricQueryResponse = z.infer<typeof MetricQueryResponseSchema>;

/** Chart AssemblyInput validation (Flint schema) */
export const ChartAssemblyInputSchema = z.object({
  chartType: z.string().min(1),
  encodings: z.record(
    z.object({
      field: z.string(),
      title: z.string().optional(),
    }),
  ),
  baseSize: z.object({
    width: z.number().int().positive(),
    height: z.number().int().positive(),
  }),
  semantic_types: z
    .record(z.enum(["Category", "Quantity", "Temporal"]))
    .optional(),
  data: z.object({
    values: z.array(z.record(z.unknown())).optional(),
    url: z.string().url().optional(),
  }),
});

export type ChartAssemblyInput = z.infer<typeof ChartAssemblyInputSchema>;

/** Metric definition validation */
export const MetricDefinitionSchema = z.object({
  name: z.string(),
  title: z.string(),
  description: z.string(),
  metric_type: z.enum([
    "sum", "count", "count_distinct", "avg",
    "min", "max", "ratio", "derived", "running_total",
  ]),
  cube_name: z.string(),
  measure_name: z.string(),
  dimensions: z.array(z.string()),
  time_dimensions: z.array(z.string()),
  format: z.record(z.unknown()).nullable().optional(),
});

export type MetricDefinition = z.infer<typeof MetricDefinitionSchema>;

/** Login request validation */
export const LoginRequestSchema = z.object({
  email: z.string().email("Invalid email address"),
  password: z.string().min(6, "Password must be at least 6 characters"),
});

export type LoginRequest = z.infer<typeof LoginRequestSchema>;

/** Login response */
export const LoginResponseSchema = z.object({
  access_token: z.string(),
  token_type: z.string(),
  user: z.object({
    id: z.string(),
    email: z.string(),
    name: z.string(),
    tenant_id: z.string(),
    roles: z.array(z.string()),
  }),
});

export type LoginResponse = z.infer<typeof LoginResponseSchema>;
```

frontend/src/types/chart.ts

16 lines · 0.5 KB

```
/** TypeScript types for Flint Chart integration. */

export interface ChartAssemblyInput {
  chartType: string;
  encodings: Record<string, { field: string }>;
  baseSize: { width: number; height: number };
  semantic_types?: Record<string, "Category" | "Quantity" | "Temporal">;
  data: {
    values?: Record<string, unknown>[];
    url?: string;
  };
}

export type ChartBackend = "vegalite" | "echarts" | "chartjs";

export type ChartOutputFormat = "png" | "svg";
```

frontend/src/types/index.ts

1 lines · 0.1 KB

```
1 export type { ChartAssemblyInput, ChartBackend, ChartOutputFormat } from "./chart";
```

frontend/tailwind.config.ts

26 lines · 0.5 KB

```
import type { Config } from "tailwindcss";

const config: Config = {
  content: ["./src/**/*.{js,ts,jsx,tsx,mdx}"],
  theme: {
    extend: {
      colors: {
        brand: {
          50: "#eef2ff",
          100: "#dbe4ff",
          200: "#bac8ff",
          300: "#91a7ff",
          400: "#748ffc",
          500: "#5c7cfa",
          600: "#4c6ef5",
          700: "#4263eb",
          800: "#3b5bdb",
          900: "#364fc7",
        },
      },
    },
  },
  plugins: [],
};

export default config;
```

frontend/tsconfig.json

23 lines · 0.6 KB

```
{
  "compilerOptions": {
    "target": "ES2022",
    "lib": ["dom", "dom.iterable", "esnext"],
    "allowJs": true,
    "skipLibCheck": true,
    "strict": true,
    "noEmit": true,
    "esModuleInterop": true,
    "module": "esnext",
    "moduleResolution": "bundler",
    "resolveJsonModule": true,
    "isolatedModules": true,
    "jsx": "preserve",
    "incremental": true,
    "plugins": [{"name": "next"}],
    "paths": {
      "@/*": ["./src/*"]
    }
  },
  "include": ["next-env.d.ts", "**/*.ts", "**/*.tsx", ".next/types/**/*.ts"],
  "exclude": ["node_modules"]
}
```

infra/docker-compose.dev.yml

142 lines · 4.1 KB

```yaml
services:
  # ======================================================================
  # Core Services
  # ======================================================================
  postgres:
    build:
      context: ./postgres
      dockerfile: Dockerfile
    image: genbi/postgres-pgvector-age:pg16
    container_name: genbi-postgres
    environment:
      POSTGRES_USER: genbi
      POSTGRES_PASSWORD: genbi
      POSTGRES_DB: genbi
      # Opt out of AGE at boot if you don't need lineage (dev convenience).
      GENBI_ENABLE_AGE: ${GENBI_ENABLE_AGE:-true}
    ports:
      - "5432:5432"
    volumes:
      - postgres_data:/var/lib/postgresql/data
      - ./postgres/init.sql:/docker-entrypoint-initdb.d/init.sql
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U genbi"]
      interval: 5s
      timeout: 5s
      retries: 5

  redis:
    image: redis:7-alpine
    container_name: genbi-redis
    ports:
      - "6379:6379"
    healthcheck:
      test: ["CMD", "redis-cli", "ping"]
      interval: 5s
      timeout: 5s
      retries: 5

  # ======================================================================
  # Application Services
  # ======================================================================
  backend:
    build:
      context: ..          # repo root, so .claude/prompts is in-context
      dockerfile: backend/Dockerfile
    container_name: genbi-backend
    env_file:
      - ../backend/.env
    ports:
      - "8000:8000"
    depends_on:
      postgres:
        condition: service_healthy
      redis:
        condition: service_healthy
    volumes:
      - ../backend:/app
      - ../scripts:/app/scripts:ro
      - ../.claude/prompts:/app/.claude/prompts:ro
      - flint_data:/tmp/genbi-charts
    healthcheck:
      test: ["CMD-SHELL", "python -c \"import urllib.request,sys; sys.exit(0 if urllib.request.urlopen('http://localhost:8000/api/v1/health').status==200 else 1)\""]
      interval: 15s
      timeout: 5s
      retries: 5
      start_period: 30s
    # NOTE: --reload removed — WatchFiles over a Windows bind mount (host
    # Z:\GITHUB\GBI\backend:/app) scans .venv too and the worker never starts.
    # Backend code changes: `docker compose restart backend`.
    command: uv run uvicorn app.main:app --host 0.0.0.0 --port 8000

  frontend:
    build:
      context: ../frontend
      dockerfile: Dockerfile
      # Dev runs from the builder stage — it has pnpm + full node_modules;
      # the production stage is a standalone bundle with neither.
      target: builder
    container_name: genbi-frontend
    environment:
      NEXT_PUBLIC_API_URL: http://localhost:8000/api/v1
    ports:
      - "3000:3000"
    depends_on:
      backend:
        condition: service_healthy
    volumes:
      - ../frontend:/app
      - /app/node_modules
    command: pnpm dev

  # ======================================================================
  # Semantic Layer
  # ======================================================================
  cube:
    image: cubejs/cube:latest
    container_name: genbi-cube
    env_file:
      - ../semantic/cube/.env
```

```
    ports:
      - "4000:4000"
    volumes:
      - ../semantic/cube:/cube/conf
    depends_on:
      postgres:
        condition: service_healthy
    healthcheck:
      test: ["CMD-SHELL", "node -e \"fetch('http://localhost:4000/').then(r=>process.exit(r.ok?0:1)).catch(()=>process.exit(1))\""]
      interval: 15s
      timeout: 5s
      retries: 5
      start_period: 30s

  # ======================================================================
  # Observability
  # ======================================================================
  prometheus:
    image: prom/prometheus:latest
    container_name: genbi-prometheus
    ports:
      - "9090:9090"
    volumes:
      - ./prometheus/prometheus.yml:/etc/prometheus/prometheus.yml

  grafana:
    image: grafana/grafana:latest
    container_name: genbi-grafana
    ports:
      - "3001:3000"
    depends_on:
      - prometheus

  otel_collector:
    image: otel/opentelemetry-collector-contrib:latest
    container_name: genbi-otel
    ports:
      - "4318:4318"
      - "8888:8888"

volumes:
  postgres_data:
  flint_data:
```

infra/docker-compose.yml

76 lines · 2.0 KB

```yaml
services:
  postgres:
    build:
      context: ./postgres
      dockerfile: Dockerfile
    image: genbi/postgres-pgvector-age:pg16
    container_name: genbi-postgres-prod
    environment:
      POSTGRES_USER: ${POSTGRES_USER:-genbi}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
      POSTGRES_DB: ${POSTGRES_DB:-genbi}
      GENBI_ENABLE_AGE: "true"
    ports:
      - "5432:5432"
    volumes:
      - postgres_data:/var/lib/postgresql/data
      - ./postgres/init.sql:/docker-entrypoint-initdb.d/init.sql
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U ${POSTGRES_USER:-genbi}"]
      interval: 10s
      timeout: 5s
      retries: 5
    restart: unless-stopped

  redis:
    image: redis:7-alpine
    container_name: genbi-redis-prod
    ports:
      - "6379:6379"
    healthcheck:
      test: ["CMD", "redis-cli", "ping"]
      interval: 10s
      timeout: 5s
      retries: 5
    restart: unless-stopped

  backend:
    build:
      context: ..          # repo root, so .claude/prompts is in-context
      dockerfile: backend/Dockerfile
    container_name: genbi-backend-prod
    env_file:
      - ../backend/.env
    ports:
      - "8000:8000"
    depends_on:
      postgres:
        condition: service_healthy
      redis:
        condition: service_healthy
    healthcheck:
      test: ["CMD-SHELL", "python -c \"import urllib.request,sys; sys.exit(0 if urllib.request.urlopen('http://localhost:8000/api/v1/health').status==200 else 1)\""]
      interval: 15s
      timeout: 5s
      retries: 5
      start_period: 30s
    restart: unless-stopped
    command: uv run gunicorn app.main:app --workers 4 --worker-class uvicorn.workers.UvicornWorker --bind 0.0.0.0:8000

  frontend:
    build:
      context: ../frontend
      dockerfile: Dockerfile
      target: production
    container_name: genbi-frontend-prod
    environment:
      NEXT_PUBLIC_API_URL: ${API_URL:-http://localhost:8000/api/v1}
    ports:
      - "3000:3000"
    depends_on:
      backend:
        condition: service_healthy
    restart: unless-stopped

volumes:
  postgres_data:
```

infra/postgres/Dockerfile

46 lines · 1.8 KB

```
# PostgreSQL 16 with BOTH pgvector and Apache AGE.
#
# The official pgvector image ships pgvector only; Apache AGE is absent, so
# `CREATE EXTENSION age` in init.sql fails on first boot. This image layers
# AGE on top of pgvector so both extensions coexist (required for the
# lineage graph in backend/app/db/graph_schema.py).
#
# Base image is Debian 12 (bookworm) — use apt-get, NOT apk (Alpine).
#
# Build pin: AGE is built from source against this image's PG headers. Tag
# advances with AGE releases — bump together with the AGE_SUPERUSER role grant
# in infra/postgres/init.sql.
FROM pgvector/pgvector:pg16

# Build deps for AGE, then compile AGE against the bundled PostgreSQL, then
# strip the build deps to keep the image small. pg_config is at /usr/bin on
# the pgvector Debian image (NOT /usr/local/bin as on Alpine variants).
#
# postgresql-server-dev-16 is required: the pgvector base image is a runtime
# image and does not ship postgres.h / the server headers AGE compiles against.
RUN apt-get update \
    && apt-get install -y --no-install-recommends \
        build-essential \
        clang \
        llvm-15-dev \
        libllvm15 \
        postgresql-server-dev-16 \
        git \
        ca-certificates \
        libreadline-dev \
        zlib1g-dev \
        flex \
        bison \
    && git clone --depth 1 --branch PG16 https://github.com/apache/age /age-src \
    && cd /age-src \
    && make PG_CONFIG=$(command -v pg_config) install \
    && cd / \
    && rm -rf /age-src \
    && apt-get purge -y build-essential clang llvm-15-dev libllvm15 \
        postgresql-server-dev-16 flex bison \
    && apt-get autoremove -y \
    && rm -rf /var/lib/apt/lists/*

# Verify the .so is installed so the image fails fast at build time if AGE
# ever stops shipping a PG16 branch.
RUN test -f $(pg_config --pkglibdir)/age.so
```

infra/postgres/init.sql

313 lines · 13.9 KB

```sql
-- ==============================================================================
-- GenBI Database Initialization
-- ==============================================================================
-- This script runs once when the PostgreSQL container is first created.
-- It sets up extensions, roles, core tables, and row-level security.
-- WARNING: destructive if run on an existing database.
--
-- Ownership split (see docs/adr/005-age-and-pgvector-image.md):
--   - This script owns: extensions, roles, the seed tenant, RLS policies.
--   - Alembic owns: table DDL (CREATE TABLE / ALTER TABLE).
--   - seed_test_data.py owns: INSERT only (no DDL).

-- ------------------------------------------------------------------------------
-- Extensions
-- ------------------------------------------------------------------------------
CREATE EXTENSION IF NOT EXISTS "pgcrypto";
CREATE EXTENSION IF NOT EXISTS "vector";

-- Apache AGE for the lineage graph. Optional: the image
-- (infra/postgres/Dockerfile) layers AGE onto pgvector. Guard so a missing
-- extension degrades gracefully instead of crashing DB init.
DO $$
BEGIN
    IF EXISTS (
        SELECT 1 FROM pg_available_extensions WHERE name = 'age'
    ) THEN
        CREATE EXTENSION IF NOT EXISTS age;
        LOAD 'age';
        SET search_path = ag_catalog, "$user", public;
        -- Create the lineage graph (guarded — create_graph raises if it exists)
        BEGIN
            SELECT * FROM ag_catalog.create_graph('genbi_graph');
        EXCEPTION WHEN OTHERS THEN
            RAISE NOTICE 'genbi_graph already exists or creation deferred.';
        END;
        RAISE NOTICE 'Apache AGE extension enabled.';
    ELSE
        RAISE NOTICE 'Apache AGE not available — lineage graph disabled (GENBI_ENABLE_AGE controls this).';
    END IF;
END $$;

-- Reset search_path for the rest of the script (AGE sets it above)
SET search_path = "$user", public;

-- ------------------------------------------------------------------------------
-- Roles
-- ------------------------------------------------------------------------------

-- Read-only role for the Cube.dev semantic layer (headless BI must never write).
DO $$
BEGIN
    IF NOT EXISTS (SELECT 1 FROM pg_roles WHERE rolname = 'cube_reader') THEN
        CREATE ROLE cube_reader LOGIN PASSWORD 'changeme_cube_reader';
    END IF;
END $$;
GRANT USAGE ON SCHEMA public TO cube_reader;
-- Tables granted after they are created (see Alembic baseline migration).
-- Default privilege so future tables in public are readable by Cube:
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO cube_reader;

-- Application runtime roles (see docs/adr/006-enforced-rls-roles.md and
-- Alembic revision 0002_app_roles — the authoritative mirror of this block).
--   genbi_app  — backend runtime (ORM + query connector); bound by RLS.
--   genbi_auth — login endpoint only; reads `users` across tenants.
-- Passwords are fixed dev defaults — rotate in production.
DO $$
BEGIN
    IF NOT EXISTS (SELECT 1 FROM pg_roles WHERE rolname = 'genbi_app') THEN
        CREATE ROLE genbi_app LOGIN PASSWORD 'genbi_app';
    END IF;
    IF NOT EXISTS (SELECT 1 FROM pg_roles WHERE rolname = 'genbi_auth') THEN
        CREATE ROLE genbi_auth LOGIN PASSWORD 'genbi_auth';
    END IF;
END $$;

-- AGE: grant the application role access to the graph catalog (dev only).
-- In prod, prefer a dedicated age_admin role.
DO $$
BEGIN
    IF EXISTS (SELECT 1 FROM pg_extension WHERE extname = 'age') THEN
        GRANT USAGE ON SCHEMA ag_catalog TO PUBLIC;
        GRANT ALL ON SCHEMA ag_catalog TO PUBLIC;
        RAISE NOTICE 'AGE catalog grants applied.';
    END IF;
END $$;

-- ------------------------------------------------------------------------------
-- Core Tables
-- ------------------------------------------------------------------------------

CREATE TABLE IF NOT EXISTS tenants (
    id          UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name        VARCHAR(255) UNIQUE NOT NULL,
    created_at  TIMESTAMPTZ NOT NULL DEFAULT NOW(),
    updated_at  TIMESTAMPTZ NOT NULL DEFAULT NOW()
);

-- Seed a default tenant
INSERT INTO tenants (id, name) VALUES
    ('00000000-0000-0000-0000-000000000001', 'default')
```

```
ON CONFLICT (id) DO NOTHING;

CREATE TABLE IF NOT EXISTS users (
    id          UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    tenant_id   UUID NOT NULL REFERENCES tenants(id),
    email       VARCHAR(255) NOT NULL,
    hashed_password VARCHAR(255) NOT NULL,
    roles       JSONB NOT NULL DEFAULT '["user"]',
    created_at  TIMESTAMPTZ NOT NULL DEFAULT NOW(),
    updated_at  TIMESTAMPTZ NOT NULL DEFAULT NOW(),
    UNIQUE(tenant_id, email)
);

-- Dev bootstrap user (DEV-ONLY — remove before production): admin@genbi.local / admin123
INSERT INTO users (id, tenant_id, email, hashed_password, roles) VALUES
    ('00000000-0000-0000-0000-000000000101', '00000000-0000-0000-0000-000000000001',
     'admin@genbi.local', '$2b$12$DMNWR/XfEsbJ2InS2iNLa.pXvy.NsZetS.Im0XppBpTOBwY6Brnyi',
     '["admin","user"]')
ON CONFLICT (tenant_id, email) DO NOTHING;

CREATE TABLE IF NOT EXISTS audit_log (
    id                  UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    session_id          UUID NOT NULL,
    user_id             UUID NOT NULL,
    tenant_id           UUID NOT NULL REFERENCES tenants(id),
    input_prompt_hash   VARCHAR(64) NOT NULL,
    generated_sql       TEXT,
    model_name          VARCHAR(100) NOT NULL,
    model_version       VARCHAR(50) NOT NULL,
    input_tokens        INTEGER NOT NULL DEFAULT 0,
    output_tokens       INTEGER NOT NULL DEFAULT 0,
    latency_ms          DOUBLE PRECISION NOT NULL DEFAULT 0,
    feedback_score      INTEGER,
    hallucination_flag  BOOLEAN NOT NULL DEFAULT FALSE,
    created_at          TIMESTAMPTZ NOT NULL DEFAULT NOW()
);

CREATE INDEX IF NOT EXISTS idx_audit_log_session ON audit_log(session_id);
CREATE INDEX IF NOT EXISTS idx_audit_log_tenant ON audit_log(tenant_id);
CREATE INDEX IF NOT EXISTS idx_audit_log_created ON audit_log(created_at);

CREATE TABLE IF NOT EXISTS conversations (
    id          UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id     UUID NOT NULL,
    tenant_id   UUID NOT NULL REFERENCES tenants(id),
    title       VARCHAR(500),
    created_at  TIMESTAMPTZ NOT NULL DEFAULT NOW(),
    updated_at  TIMESTAMPTZ NOT NULL DEFAULT NOW()
);

CREATE INDEX IF NOT EXISTS idx_conversations_tenant ON conversations(tenant_id);

-- Chat turns (Phase 14): one row per user/assistant message. Mirrors
-- Alembic 0004_messages (the authoritative copy).
CREATE TABLE IF NOT EXISTS messages (
    id              UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    conversation_id UUID NOT NULL REFERENCES conversations(id),
    tenant_id       UUID NOT NULL REFERENCES tenants(id),
    role            VARCHAR(20) NOT NULL,
    content         TEXT NOT NULL,
    generated_sql   TEXT,
    created_at      TIMESTAMPTZ NOT NULL DEFAULT NOW()
);

CREATE INDEX IF NOT EXISTS idx_messages_conversation ON messages(conversation_id);
CREATE INDEX IF NOT EXISTS idx_messages_tenant ON messages(tenant_id);

-- Reports (Phase 16): LLM-planned multi-chart reports. Mirrors Alembic
-- 0005_reports (the authoritative copy).
CREATE TABLE IF NOT EXISTS reports (
    id          UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    tenant_id   UUID NOT NULL REFERENCES tenants(id),
    user_id     UUID NOT NULL,
    prompt      TEXT NOT NULL,
    title       VARCHAR(500) NOT NULL,
    summary     TEXT,
    status      VARCHAR(20) NOT NULL DEFAULT 'complete',
    created_at  TIMESTAMPTZ NOT NULL DEFAULT NOW()
);

CREATE INDEX IF NOT EXISTS idx_reports_tenant ON reports(tenant_id);
CREATE INDEX IF NOT EXISTS idx_reports_user ON reports(user_id);

CREATE TABLE IF NOT EXISTS report_sections (
    id            UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    report_id     UUID NOT NULL REFERENCES reports(id),
    tenant_id     UUID NOT NULL REFERENCES tenants(id),
    position      INTEGER NOT NULL DEFAULT 0,
    metric_name   VARCHAR(255) NOT NULL,
    section_title VARCHAR(500) NOT NULL,
    chart_spec    JSONB NOT NULL DEFAULT '{}',
    chart_svg     TEXT,
    data_total    DOUBLE PRECISION,
    row_count     INTEGER NOT NULL DEFAULT 0,
    narrative     TEXT,
    created_at    TIMESTAMPTZ NOT NULL DEFAULT NOW()
);

CREATE INDEX IF NOT EXISTS idx_report_sections_report ON report_sections(report_id);
CREATE INDEX IF NOT EXISTS idx_report_sections_tenant ON report_sections(tenant_id);

-- ---------------------------------------------------------------------------
-- Schema Embeddings (for NL2SQL semantic search)
-- ---------------------------------------------------------------------------
-- Expanded to match scripts/embed_schema.py: one row per TABLE (not per
```

```
-- column), holding the full column JSON and the text that was embedded.
-- The embedding is generated from build_embedding_text() in that script.

CREATE TABLE IF NOT EXISTS schema_embeddings (
    id                  UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    tenant_id           UUID NOT NULL REFERENCES tenants(id),
    table_schema        VARCHAR(255) NOT NULL DEFAULT 'public',
    table_name          VARCHAR(255) NOT NULL,
    full_name           VARCHAR(512) NOT NULL,
    table_description   TEXT NOT NULL DEFAULT '',
    columns_json        JSONB NOT NULL DEFAULT '[]',
    embedding_text      TEXT NOT NULL DEFAULT '',
    embedding           VECTOR(1536),
    created_at          TIMESTAMPTZ NOT NULL DEFAULT NOW(),
    updated_at          TIMESTAMPTZ NOT NULL DEFAULT NOW(),
    UNIQUE(tenant_id, table_schema, table_name)
);

CREATE INDEX IF NOT EXISTS idx_schema_embeddings_lookup
    ON schema_embeddings USING ivfflat (embedding vector_cosine_ops)
    WITH (lists = 100);
CREATE INDEX IF NOT EXISTS idx_schema_embeddings_tenant
    ON schema_embeddings(tenant_id);

-- ---------------------------------------------------------------------------
-- Agent Examples (few-shot prompts)
-- ---------------------------------------------------------------------------

CREATE TABLE IF NOT EXISTS agent_examples (
    id              UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    agent_name      VARCHAR(100) NOT NULL,
    nl_query        TEXT NOT NULL,
    expected_sql    TEXT NOT NULL,
    tags            JSONB DEFAULT '[]',
    embedding       VECTOR(1536),
    tenant_id       UUID NOT NULL REFERENCES tenants(id),
    created_at      TIMESTAMPTZ NOT NULL DEFAULT NOW()
);

CREATE INDEX IF NOT EXISTS idx_agent_examples_agent ON agent_examples(agent_name);
CREATE INDEX IF NOT EXISTS idx_agent_examples_tenant ON agent_examples(tenant_id);

-- ---------------------------------------------------------------------------
-- Row-Level Security
-- ---------------------------------------------------------------------------
-- Every tenant-scoped table gets a tenant_isolation policy. The session GUC
-- `app.current_tenant_id` is set per-transaction by PostgreSQLConnector from
-- the JWT claim. FORCE is applied so even the table OWNER is subject to RLS
-- (otherwise the owner bypasses the policy and isolation is ineffective).
--
-- NOTE: tenants(id) itself is NOT enrolled — it is a shared control table.

ALTER TABLE users              ENABLE ROW LEVEL SECURITY;
ALTER TABLE audit_log          ENABLE ROW LEVEL SECURITY;
ALTER TABLE conversations      ENABLE ROW LEVEL SECURITY;
ALTER TABLE schema_embeddings  ENABLE ROW LEVEL SECURITY;
ALTER TABLE agent_examples     ENABLE ROW LEVEL SECURITY;

ALTER TABLE users              FORCE ROW LEVEL SECURITY;
ALTER TABLE audit_log          FORCE ROW LEVEL SECURITY;
ALTER TABLE conversations      FORCE ROW LEVEL SECURITY;
ALTER TABLE messages           ENABLE ROW LEVEL SECURITY;
ALTER TABLE messages           FORCE ROW LEVEL SECURITY;
ALTER TABLE reports            ENABLE ROW LEVEL SECURITY;
ALTER TABLE reports            FORCE ROW LEVEL SECURITY;
ALTER TABLE report_sections    ENABLE ROW LEVEL SECURITY;
ALTER TABLE report_sections    FORCE ROW LEVEL SECURITY;
ALTER TABLE schema_embeddings  FORCE ROW LEVEL SECURITY;
ALTER TABLE agent_examples     FORCE ROW LEVEL SECURITY;

CREATE POLICY tenant_isolation ON users
    USING (tenant_id = current_setting('app.current_tenant_id', true)::UUID);
CREATE POLICY tenant_isolation ON audit_log
    USING (tenant_id = current_setting('app.current_tenant_id', true)::UUID);
CREATE POLICY tenant_isolation ON conversations
    USING (tenant_id = current_setting('app.current_tenant_id', true)::UUID);
CREATE POLICY tenant_isolation ON messages
    USING (tenant_id = current_setting('app.current_tenant_id', true)::UUID);
CREATE POLICY tenant_isolation ON reports
    USING (tenant_id = current_setting('app.current_tenant_id', true)::UUID);
CREATE POLICY tenant_isolation ON report_sections
    USING (tenant_id = current_setting('app.current_tenant_id', true)::UUID);
CREATE POLICY tenant_isolation ON schema_embeddings
    USING (tenant_id = current_setting('app.current_tenant_id', true)::UUID);
CREATE POLICY tenant_isolation ON agent_examples
    USING (tenant_id = current_setting('app.current_tenant_id', true)::UUID);

-- ---------------------------------------------------------------------------
-- Application Role Grants + Login-Lookup Policy
-- ---------------------------------------------------------------------------
-- Mirrors Alembic revision 0002_app_roles (the authoritative copy). The
-- migration re-applies this block idempotently, so fresh containers (this
-- file) and existing volumes (alembic) converge on the same state.

GRANT USAGE ON SCHEMA public TO genbi_app, genbi_auth;
GRANT SELECT ON tenants TO genbi_app;
GRANT SELECT, INSERT, UPDATE, DELETE ON
    users, audit_log, conversations, messages,
    reports, report_sections, schema_embeddings, agent_examples
    TO genbi_app;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO genbi_app;
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO genbi_app;

-- genbi_auth: SELECT on users ONLY — the single RLS carve-out a credential
-- lookup requires. Scoped TO the role so it widens nothing else.
```

```
GRANT SELECT ON users TO genbi_auth;
CREATE POLICY users_login_lookup ON users
    FOR SELECT TO genbi_auth USING (true);
```

infra/prometheus/prometheus.yml

11 lines · 0.2 KB

```yaml
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: "genbi-backend"
    static_configs:
      - targets: ["backend:8000"]

  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]
```

LICENSE

21 lines · 1.0 KB


```
MIT License

Copyright (c) 2026 Iven Kwan

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

Makefile

56 lines · 2.0 KB

```
# GenBI — development task runner.
# Run `make help` for the target list. All targets are idempotent unless noted.

.DEFAULT_GOAL := help
COMPOSE_DEV := docker compose -f infra/docker-compose.dev.yml
COMPOSE_PROD := docker compose -f infra/docker-compose.yml

.PHONY: help setup up down restart logs ps verify secrets migrate seed reset clean deps

help: ## Show this help
 @awk 'BEGIN {FS = ":.*##"; printf "Usage: make \033[36m<target>\033[0m\n\n"} \
      /^[a-zA-Z_-]+:.*?##/ { printf "  \033[36m%-12s\033[0m %s\n", $$1, $$2 }' $(MAKEFILE_LIST)

setup: ## First-time setup: check prereqs, gen secrets, install deps, build
 @scripts/setup.sh

up: ## Start the dev stack (detached)
 @$(COMPOSE_DEV) up -d
 @echo "Stack starting. Run 'make verify' once services are healthy."

down: ## Stop the dev stack (keeps volumes/data)
 @$(COMPOSE_DEV) down

restart: ## Restart the dev stack
 @$(COMPOSE_DEV) restart

logs: ## Tail logs for all services (Ctrl-C to exit)
 @$(COMPOSE_DEV) logs -f --tail=100

ps: ## Show service status + health
 @$(COMPOSE_DEV) ps

verify: ## Run smoke checks against the running stack
 @scripts/verify.sh

secrets: ## Regenerate .env files with fresh random secrets
 @scripts/gen-env.sh --force

migrate: ## Apply Alembic migrations inside the backend container
 @$(COMPOSE_DEV) exec -T backend uv run alembic upgrade head

seed: ## Load synthetic test data (requires `make migrate` first — tables come from Alembic 0003)
 @$(COMPOSE_DEV) exec -T -e PYTHONPATH=/app backend uv run python scripts/seed_test_data.py

reset: ##    NUKE all data and restart (down -v && up)
 @$(COMPOSE_DEV) down -v
 @$(COMPOSE_DEV) up -d

clean: ## Remove build artifacts, venvs, node_modules, Docker volumes
 @$(COMPOSE_DEV) down -v 2>/dev/null || true
 rm -rf backend/.venv backend/uv.lock frontend/node_modules frontend/.next
 @echo "Cleaned. Run 'make setup' to rebuild."

deps: ## Install host-side dev dependencies (backend uv + frontend pnpm)
 cd backend && uv sync --dev
 cd frontend && pnpm install
```

pyproject.toml

69 lines · 1.7 KB

```toml
[project]
name = "genbi"
version = "0.1.0"
description = "Generative BI Platform — NL2SQL, auto-visualization, narrative insights"
requires-python = ">=3.12"
license = {text = "MIT"}
authors = [
    {name = "Iven Kwan"}
]
keywords = ["bi", "nl2sql", "generative-ai", "data-visualization", "agentic"]

[tool.uv.workspace]
members = ["backend"]

[tool.uv]
dev-dependencies = [
    "pytest>=8.3",
    "pytest-asyncio>=0.24",
    "pytest-cov>=5.0",
    "ruff>=0.6",
    "mypy>=1.11",
    "pre-commit>=3.8",
]

[tool.ruff]
target-version = "py312"
line-length = 100

[tool.ruff.lint]
select = [
    "E",   # pycodestyle errors
    "W",   # pycodestyle warnings
    "F",   # pyflakes
    "I",   # isort
    "N",   # pep8-naming
    "UP",  # pyupgrade
    "B",   # flake8-bugbear
    "SIM", # flake8-simplify
]
ignore = ["E501"]  # line-too-long handled by formatter

[tool.ruff.lint.flake8-bugbear]
extend-immutable-calls = ["fastapi.Depends"]

[tool.ruff.lint.per-file-ignores]
# N815: chartType/baseSize are the Flint chart-spec JSON keys — must keep casing.
"backend/app/models/chart.py" = ["N815"]
# F841: ingest_table's cypher execute is side-effect-only (result intentionally
# unread). Parameterizing the AGE cypher queries is a tracked follow-up.
"backend/app/db/graph_schema.py" = ["F841"]

[tool.ruff.format]
quote-style = "double"
indent-style = "space"

[tool.mypy]
python_version = "3.12"
strict = true
warn_return_any = true
warn_unused_ignores = true

[tool.pytest.ini_options]
asyncio_mode = "auto"
testpaths = ["backend/tests"]
markers = [
    "unit: unit tests (fast, no external services)",
    "integration: integration tests (requires test DB/Redis)",
    "e2e: end-to-end tests (full pipeline, slow)",
]
```

README.md

411 lines · 18.3 KB

```
# GenBI — Generative BI Platform

**Natural language → SQL → chart + narrative. Governed, explainable, multi-tenant.**

GenBI lets business users query enterprise data warehouses in plain English and
returns a validated SQL query, a rendered chart, and a written insight — all in
a single interaction, with row-level security, PII masking, and full audit
tracing enforced on every request.

> **Status:** actively developed scaffold. The agent pipeline, semantic layer,
> and governance controls are implemented; see [Project Status](#project-status)
> for what is verified vs. in progress.

---

## Table of Contents

- [Quick Start](#quick-start)
- [What It Does](#what-it-does)
- [Architecture](#architecture)
- [Tech Stack](#tech-stack)
- [Project Structure](#project-structure)
- [Documentation](#documentation)
- [Common Tasks](#common-tasks)
- [Configuration](#configuration)
- [Testing](#testing)
- [Project Status](#project-status)
- [Contributing](#contributing)
- [License](#license)

---

## Quick Start

### Prerequisites

| Tool | Version | Why |
|---|---|---|
| [Docker](https://docs.docker.com/get-docker/) + Compose v2 | latest | Runs the full stack |
| [uv](https://docs.astral.sh/uv/) | latest | Python package manager (backend) |
| [pnpm](https://pnpm.io/) | latest | Node package manager (frontend) |
| Node.js | 22+ | Frontend runtime |
| Python | 3.12+ | Backend runtime |

### One-command setup

```bash
git clone <repo-url> && cd GBI
make setup
```

`make setup` checks prerequisites, generates `.env` files with random secrets,
installs dependencies, builds Docker images, starts the stack, runs migrations,
and verifies every service is healthy. On success it prints the access URLs.

Demo credentials

A dev-only user is seeded into a fresh database (via `infra/postgres/init.sql`):

Email	Password	Roles
`admin@genbi.local`	`admin123`	`admin`, `user`

> Dev-only — remove or rotate before any production deployment. Existing
> databases created before this seed need `make reset` (init.sql only runs on
> fresh volumes).

Database roles (enforced tenant isolation)

The backend never connects as a superuser (see
[ADR 006](docs/adr/006-enforced-rls-roles.md)). Roles are created by
`init.sql` on fresh volumes and by Alembic `0002_app_roles` everywhere else:

Role	Password (dev)	Used by	Can do
`genbi` (owner)	`genbi`	Alembic, admin scripts (`DATABASE_URL_SYNC`)	Everything
`genbi_app`	`genbi_app`	Backend runtime + query connector (`DATABASE_URL`)	RLS-bound DML/SELECT; sees only rows matching `app.current_tenant_i
d`
`genbi_auth`	`genbi_auth`	Login endpoint (`DATABASE_URL_AUTH`)	Read `users` across tenants; nothing else

All dev passwords are fixed defaults — **rotate in production**. Analytics
tables (`sales`, `orders`, …, `web_users`) are FORCE-RLS tenant-scoped like
the metadata tables (Alembic `0003_analytics`).

> **Upgrading an existing stack:** `make migrate` creates the roles and
> analytics tables, then regenerate env files so the runtime uses the new
> role URLs: `scripts/gen-env.sh --force && make restart`. Old seed data with
> an analytics `users` table conflicts with `web_users` — the cleanest path
> is `make reset` (fresh volume + init.sql) and re-seed.

<details>
<summary><b>Manual setup (if you prefer not to use the Makefile)</b></summary>

```bash
# 1. Generate env files (creates backend/.env and semantic/cube/.env)
scripts/gen-env.sh

# 2. Install host-side dev deps
cd backend && uv sync --dev && cd ..
cd frontend && pnpm install && cd ..
```
```

```

# 3. Build and start the stack
docker compose -f infra/docker-compose.dev.yml up -d --build

# 4. Apply migrations
docker compose -f infra/docker-compose.dev.yml exec backend uv run alembic upgrade head

# 5. Verify
scripts/verify.sh
```

</details>

Access the platform

Service	URL	Notes
Frontend	http://localhost:3000	Chat UI with SSE streaming
Backend API (Swagger)	http://localhost:8000/docs	Interactive OpenAPI docs
Health	http://localhost:8000/api/v1/health	Liveness + readiness probes
Cube (semantic layer)	http://localhost:4000	Metrics API
Prometheus	http://localhost:9090	Metrics dashboard
Grafana	http://localhost:3001	`admin` / `admin`

You'll need to set `ANTHROPIC_API_KEY` in `backend/.env` to enable the LLM
features (NL2SQL, narrative generation).

What It Does

A single user query flows through a governed multi-agent pipeline:

```
User Query (NL)
  → RouterAgent        classify intent, build dispatch plan
    → NL2SQLAgent      generate SQL with pgvector schema context + Cube metrics
      → ValidationAgent   read-only + destructive-pattern safety gate
        → Connector        execute (read-only, tenant-scoped, 30s timeout)
          → ChartGenAgent   Flint chart spec + render + hallucination check
            → NarrativeAgent   written insight paragraph
              → AuditLog          persist full trace
```

Governance at every step:
- **Read-only enforcement** — `ValidationAgent` rejects any non-SELECT or
  destructive SQL before execution; the connector sets
  `SET TRANSACTION READ ONLY` at the DB level.
- **Multi-tenant isolation** — PostgreSQL row-level security (RLS) with
  `FORCE` scopes every query by `tenant_id` from the JWT.
- **PII masking** — sensitive columns are masked before results reach agents
  or the cache, so no PII ever enters LLM context.
- **Audit trail** — every LLM call logs tokens, latency, model version, and a
  prompt hash to the `audit_log` table.
- **Chart hallucination detection** — a deterministic validator checks chart
  specs against the underlying data and auto-corrects common errors.

Architecture

GenBI is built on two open-source foundations, with a semantic layer,
governance controls, and a Next.js frontend layered on top.

- **[DB-GPT](https://github.com/eosphoros-ai/DB-GPT)** — agentic AI data
  application framework (Text2SQL, sandboxed execution, RAG).
- **[Flint Chart](https://github.com/microsoft/flint-chart)** — MCP-native
  chart rendering (unified spec → Vega-Lite / ECharts / Chart.js).

```
┌──────────────────────────────────────────────┐
│  Interface      Next.js 15 · FastAPI REST · SSE     │
├──────────────────────────────────────────────┤
│  Orchestration  LangGraph agents (NL2SQL, ChartGen, │
│                 Narrative, Router, Validation)      │
├──────────────────────────────────────────────┤
│  Semantic       Cube-native metric catalog → Cube API│
├──────────────────────────────────────────────┤
│  Data           PostgreSQL · pgvector · Apache AGE  │
│                 (connectors: read-only, tenant-scoped)│
├──────────────────────────────────────────────┤
│  Governance     RLS · PII masking · audit log · JWT │
└──────────────────────────────────────────────┘
```

For the full design rationale, see the
[Architecture Overview](docs/architecture-overview.md) and the
[Architecture Decision Records](#architecture-decision-records-adrs).

Tech Stack

Layer	Technology
Backend	Python 3.12 · FastAPI · Uvicorn / Gunicorn · Pydantic v2 · SQLAlchemy 2.0 (async)
LLM	Anthropic Claude (Opus for SQL reasoning, Haiku for speed) · LangChain / LangGraph · LlamaIndex
Semantic layer	Cube.dev (headless BI, native data models)
Databases	PostgreSQL 16 · pgvector (schema embeddings) · Apache AGE (lineage graph) · Redis (cache)
Frontend	Next.js 15 (App Router) · TypeScript 5.9 · Tailwind CSS v4 · shadcn/ui · Zod
Charts	Flint MCP → Vega-Lite / ECharts / Chart.js
Observability	OpenTelemetry · Langfuse (LLM tracing) · Prometheus + Grafana
Auth	JWT + RBAC · PostgreSQL row-level security
CI/CD	GitHub Actions · Docker Compose

Package managers: `uv` for Python (never `pip`), `pnpm` for Node (never
`npm`/`yarn`). See [`CLAUDE.md`](CLAUDE.md) §2 for the full conventions.

Project Structure

```
genbi/
├── backend/               FastAPI app + LangGraph agents
│   ├── app/
│   │   ├── agents/        NL2SQL, ChartGen, Narrative, Router, Validation
│   │   ├── api/v1/        Versioned REST routers (/api/v1/*)
│   │   ├── connectors/    Read-only SQLAlchemy DB connectors
│   │   ├── core/          config, auth, cache, llm_client, masking, logging
│   │   ├── db/            ORM models, sessions, Alembic migrations
│   │   ├── semantic/      Cube.dev client
│   │   └── services/      chat_service (pipeline orchestrator)
│   ├── tests/             unit, integration, golden evals
│   ├── alembic.ini
│   └── Dockerfile
├── frontend/              Next.js 15 + shadcn/ui
│   └── src/{app,components,lib,types}/
├── semantic/              Cube.dev project + metric catalog (schema/)
├── infra/                 Docker Compose (dev + prod), Postgres init, Prometheus
│   └── postgres/Dockerfile   Custom image: pgvector + Apache AGE
├── scripts/               setup.sh, gen-env.sh, verify.sh, embed_schema.py
├── docs/                  Architecture docs + ADRs + OpenAPI
├── Makefile               setup / up / down / verify / migrate / seed / ...
└── CLAUDE.md              AI coding-assistant context (conventions + rules)
```

Documentation

Start here
- **docs/index.md** — curated entry point with quick links

Architecture & Design
- [Architecture Overview](docs/architecture-overview.md) — four-tier design, data flow, tech stack
- [Agent System](docs/agent-system.md) — multi-agent LangGraph pipeline, agent roster, safety gate
- [Core Services](docs/core-services.md) — config, LLM client, cache, auth, PII masking, observability
- [Data Layer](docs/data-layer.md) — ORM models, read-only connectors, AGE graph schema, migrations
- [Semantic Layer](docs/semantic-layer.md) — Cube-native catalog → Cube.dev → Python client
- [Frontend Guide](docs/frontend-guide.md) — Next.js App Router, shadcn/ui, SSE chat, Zod validators

API
- [API Reference](docs/api-reference.md) — REST endpoints, request/response schemas, SSE streaming
- [OpenAPI Spec](docs/api/openapi.yaml) — machine-readable API contract

Operations
- [Infrastructure](docs/infrastructure.md) — Docker Compose, CI/CD, env vars, scripts

Architecture Decision Records (ADRs)
- [ADR 001 — Dual chart stack](docs/adr/001-dual-chart-stack.md) — Flint MCP + AntV G2
- [ADR 002 — Agent architecture](docs/adr/002-agent-architecture.md) — LangGraph multi-agent topology
- [ADR 003 — vis-flint protocol](docs/adr/003-vis-flint-protocol.md) — chart embedding protocol
- [ADR 004 — Flint bridge fallback](docs/adr/004-flint-bridge-fallback.md) — render fallback path
- [ADR 005 — AGE + pgvector image](docs/adr/005-age-and-pgvector-image.md) — custom Postgres image for both extensions
- [ADR 006 — Enforced RLS roles](docs/adr/006-enforced-rls-roles.md) — non-superuser runtime roles, RLS actually enforced
- [ADR 007 — Cube-native semantic layer](docs/adr/007-cube-native-semantic-layer.md) — metric catalog as Cube data models (dbt tier dropped)
- [ADR 008 — Per-tenant Cube data path](docs/adr/008-per-tenant-cube-data-path.md) — JWT tenant claims → per-tenant driver GUC → RLS, fail-closed

AI assistant context
- CLAUDE.md — full project context, hard rules, and conventions for AI coding tools

---

Common Tasks

All tasks are wrapped in the [`Makefile`](Makefile). Run `make help` for the
full list.

Task	Command	Notes
First-time setup	`make setup`	Prereqs → secrets → deps → build → migrate → verify
Start the stack	`make up`	Dev compose, detached
Stop the stack	`make down`	Keeps data volumes
Tail logs	`make logs`	All services, Ctrl-C to exit
Service status	`make ps`	Health + state
Smoke test	`make verify`	Health endpoints + DB + Redis + AGE
Apply migrations	`make migrate`	`alembic upgrade head` in backend
Load test data	`make seed`	Synthetic 10-table dataset
Regenerate secrets	`make secrets`	New random JWT / Cube / Fernet keys
Nuke + restart	`make reset`	`down -v && up` — destroys data

Adding a new metric

1. Define it in `semantic/cube/schema/{domain}.yml` (Cube data-model YAML).
2. Run `uv run pytest backend/tests/semantic/test_catalog.py` — structural invariants.
3. Restart Cube (`make restart`) — `/meta` reflects the change (5-min client cache).
4. Re-embed the schema: `uv run python scripts/embed_schema.py`.

See the [Semantic Layer docs](docs/semantic-layer.md) for details.

Adding a new agent

1. Create `backend/app/agents/{name}_agent.py` extending `BaseAgent`.
2. Register in `backend/app/agents/registry.py`.
3. Add tests in `backend/tests/agents/` with mocked LLM (`tests/fixtures/llm_mock.py`).
4. Update `RouterAgent`'s dispatch table.

See the [Agent System docs](docs/agent-system.md) for the full protocol.

Configuration

All configuration is via environment variables, loaded by
[`backend/app/core/config.py`](backend/app/core/config.py) with Pydantic
Settings. Never commit `.env` files.

Variable	Required	Description
`ANTHROPIC_API_KEY`	yes	Claude API access
`DATABASE_URL`	yes	`postgresql+asyncpg://...`
`REDIS_URL`	yes	`redis://host:6379/0`
`JWT_SECRET_KEY`	yes (prod)	Random 32+ hex; **boot fails if default in non-dev**
`CUBE_API_URL` / `CUBE_API_SECRET`	yes	Cube.dev semantic layer
`TENANT_ENCRYPTION_KEY`	yes (prod)	Fernet key for tenant secrets
`LANGFUSE_*`	optional	LLM tracing (disabled if unset)
`OTEL_EXPORTER_OTLP_ENDPOINT`	optional	OTel collector

`scripts/gen-env.sh` (via `make setup` / `make secrets`) generates
`backend/.env` and `semantic/cube/.env` from `.env.example` with random secrets.
See `.env.example` for the full list with defaults.

Testing

```bash
# Backend unit + integration tests (mocked LLM, no live services)
cd backend && uv run pytest tests/ -v -m "not e2e"

# Golden NL2SQL eval gate (20 queries, 90% accuracy threshold)
cd backend && uv run pytest tests/evals/ -v

# Lint + format check
cd backend && uv run ruff check . && uv run ruff format --check .

# Frontend
cd frontend && pnpm lint && pnpm typecheck
```

The golden NL2SQL suite is a **merge gate**: 20 NL/SQL pairs scored across
three tolerance levels (exact / fuzzy / semantic), requiring ≥90% suite
accuracy. The LLM transport is mocked, but the agent's prompt-building, JSON
parsing, and destructive-pattern detection all run for real. See
[`backend/tests/evals/`](backend/tests/evals) and
[CLAUDE.md §9](CLAUDE.md) for the testing philosophy.

Project Status

Area	Status	Notes
Agent pipeline (NL→SQL→Chart→Narrative)	Implemented	Sync + SSE streaming
Validation safety gate	Hardened	EXPLAIN-backed row estimates + >1M-row confirmation flow (Phase 13); deterministic pattern gate remains
Semantic layer (Cube-native catalog)	Implemented	10 cubes / 23 measures; tenant-scoped /metrics/query + Explore page (ADR 007/008)
Governance: audit trail + login throttling	Implemented	Every LLM call audited (tenant-scoped, fail-open); 5 failed logins → 429; feedback API + per-role query policy (Phases 12/15)
Reports (multi-chart from NL)	Implemented	LLM-planned + deterministic execution, persisted + browsable UI (Phase 16)
NL2SQL schema grounding (pgvector retrieval)	Implemented	Schema context + few-shot into every prompt (Phase 11); needs `embed_schema.py --examples` + OPENAI_API_KEY
Multi-tenant RLS + JWT	Enforced	Non-superuser runtime roles + FORCE RLS on all tenant tables, incl. analytics (ADR 006)
PII masking	Wired	Applied at the execution chokepoint
Chart hallucination detection	Implemented	6 validation categories + auto-correct
Dual-tier cache (L1 LRU + L2 Redis)	Implemented	Schema, metrics, results, charts
Observability (OTel + Langfuse + Prometheus)	Wired	`/metrics` endpoint on backend
Frontend chat UI	Implemented	SSE streaming, stage rendering, conversation sidebar + multi-turn history (Phase 14)
Environment provisioning (`make setup`)	Implemented	One-command bootstrap
End-to-end verification	Verified	120 tests pass, `make verify` green, CI builds + eval gate (Phases 7–12)

> Phase 7 (build & test verification) closed out Phases 5b/6 as VERIFIED.
> Phases 8–12 delivered the auth + chat vertical slice, enforced tenant
> isolation (non-superuser roles, ADR 006), the Cube-native metric catalog
> (ADR 007), tenant-scoped Metrics + Explore (ADR 008), NL2SQL schema
> grounding, and governance (audit trail + login throttling).

> See [`todo.md`](todo.md) for the phase tracker (Phases 1–12) and the
> implementation/verification history.

Contributing

1. Read CLAUDE.md first — it codifies the hard rules (security,
   LLM patterns, coding conventions) that every change must respect.
2. Use the package managers (`uv`, `pnpm`) — never `pip` / `npm`.
3. All API routes are versioned under `/api/v1/`.
4. Mock the LLM in tests; never call real APIs in unit/integration tests.
5. Run `make verify` before requesting review.

Protected files (require security review to change)

Per CLAUDE.md §13, these are gated:
- `backend/app/core/auth.py` — JWT + RBAC
- `backend/app/agents/validation/validation_agent.py` — SQL safety gate
- `backend/app/db/migrations/` — Alembic history (never edit existing files)
- `infra/postgres/init.sql` — DB initialization

License

[MIT](LICENSE) © GenBI contributors.

scripts/db_admin.py

38 lines · 1.4 KB

```python
"""Shared admin-DB helpers for scripts that run as the owner role.

seed_test_data.py and embed_schema.py do DDL/DML that the RLS-bound runtime
role (genbi_app) must never perform, and FORCE ROW LEVEL SECURITY binds even
the table owner — so admin scripts connect as the owner (DATABASE_URL_SYNC)
AND set the tenant GUC before touching tenant-scoped rows.
"""

import asyncpg
from sqlalchemy.engine import make_url

from app.core.config import settings


def owner_dsn(connection_url: str | None = None) -> str:
    """Owner DSN for asyncpg (postgresql://user:pass@host:port/db).

    Defaults to settings.DATABASE_URL_SYNC (owner credentials).
    """
    url = make_url(connection_url or settings.DATABASE_URL_SYNC)
    if url.drivername != "postgresql":
        url = url.set(drivername="postgresql")
    return url.render_as_string(hide_password=False)


async def owner_connect(connection_url: str | None = None) -> asyncpg.Connection:
    """Connection as the owner role (DDL + admin DML)."""
    return await asyncpg.connect(owner_dsn(connection_url))


async def set_tenant_guc(conn: asyncpg.Connection, tenant_id: str) -> None:
    """Set the session-level tenant GUC.

    Required for INSERT/SELECT on RLS-FORCED tables even as the owner. Session
    scope (is_local=false) is intentional: admin scripts drive one connection
    per tenant pass.
    """
    await conn.execute("SELECT set_config('app.current_tenant_id', $1, false)", tenant_id)
```

scripts/embed_schema.py

352 lines · 12.1 KB

```python
"""Schema embedding pipeline — nightly sync of database schema to pgvector.

Generates embeddings for all table/column metadata and stores them in the
schema_embeddings table for semantic search by NL2SQLAgent.

Connects as the OWNER role (scripts/db_admin.owner_connect): schema_embeddings
is RLS-enrolled, so the tenant GUC is set on the connection before any write
(FORCE RLS binds the owner too).

Usage (from the repo root, backend venv active):
    PYTHONPATH=backend uv run python scripts/embed_schema.py                # full sync
    PYTHONPATH=backend uv run python scripts/embed_schema.py --domain sales  # single domain
    PYTHONPATH=backend uv run python scripts/embed_schema.py --dry-run       # preview without writing
"""

import argparse
import asyncio
import json
import sys
from datetime import datetime, timezone
from pathlib import Path
from uuid import uuid4

from app.core.logging import logger
from db_admin import owner_connect, set_tenant_guc

DEFAULT_TENANT_ID = "00000000-0000-0000-0000-000000000001"


async def get_schema_metadata(conn, schema: str = "public") -> list[dict]:
    """Retrieve all table and column metadata from information_schema.

    Returns a list of dicts with table/column metadata grouped per table.
    """
    rows = await conn.fetch(
        """
        SELECT
            t.table_schema,
            t.table_name,
            c.column_name,
            c.data_type,
            c.udt_name,
            c.is_nullable,
            c.ordinal_position,
            pgd.description AS column_description,
            obj_description(pgc.oid) AS table_description
        FROM information_schema.tables t
        JOIN information_schema.columns c
            ON t.table_schema = c.table_schema
            AND t.table_name = c.table_name
        LEFT JOIN pg_catalog.pg_class pgc
            ON pgc.relname = t.table_name
            AND pgc.relnamespace = (
                SELECT oid FROM pg_catalog.pg_namespace WHERE nspname = t.table_schema
            )
        LEFT JOIN pg_catalog.pg_description pgd
            ON pgd.objoid = pgc.oid
            AND pgd.objsubid = c.ordinal_position
        WHERE t.table_schema = $1
            AND t.table_type = 'BASE TABLE'
        ORDER BY t.table_name, c.ordinal_position
        """,
        schema,
    )

    # Group columns by table
    tables: dict[str, dict] = {}
    for row in rows:
        record = dict(row)
        key = f"{record['table_schema']}.{record['table_name']}"
        if key not in tables:
            tables[key] = {
                "table_schema": record["table_schema"],
                "table_name": record["table_name"],
                "full_name": key,
                "table_description": record["table_description"] or "",
                "columns": [],
            }

        tables[key]["columns"].append({
            "column_name": record["column_name"],
            "data_type": record["data_type"],
            "udt_name": record["udt_name"],
            "is_nullable": record["is_nullable"],
            "ordinal_position": record["ordinal_position"],
            "description": record["column_description"] or "",
        })

    return list(tables.values())


def build_embedding_text(table: dict) -> str:
    """Build a text representation of a table for embedding.

    The embedding text is what gets vectorized — it should be rich with
    semantic meaning about the table and its columns.
    """
    parts = [f"Table: {table['full_name']}"]

    if table.get("table_description"):
```

```
        parts.append(f"Description: {table['table_description']}")

    parts.append("Columns:")
    for col in table["columns"]:
        col_text = f"  - {col['column_name']} ({col['data_type']}"
        if col["udt_name"] != col["data_type"]:
            col_text += f"/{col['udt_name']}"
        col_text += ")"
        if col.get("description"):
            col_text += f": {col['description']}"
        parts.append(col_text)

    return "\n".join(parts)


async def generate_embedding(text: str) -> list[float]:
    """Generate an embedding vector for schema metadata text.

    Delegates to the shared provider (app.core.embeddings — OpenAI
    text-embedding-3-small; the original Anthropic embeddings call targeted
    a model that does not exist).
    """
    from app.core.embeddings import embed_text

    return await embed_text(text)


async def sync_schema(
    conn,
    schema: str = "public",
    dry_run: bool = False,
) -> dict[str, int]:
    """Sync schema metadata to the schema_embeddings table.

    For each table in information_schema:
        1. Build embedding text from table + column metadata
        2. Generate embedding via Anthropic API
        3. Upsert into schema_embeddings with tenant_id

    Returns:
        Dict with counts: tables_processed, embeddings_generated, errors.
    """
    tables = await get_schema_metadata(conn, schema)

    if not tables:
        logger.warning(f"No tables found in schema '{schema}'")
        return {"tables_processed": 0, "embeddings_generated": 0, "errors": 0}

    logger.info(f"Found {len(tables)} tables in schema '{schema}'")

    # schema_embeddings is RLS-enrolled and FORCEd — even the owner needs the
    # tenant GUC set before writing.
    await set_tenant_guc(conn, DEFAULT_TENANT_ID)

    processed = 0
    embeddings = 0
    errors = 0

    for table in tables:
        processed += 1
        embedding_text = build_embedding_text(table)
        columns_json = json.dumps(table["columns"], default=str)

        try:
            if not dry_run:
                # Generate embedding
                embedding_vector = await generate_embedding(embedding_text)
                embeddings += 1

                # Upsert (single transaction per table; embedding API call
                # must not hold a transaction open).
                async with conn.transaction():
                    await conn.execute(
                        """
                        INSERT INTO schema_embeddings (
                            id, tenant_id, table_schema, table_name,
                            full_name, table_description, columns_json,
                            embedding_text, embedding, created_at, updated_at
                        ) VALUES (
                            $1, $2, $3, $4,
                            $5, $6, $7,
                            $8, $9::vector(1536), $10, $10
                        )
                        ON CONFLICT (tenant_id, table_schema, table_name)
                        DO UPDATE SET
                            columns_json = EXCLUDED.columns_json,
                            table_description = EXCLUDED.table_description,
                            embedding_text = EXCLUDED.embedding_text,
                            embedding = EXCLUDED.embedding,
                            updated_at = EXCLUDED.updated_at
                        """,
                        str(uuid4()),
                        DEFAULT_TENANT_ID,
                        table["table_schema"],
                        table["table_name"],
                        table["full_name"],
                        table.get("table_description", ""),
                        columns_json,
                        embedding_text,
                        str(embedding_vector),
                        datetime.now(timezone.utc),
                    )

                logger.info(
                    f"Embedded table {table['full_name']} "
```

```
                    f"({len(table['columns'])} columns, "
                    f"{len(embedding_vector)}-dim vector)"
                )

        except Exception as e:
            errors += 1
            logger.error(f"Failed to embed table {table['full_name']}: {e}")

    return {
        "tables_processed": processed,
        "embeddings_generated": embeddings,
        "errors": errors,
    }


async def sync_examples(conn, examples_file: str, tenant_id: str = DEFAULT_TENANT_ID) -> int:
    """(Re)seed agent_examples with the golden NL/SQL pairs for few-shot retrieval.

    Idempotent: deletes existing nl2sql examples for the tenant, then inserts
    the golden set with query embeddings. Runs as the owner with the tenant
    GUC set (agent_examples is RLS-enforced).
    """
    from app.core.embeddings import vector_literal

    path = Path(examples_file)
    if not path.is_file():
        logger.error("Examples file not found: %s", path)
        return 0

    cases = json.loads(path.read_text(encoding="utf-8"))
    if not cases:
        logger.warning("Examples file is empty: %s", path)
        return 0

    await set_tenant_guc(conn, tenant_id)
    async with conn.transaction():
        await conn.execute(
            "DELETE FROM agent_examples WHERE agent_name = $1 AND tenant_id = $2::uuid",
            "nl2sql",
            tenant_id,
        )
        for case in cases:
            nl_query = case.get("nl_query", "")
            expected_sql = case.get("expected_sql", "")
            if not nl_query or not expected_sql:
                continue
            embedding = await generate_embedding(nl_query)
            await conn.execute(
                "INSERT INTO agent_examples "
                "(agent_name, nl_query, expected_sql, tags, embedding, tenant_id) "
                "VALUES ($1, $2, $3, $4::jsonb, $5::vector(1536), $6::uuid)",
                "nl2sql",
                nl_query,
                expected_sql,
                json.dumps([case.get("category", "golden")]),
                vector_literal(embedding),
                tenant_id,
            )

    logger.info("Few-shot examples seeded", count=len(cases), tenant=tenant_id)
    return len(cases)


async def _invalidate_caches(tenant_id: str) -> None:
    """Best-effort: drop cached schema context so retrieval sees fresh rows."""
    try:
        from app.core.cache import get_cache

        await get_cache().invalidate_schema(tenant_id)
        logger.info("Schema context cache invalidated", tenant=tenant_id)
    except Exception as e:
        logger.warning("Cache invalidation skipped (non-fatal): %s", e)


async def main():
    parser = argparse.ArgumentParser(
        description="Sync database schema metadata to pgvector embeddings"
    )
    parser.add_argument(
        "--schema", default="public",
        help="Database schema to embed (default: public)",
    )
    parser.add_argument(
        "--domain",
        help="Limit to tables matching this prefix (e.g. 'sales')",
    )
    parser.add_argument(
        "--dry-run", action="store_true",
        help="Preview tables without generating embeddings or writing",
    )
    parser.add_argument(
        "--examples", action="store_true",
        help="Also seed agent_examples with the golden NL/SQL pairs",
    )
    parser.add_argument(
        "--examples-file",
        default=None,
        help="Golden JSON path for --examples "
        "(default: backend/tests/evals/nl2sql_golden.json relative to repo root)",
    )
    parser.add_argument(
        "--connection-url",
        default=None,
        help="Owner PostgreSQL connection URL (default: DATABASE_URL_SYNC)",
    )
```

```
    args = parser.parse_args()

    conn = await owner_connect(args.connection_url)
    try:
        logger.info(
            "Starting schema embedding sync",
            schema=args.schema,
            domain=args.domain,
            dry_run=args.dry_run,
        )

        result = await sync_schema(
            conn=conn,
            schema=args.schema,
            dry_run=args.dry_run,
        )

        if args.examples and not args.dry_run:
            examples_file = args.examples_file
            if examples_file is None:
                repo_root = Path(__file__).resolve().parents[1]
                examples_file = str(repo_root / "backend" / "tests" / "evals" / "nl2sql_golden.json")
            await sync_examples(conn, examples_file)
    finally:
        await conn.close()

    logger.info(
        "Schema embedding sync complete",
        tables_processed=result["tables_processed"],
        embeddings_generated=result["embeddings_generated"],
        errors=result["errors"],
    )

    if result["errors"] > 0:
        sys.exit(1)

    if not args.dry_run:
        await _invalidate_caches(DEFAULT_TENANT_ID)


if __name__ == "__main__":
    asyncio.run(main())
```

scripts/gen-env.sh

127 lines · 4.8 KB

```
#!/usr/bin/env bash
# Generate backend/.env and semantic/cube/.env from templates with random
# secrets. Idempotent: refuses to overwrite existing files unless --force.
#
# Usage:
#   scripts/gen-env.sh           # write missing files only
#   scripts/gen-env.sh --force   # overwrite existing files
#
# Designed to be safe to re-run (make secrets). The ANTHROPIC_API_KEY is left
# as a placeholder — it requires a real key and is the one secret we cannot
# generate. A warning is printed if it is still a placeholder.

set -euo pipefail

REPO_ROOT="$(cd "$(dirname "${BASH_SOURCE[0]}")/.." && pwd)"
FORCE=0
for arg in "$@"; do
  case "$arg" in
    --force|-f) FORCE=1 ;;
    *) echo "Unknown arg: $arg" >&2; exit 2 ;;
  esac
done

# --- helpers ----------------------------------------------------------------

rand_hex() { openssl rand -hex "$1"; }

gen_fernet() {
  python3 -c "from cryptography.fernet import Fernet; print(Fernet.generate_key().decode())" 2>/dev/null \
    || python3 -c "import base64,os; print(base64.urlsafe_b64encode(os.urandom(32)).decode())"
}

# Emit a line only if the key isn't already going to be replaced. We rewrite
# the whole file from the template each time.

# Returns 0 (true) when we SHOULD write (file missing, or --force).
should_write() {
  local target="$1"
  if [[ "$FORCE" -eq 1 ]]; then
    return 0
  fi
  if [[ -f "$target" ]]; then
    echo "  exists: $target (use --force to overwrite)"
    return 1
  fi
  return 0
}

# --- backend/.env -----------------------------------------------------------

BACKEND_ENV="$REPO_ROOT/backend/.env"
TEMPLATE="$REPO_ROOT/.env.example"

if should_write "$BACKEND_ENV"; then
  if [[ ! -f "$TEMPLATE" ]]; then
    echo "ERROR: template $TEMPLATE not found" >&2; exit 1
  fi

  JWT_SECRET="$(rand_hex 32)"
  CUBE_SECRET="$(rand_hex 32)"
  TENANT_KEY="$(gen_fernet)"
  # Owner credentials (migrations/admin only). The runtime DATABASE_URL uses
  # the RLS-bound genbi_app role, and the login path genbi_auth — fixed dev
  # defaults matching infra/postgres/init.sql + Alembic 0002_app_roles.
  # Rotate in production.
  PG_USER="genbi"
  PG_PASS="genbi"
  PG_DB="genbi"

  # Substitute placeholders. DATABASE_URL/REDIS_URL use the compose service
  # names ('postgres', 'redis') as hosts — the .env file is consumed by the
  # backend container, where 'localhost' would mean the container itself.
  # Host-based (non-Docker) dev should override these via a local .env override.
  sed \
    -e "s|^ANTHROPIC_API_KEY=.*|ANTHROPIC_API_KEY=sk-ant-REPLACE-ME|" \
    -e "s|^OPENAI_API_KEY=.*|OPENAI_API_KEY=sk-REPLACE-ME|" \
    -e "s|^DATABASE_URL=.*|DATABASE_URL=postgresql+asyncpg://genbi_app:genbi_app@postgres:5432/${PG_DB}|" \
    -e "s|^DATABASE_URL_AUTH=.*|DATABASE_URL_AUTH=postgresql+asyncpg://genbi_auth:genbi_auth@postgres:5432/${PG_DB}|" \
    -e "s|^DATABASE_URL_SYNC=.*|DATABASE_URL_SYNC=postgresql://${PG_USER}:${PG_PASS}@postgres:5432/${PG_DB}|" \
    -e "s|^REDIS_URL=.*|REDIS_URL=redis://redis:6379/0|" \
    -e "s|^CUBE_API_URL=.*|CUBE_API_URL=http://cube:4000/cubejs-api/v1|" \
    -e "s|^JWT_SECRET_KEY=.*|JWT_SECRET_KEY=${JWT_SECRET}|" \
    -e "s|^CUBE_API_SECRET=.*|CUBE_API_SECRET=${CUBE_SECRET}|" \
    -e "s|^TENANT_ENCRYPTION_KEY=.*|TENANT_ENCRYPTION_KEY=${TENANT_KEY}|" \
    "$TEMPLATE" > "$BACKEND_ENV"

  echo "  wrote:  $BACKEND_ENV"
  echo "          (JWT_SECRET_KEY, CUBE_API_SECRET, TENANT_ENCRYPTION_KEY randomized)"
  if grep -q "REPLACE-ME" "$BACKEND_ENV"; then
    echo "      ANTHROPIC_API_KEY / OPENAI_API_KEY are placeholders — set them before running LLM/embedding features."
  fi
fi

# --- semantic/cube/.env -----------------------------------------------------

CUBE_ENV="$REPO_ROOT/semantic/cube/.env"
CUBE_TEMPLATE="$REPO_ROOT/semantic/cube/.env.example"

if should_write "$CUBE_ENV"; then
  if [[ ! -f "$CUBE_TEMPLATE" ]]; then
```

```
    echo "ERROR: template $CUBE_TEMPLATE not found" >&2; exit 1
  fi

  # Reuse the CUBE_API_SECRET generated above so backend and Cube agree.
  # If backend/.env already existed (not rewritten this run), read it from there.
  if [[ -z "${CUBE_SECRET:-}" ]]; then
    CUBE_SECRET="$(grep '^CUBE_API_SECRET=' "$BACKEND_ENV" 2>/dev/null | cut -d= -f2- || true)"
  fi
  if [[ -z "${CUBE_SECRET}" ]]; then
    CUBE_SECRET="$(rand_hex 32)"
  fi

  PG_PASS="genbi"
  # cube_reader: read-only role created by infra/postgres/init.sql. Dev
  # default password — rotate in production.
  sed \
    -e "s|^CUBEJS_DB_HOST=.*|CUBEJS_DB_HOST=postgres|" \
    -e "s|^CUBEJS_DB_PASS=.*|CUBEJS_DB_PASS=changeme_cube_reader|" \
    -e "s|^CUBEJS_API_SECRET=.*|CUBEJS_API_SECRET=${CUBE_SECRET}|" \
    "$CUBE_TEMPLATE" > "$CUBE_ENV"

  echo "  wrote:  $CUBE_ENV"
  echo "           (CUBEJS_DB_* → postgres host + cube_reader role, CUBEJS_API_SECRET aligned with backend)"
fi

echo ""
echo "  environment files ready"
```

scripts/seed_test_data.py

452 lines · 16.3 KB

```python
"""Test data infrastructure — synthetic analytics data for development.

Tables (created by Alembic migration 0003_analytics, all FORCE-RLS tenant-
scoped): sales, customers, orders, transactions, web_users, products,
regions, sales_representatives, deals, activity. This script only INSERTs
data, connecting as the OWNER role (scripts/db_admin.owner_connect) and
setting the tenant GUC per tenant pass (FORCE RLS binds the owner too).

Usage (from the repo root, backend venv active; in-container via `make seed`):
    PYTHONPATH=backend uv run python scripts/seed_test_data.py              # seed all tables
    PYTHONPATH=backend uv run python scripts/seed_test_data.py --tenants 5   # multi-tenant
"""

import argparse
import asyncio
import random
import uuid
from datetime import datetime, timedelta

from app.core.logging import logger
from db_admin import owner_connect, set_tenant_guc


# ---------------------------------------------------------------------------
# Synthetic data generators
# ---------------------------------------------------------------------------

REGIONS = ["North", "South", "East", "West", "Central"]
COUNTRIES = [
    "United States", "Germany", "France", "Japan", "United Kingdom",
    "Canada", "Australia", "Brazil", "India", "Singapore",
]
PRODUCT_CATEGORIES = [
    "Software", "Hardware", "Services", "Support", "Training",
]
PRODUCT_NAMES = [
    "GenBI Enterprise", "GenBI Pro", "GenBI Starter",
    "Data Connector Pack", "Advanced Analytics Add-on",
    "AML Screening Module", "Risk Dashboard Suite",
]
CUSTOMER_NAMES = [
    "Acme Corp", "Globex Inc", "Initech", "Umbrella Co",
    "Stark Industries", "Wayne Enterprises", "Cyberdyne Systems",
    "Weyland-Yutani", "Oscorp", "Aperture Science",
    "Hooli", "Pied Piper", "Massive Dynamic", "Soylent Corp",
    "Dunder Mifflin", "Sterling Cooper", "Los Pollos Hermanos",
    "Oceanic Airlines", "Monarch Sciences", "Buy n Large",
]
SALES_REPS = [
    "Alice Johnson", "Bob Smith", "Carol Williams", "Dave Brown",
    "Eve Davis", "Frank Miller", "Grace Wilson", "Henry Moore",
    "Iris Taylor", "Jack Anderson",
]


def random_date(start: datetime, end: datetime) -> datetime:
    """Generate a random date between start and end."""
    delta = end - start
    random_days = random.randint(0, delta.days)
    return start + timedelta(days=random_days)


def generate_sales(tenant_id: str, num_rows: int = 500) -> list[dict]:
    """Generate synthetic sales transactions."""
    rows = []
    start = datetime(2024, 1, 1)
    end = datetime(2026, 6, 30)

    for _ in range(num_rows):
        rows.append({
            "id": str(uuid.uuid4()),
            "tenant_id": tenant_id,
            "region": random.choice(REGIONS),
            "product_id": str(uuid.uuid4()),
            "product_name": random.choice(PRODUCT_NAMES),
            "revenue": round(random.uniform(100, 500000), 2),
            "units": random.randint(1, 100),
            "transaction_date": random_date(start, end),
            "rep_id": str(uuid.uuid4()),
        })

    return rows


def generate_customers(tenant_id: str, num_rows: int = 100) -> list[dict]:
    """Generate synthetic customer records."""
    rows = []
    for _ in range(num_rows):
        name = random.choice(CUSTOMER_NAMES)
        rows.append({
            "id": str(uuid.uuid4()),
            "tenant_id": tenant_id,
            "name": name,
            "email": f"contact@{name.lower().replace(' ', '-')}.com",
            "country": random.choice(COUNTRIES),
            "signup_date": random_date(datetime(2022, 1, 1), datetime(2026, 6, 1)),
            "status": random.choice(["active", "active", "active", "inactive", "churned"]),
        })

    return rows
```

```python


def generate_orders(tenant_id: str, customers: list[dict], num_rows: int = 1000) -> list[dict]:
    """Generate synthetic orders linked to customers."""
    rows = []
    start = datetime(2024, 1, 1)
    end = datetime(2026, 6, 30)

    for _ in range(num_rows):
        customer = random.choice(customers)
        rows.append({
            "id": str(uuid.uuid4()),
            "tenant_id": tenant_id,
            "customer_id": customer["id"],
            "product_id": str(uuid.uuid4()),
            "amount": round(random.uniform(50, 100000), 2),
            "order_date": random_date(start, end),
            "status": random.choice(["completed", "completed", "completed", "pending", "cancelled"]),
        })

    return rows


def generate_transactions(tenant_id: str, num_rows: int = 2000) -> list[dict]:
    """Generate synthetic financial transactions."""
    rows = []
    start = datetime(2024, 1, 1)
    end = datetime(2026, 6, 30)

    for _ in range(num_rows):
        rows.append({
            "id": str(uuid.uuid4()),
            "tenant_id": tenant_id,
            "amount": round(random.uniform(-10000, 100000), 2),
            "transaction_date": random_date(start, end),
            "type": random.choice(["deposit", "withdrawal", "transfer", "payment"]),
            "status": random.choice(["completed", "completed", "pending", "failed"]),
        })

    return rows


def generate_users(tenant_id: str, num_rows: int = 200) -> list[dict]:
    """Generate synthetic web-user records (analytics data, NOT app logins)."""
    rows = []
    start = datetime(2023, 1, 1)
    end = datetime(2026, 6, 30)

    for _ in range(num_rows):
        signup = random_date(start, end)
        last_login = random_date(signup, min(signup + timedelta(days=180), end))
        rows.append({
            "id": str(uuid.uuid4()),
            "tenant_id": tenant_id,
            "name": f"User {random.randint(1, 99999)}",
            "email": f"user{random.randint(1, 99999)}@example.com",
            "country": random.choice(COUNTRIES),
            "signup_date": signup,
            "last_login": last_login,
            "status": random.choice(["active", "active", "inactive"]),
        })

    return rows


def generate_products(tenant_id: str) -> list[dict]:
    """Generate product catalog."""
    rows = []
    for name in PRODUCT_NAMES:
        rows.append({
            "id": str(uuid.uuid4()),
            "tenant_id": tenant_id,
            "product_name": name,
            "category": random.choice(PRODUCT_CATEGORIES),
            "price": round(random.uniform(99, 49999), 2),
        })
    return rows


def generate_regions(tenant_id: str) -> list[dict]:
    """Generate region lookup table."""
    return [
        {"id": str(uuid.uuid4()), "tenant_id": tenant_id, "region_name": r}
        for r in REGIONS
    ]


def generate_sales_reps(tenant_id: str, regions: list[dict]) -> list[dict]:
    """Generate sales representative records."""
    rows = []
    for name in SALES_REPS:
        region = random.choice(regions)
        rows.append({
            "id": str(uuid.uuid4()),
            "tenant_id": tenant_id,
            "name": name,
            "region_id": region["id"],
            "email": f"{name.lower().replace(' ', '.')}@company.com",
            "hire_date": random_date(datetime(2020, 1, 1), datetime(2026, 1, 1)),
        })
    return rows


def generate_deals(tenant_id: str, reps: list[dict], regions: list[dict], num_rows: int = 300) -> list[dict]:
    """Generate synthetic deals linked to reps and regions."""
```

```
    rows = []
    start = datetime(2024, 1, 1)
    end = datetime(2026, 6, 30)

    for _ in range(num_rows):
        rep = random.choice(reps)
        region = random.choice(regions)
        rows.append({
            "id": str(uuid.uuid4()),
            "tenant_id": tenant_id,
            "amount": round(random.uniform(1000, 1000000), 2),
            "rep_id": rep["id"],
            "region_id": region["id"],
            "close_date": random_date(start, end),
            "stage": random.choice(["prospecting", "negotiation", "closed_won", "closed_lost"]),
        })

    return rows


def generate_activity(tenant_id: str, users: list[dict], num_rows: int = 5000) -> list[dict]:
    """Generate synthetic user activity log."""
    rows = []
    start = datetime(2025, 1, 1)
    end = datetime(2026, 6, 30)

    for _ in range(num_rows):
        user = random.choice(users)
        rows.append({
            "id": str(uuid.uuid4()),
            "tenant_id": tenant_id,
            "user_id": user["id"],
            "activity_date": random_date(start, end),
            "event_type": random.choice(["login", "query", "export", "dashboard_view", "report_create"]),
        })

    return rows


# ---------------------------------------------------------------------------
# Seed runner
# ---------------------------------------------------------------------------
# Column-order constants mirror the generator dicts 1:1; inserts go through
# asyncpg executemany with $n positional parameters.

REGION_COLS = ("id", "tenant_id", "region_name")
PRODUCT_COLS = ("id", "tenant_id", "product_name", "category", "price")
CUSTOMER_COLS = ("id", "tenant_id", "name", "email", "country", "signup_date", "status")
WEB_USER_COLS = (
    "id", "tenant_id", "name", "email", "country", "signup_date", "last_login", "status",
)
REP_COLS = ("id", "tenant_id", "name", "region_id", "email", "hire_date")
DEAL_COLS = ("id", "tenant_id", "amount", "rep_id", "region_id", "close_date", "stage")
SALES_COLS = (
    "id", "tenant_id", "region", "product_id", "product_name", "revenue",
    "units", "transaction_date", "rep_id",
)
ORDER_COLS = ("id", "tenant_id", "customer_id", "product_id", "amount", "order_date", "status")
TRANSACTION_COLS = ("id", "tenant_id", "amount", "transaction_date", "type", "status")
ACTIVITY_COLS = ("id", "tenant_id", "user_id", "activity_date", "event_type")


def _tuples(rows: list[dict], cols: tuple[str, ...]) -> list[tuple]:
    """Project generated dicts to value tuples in column order."""
    return [tuple(row[c] for c in cols) for row in rows]


async def _clear_data(conn) -> None:
    """Truncate all analytics tables (RLS does not filter TRUNCATE)."""
    await conn.execute("TRUNCATE TABLE public.activity")
    await conn.execute("TRUNCATE TABLE public.transactions")
    await conn.execute("TRUNCATE TABLE public.orders")
    await conn.execute("TRUNCATE TABLE public.sales")
    await conn.execute("TRUNCATE TABLE public.deals")
    await conn.execute("TRUNCATE TABLE public.sales_representatives")
    await conn.execute("TRUNCATE TABLE public.web_users")
    await conn.execute("TRUNCATE TABLE public.customers")
    await conn.execute("TRUNCATE TABLE public.products")
    await conn.execute("TRUNCATE TABLE public.regions")
    logger.info("Cleared all seed data")


async def _seed_tenant(conn, t_id: str) -> int:
    """Generate and insert one tenant's data. Returns rows inserted."""
    regions = generate_regions(t_id)
    products = generate_products(t_id)
    customers = generate_customers(t_id)
    users = generate_users(t_id)
    reps = generate_sales_reps(t_id, regions)
    deals = generate_deals(t_id, reps, regions)
    sales = generate_sales(t_id)
    orders = generate_orders(t_id, customers)
    transactions = generate_transactions(t_id)
    activity = generate_activity(t_id, users)

    inserted = 0

    # FORCE RLS binds the owner too — the tenant GUC must be set on this
    # connection before any tenant-scoped DML.
    await set_tenant_guc(conn, t_id)

    async with conn.transaction():
        await conn.executemany(
            "INSERT INTO public.regions (id, tenant_id, region_name) "
            "VALUES ($1, $2, $3) ON CONFLICT (id) DO NOTHING",
```

```
            _tuples(regions, REGION_COLS),
        )
        inserted += len(regions)
        await conn.executemany(
            "INSERT INTO public.products (id, tenant_id, product_name, category, price) "
            "VALUES ($1, $2, $3, $4, $5) ON CONFLICT (id) DO NOTHING",
            _tuples(products, PRODUCT_COLS),
        )
        inserted += len(products)
        await conn.executemany(
            "INSERT INTO public.customers "
            "(id, tenant_id, name, email, country, signup_date, status) "
            "VALUES ($1, $2, $3, $4, $5, $6, $7) ON CONFLICT (id) DO NOTHING",
            _tuples(customers, CUSTOMER_COLS),
        )
        inserted += len(customers)
        await conn.executemany(
            "INSERT INTO public.web_users "
            "(id, tenant_id, name, email, country, signup_date, last_login, status) "
            "VALUES ($1, $2, $3, $4, $5, $6, $7, $8) ON CONFLICT (id) DO NOTHING",
            _tuples(users, WEB_USER_COLS),
        )
        inserted += len(users)
        await conn.executemany(
            "INSERT INTO public.sales_representatives "
            "(id, tenant_id, name, region_id, email, hire_date) "
            "VALUES ($1, $2, $3, $4, $5, $6) ON CONFLICT (id) DO NOTHING",
            _tuples(reps, REP_COLS),
        )
        inserted += len(reps)
        await conn.executemany(
            "INSERT INTO public.deals "
            "(id, tenant_id, amount, rep_id, region_id, close_date, stage) "
            "VALUES ($1, $2, $3, $4, $5, $6, $7) ON CONFLICT (id) DO NOTHING",
            _tuples(deals, DEAL_COLS),
        )
        inserted += len(deals)

        for i in range(0, len(sales), 500):
            chunk = _tuples(sales[i:i + 500], SALES_COLS)
            await conn.executemany(
                "INSERT INTO public.sales "
                "(id, tenant_id, region, product_id, product_name, revenue, units, "
                "transaction_date, rep_id) "
                "VALUES ($1, $2, $3, $4, $5, $6, $7, $8, $9) ON CONFLICT (id) DO NOTHING",
                chunk,
            )
            inserted += len(chunk)
        for i in range(0, len(orders), 500):
            chunk = _tuples(orders[i:i + 500], ORDER_COLS)
            await conn.executemany(
                "INSERT INTO public.orders "
                "(id, tenant_id, customer_id, product_id, amount, order_date, status) "
                "VALUES ($1, $2, $3, $4, $5, $6, $7) ON CONFLICT (id) DO NOTHING",
                chunk,
            )
            inserted += len(chunk)
        for i in range(0, len(transactions), 500):
            chunk = _tuples(transactions[i:i + 500], TRANSACTION_COLS)
            await conn.executemany(
                "INSERT INTO public.transactions "
                "(id, tenant_id, amount, transaction_date, type, status) "
                "VALUES ($1, $2, $3, $4, $5, $6) ON CONFLICT (id) DO NOTHING",
                chunk,
            )
            inserted += len(chunk)
        for i in range(0, len(activity), 500):
            chunk = _tuples(activity[i:i + 500], ACTIVITY_COLS)
            await conn.executemany(
                "INSERT INTO public.activity "
                "(id, tenant_id, user_id, activity_date, event_type) "
                "VALUES ($1, $2, $3, $4, $5) ON CONFLICT (id) DO NOTHING",
                chunk,
            )
            inserted += len(chunk)

    return inserted


async def seed_database(
    connection_url: str | None,
    tenant_id: str = "00000000-0000-0000-0000-000000000001",
    num_tenants: int = 1,
) -> dict:
    """Insert synthetic data into the (migration-created) analytics tables."""
    stats = {"rows_inserted": 0, "tenants_seeded": 0}

    conn = await owner_connect(connection_url)
    try:
        for t in range(num_tenants):
            t_id = tenant_id if t == 0 else str(uuid.uuid4())
            stats["rows_inserted"] += await _seed_tenant(conn, t_id)
            stats["tenants_seeded"] += 1
            logger.info("Seeded tenant %s", t_id)
    finally:
        await conn.close()

    return stats


async def main():
    parser = argparse.ArgumentParser(
        description="Seed test database with synthetic data"
    )
    parser.add_argument(
```

```
        "--connection-url",
        default=None,
        help="Owner PostgreSQL connection URL (default: DATABASE_URL_SYNC)",
    )
    parser.add_argument(
        "--tenants", type=int, default=1,
        help="Number of tenants to seed (default: 1)",
    )
    parser.add_argument(
        "--clear", action="store_true",
        help="Truncate all seed tables before seeding",
    )
    args = parser.parse_args()

    conn = await owner_connect(args.connection_url)
    try:
        if args.clear:
            await _clear_data(conn)
    finally:
        await conn.close()

    stats = await seed_database(
        connection_url=args.connection_url,
        num_tenants=args.tenants,
    )

    print()
    print("=" * 50)
    print("Test Data Seed Complete")
    print("=" * 50)
    print("Tenants seeded:  " + str(stats["tenants_seeded"]))
    print("Rows inserted:   " + str(stats["rows_inserted"]))
    print()


if __name__ == "__main__":
    asyncio.run(main())
```

scripts/setup.sh

163 lines · 5.9 KB

```
#!/usr/bin/env bash
# GenBI first-time environment setup.
#
# Checks prerequisites, generates .env files, installs host-side dev deps,
# builds Docker images, starts infra, runs migrations, and verifies the stack.
# Safe to re-run: each step is skipped if its output already exists.
#
# This script does NOT install system packages (Docker, uv, pnpm) — host
# environments differ too much. It checks for them and tells you what's missing.

set -euo pipefail

REPO_ROOT="$(cd "$(dirname "${BASH_SOURCE[0]}")/.." && pwd)"
cd "$REPO_ROOT"

COMPOSE_DEV="docker compose -f infra/docker-compose.dev.yml"

# Color helpers
green() { printf "\033[32m%s\033[0m\n" "$1"; }
yellow() { printf "\033[33m%s\033[0m\n" "$1"; }
red()   { printf "\033[31m%s\033[0m\n" "$1"; }
step()  { printf "\n\033[1m  %s\033[0m\n" "$1"; }

# --- 1. Prerequisites -------------------------------------------------------
step "Checking prerequisites"

check_cmd() {
  if command -v "$1" >/dev/null 2>&1; then
    echo "    $1 ($(command -v "$1"))"
  else
    echo "    $1 not found"
    return 1
  fi
}

MISSING=0
check_cmd docker    || MISSING=1
check_cmd uv        || MISSING=1
check_cmd pnpm      || MISSING=1
check_cmd node      || MISSING=1
check_cmd python3   || MISSING=1

# Version checks
if command -v node >/dev/null 2>&1; then
  NODE_MAJOR="$(node -e 'console.log(process.versions.node.split(".")[0])')"
  if [[ "$NODE_MAJOR" -lt 22 ]]; then
    echo "      node $NODE_MAJOR found; GenBI targets Node 22+"
  fi
fi
if command -v python3 >/dev/null 2>&1; then
  PY_MINOR="$(python3 -c 'import sys; print(f"{sys.version_info.major}.{sys.version_info.minor}")')"
  if [[ "$PY_MINOR" < "3.12" ]]; then
    echo "      python $PY_MINOR found; GenBI targets 3.12+"
  fi
fi

if [[ "$MISSING" -eq 1 ]]; then
  echo ""
  red "Missing prerequisites. Install them, then re-run 'make setup':"
  echo "  • Docker:    https://docs.docker.com/get-docker/"
  echo "  • uv:        curl -LsSf https://astral.sh/uv/install.sh | sh"
  echo "  • pnpm:      npm install -g pnpm   (needs Node 22+)"
  exit 1
fi

# Docker daemon running?
if ! docker info >/dev/null 2>&1; then
  red "Docker daemon is not running. Start Docker Desktop (or dockerd) and re-run."
  exit 1
fi

# --- 2. Generate .env files -------------------------------------------------
step "Generating environment files"
scripts/gen-env.sh

# --- 3. Host-side dev dependencies ------------------------------------------
step "Installing host-side dev dependencies (for local non-Docker dev)"

if [[ ! -d backend/.venv ]]; then
  (cd backend && uv sync --dev)
  green "    backend deps installed (backend/.venv)"
else
  echo "    backend/.venv exists — skipping"
fi

# Generate a committed uv.lock if missing (Docker build wants it)
if [[ ! -f backend/uv.lock ]]; then
  (cd backend && uv lock)
  yellow "      Generated backend/uv.lock — commit it for reproducible builds"
fi

if [[ ! -d frontend/node_modules ]]; then
  (cd frontend && pnpm install)
  green "    frontend deps installed (frontend/node_modules)"
else
  echo "    frontend/node_modules exists — skipping"
fi

if [[ ! -f frontend/pnpm-lock.yaml ]]; then
  (cd frontend && pnpm install --lockfile-only 2>/dev/null || pnpm install)
```

```
  yellow "      Generated frontend/pnpm-lock.yaml — commit it for reproducible builds"
fi

# --- 4. Build images -------------------------------------------------------
step "Building Docker images"
$COMPOSE_DEV build
green "    images built"

# --- 5. Start infra (postgres + redis) and wait healthy ----------------------
step "Starting infrastructure services"
$COMPOSE_DEV up -d postgres redis

echo "  waiting for postgres + redis to be healthy..."
for i in $(seq 1 30); do
  STATUS="$($COMPOSE_DEV ps --format json 2>/dev/null | grep -c '"Health":"healthy"' || true)"
  if [[ "$STATUS" -ge 2 ]]; then break; fi
  sleep 2
done

# --- 6. Apply migrations ----------------------------------------------------
step "Applying database migrations"
# init.sql already created the baseline schema on first boot; alembic stamp
# marks the current revision so future `alembic upgrade head` works.
$COMPOSE_DEV exec -T backend sh -c \
  'uv run alembic current 2>/dev/null | grep -q "0001_baseline" \
   || (uv run alembic stamp 0001_baseline && echo "  stamped at 0001_baseline") \
   || echo "  (alembic not yet available — schema is via init.sql)"' \
  && green "    migrations current" \
  || yellow "      could not run alembic — schema is via init.sql; check 'make migrate' later"

# --- 7. Bring up the full stack ---------------------------------------------
step "Starting full stack"
$COMPOSE_DEV up -d
echo "  services starting in the background..."

# --- 8. Verify --------------------------------------------------------------
step "Verifying stack (this waits for services to become healthy)"
sleep 5
if scripts/verify.sh; then
  green ""
  green "  GenBI is up!"
else
  yellow ""
  yellow "    Some checks failed. The stack may still be starting —"
  echo "        wait 30s and run 'make verify' again."
  echo "        Tail logs with: make logs"
fi

# --- 9. Print access URLs ---------------------------------------------------
cat <<EOF

────────────────────────────────────────────────────────
 GenBI access URLs:
   Frontend:     http://localhost:3000
   Backend API:  http://localhost:8000/docs
   Health:       http://localhost:8000/api/v1/health
   Cube:         http://localhost:4000
   Prometheus:   http://localhost:9090
   Grafana:      http://localhost:3001   (admin/admin)
────────────────────────────────────────────────────────
 Next: set ANTHROPIC_API_KEY in backend/.env to enable LLM features.
 Common commands: make logs | make ps | make verify | make down
EOF
```

scripts/verify.sh

167 lines · 7.4 KB

```
#!/usr/bin/env bash
# Smoke test for a running GenBI stack. Checks each service is reachable and
# wired correctly. Exits non-zero on first failure with a readable message.
#
# Usage: scripts/verify.sh
# Assumes the dev stack is up (docker compose -f infra/docker-compose.dev.yml up -d)

set -uo pipefail

REPO_ROOT="$(cd "$(dirname "${BASH_SOURCE[0]}")/.." && pwd)"
PASS=0
FAIL=0
ok()   { echo "    $1"; PASS=$((PASS+1)); }
bad()  { echo "    $1"; FAIL=$((FAIL+1)); }

echo "GenBI stack verification"
echo "————————————————————————"

# --- helpers -----------------------------------------------------------------
# Run a psql command inside the postgres container.
psql_exec() {
  docker exec genbi-postgres psql -U genbi -d genbi -tA -c "$1" 2>/dev/null
}

# Same, but stderr kept — expected-failure checks need the error text.
psql_full() {
  docker exec genbi-postgres psql -U genbi -d genbi -tA -c "$1" 2>&1
}

# --- 1. Backend liveness -----------------------------------------------------
if curl -sf http://localhost:8000/api/v1/health >/dev/null 2>&1; then
  ok "backend liveness (GET /api/v1/health → 200)"
else
  bad "backend liveness — is the backend up on :8000?"
fi

# --- 2. Backend readiness (real DB + Redis pings) ----------------------------
READY=$(curl -sf http://localhost:8000/api/v1/health/ready 2>/dev/null || true)
if echo "$READY" | grep -q '"status":"ready"'; then
  ok "backend readiness (DB + Redis reachable, status=ready)"
elif echo "$READY" | grep -q '"status":"degraded"'; then
  bad "backend readiness DEGRADED: $(echo "$READY" | tr -d '\n')"
else
  bad "backend readiness — /health/ready did not return ready status"
fi

# --- 3. Cube -----------------------------------------------------------------
if curl -sf http://localhost:4000/ >/dev/null 2>&1 \
   || curl -sf http://localhost:4000/cubejs-api/v1/load >/dev/null 2>&1; then
  ok "cube reachable on :4000"
else
  bad "cube — not responding on :4000"
fi

# --- 4. Redis ----------------------------------------------------------------
if docker exec genbi-redis redis-cli ping 2>/dev/null | grep -q PONG; then
  ok "redis ping → PONG"
else
  bad "redis — no PONG (is genbi-redis running?)"
fi

# --- 5. Postgres: seed + AGE -------------------------------------------------
TENANT_COUNT="$(psql_exec 'SELECT count(*) FROM tenants;' 2>/dev/null || echo '0')"
if [[ "${TENANT_COUNT:-0}" -ge 1 ]]; then
  ok "postgres — seed tenant present ($TENANT_COUNT tenant(s))"
else
  bad "postgres — no tenants (init.sql / migrations not applied?)"
fi

# AGE check (optional — degrades gracefully if disabled)
AGE_GRAPH="$(psql_exec "SELECT count(*) FROM ag_catalog.ag_graph WHERE name='genbi_graph';" 2>/dev/null || echo '')"
if [[ "${AGE_GRAPH:-}" == "1" ]]; then
  ok "postgres — Apache AGE graph 'genbi_graph' exists"
elif [[ -z "${AGE_GRAPH:-}" ]]; then
  echo "      Apache AGE not available — lineage graph disabled (GENBI_ENABLE_AGE controls this)"
else
  bad "postgres — AGE present but 'genbi_graph' missing (run init_age_graph)"
fi

# --- 6. RLS enforcement (Phase 8b: runtime roles are actually bound) --------
# NOTE: multi-statement psql -c prints "SET" for the SET ROLE line — pipe
# through tail to keep only the query result.
APP_USERS="$(psql_exec "SET ROLE genbi_app; SELECT count(*) FROM users;" | tail -1 || true)"
if [[ "${APP_USERS:-x}" == "0" ]]; then
  ok "RLS enforced — genbi_app sees 0 users without tenant GUC"
else
  bad "RLS NOT enforced — genbi_app saw ${APP_USERS:-<error>} users without GUC (roles migrated?)"
fi

AUTH_USERS="$(psql_exec "SET ROLE genbi_auth; SELECT count(*) FROM users;" | tail -1 || true)"
if [[ "${AUTH_USERS:-x}" =~ ^[0-9]+$ ]] && [[ "${AUTH_USERS}" -ge 1 ]]; then
  ok "auth role can look up users ($AUTH_USERS row(s), cross-tenant by design)"
else
  bad "auth role cannot read users (login path broken — Alembic 0002 applied?)"
fi

AUTH_AUDIT="$(psql_full "SET ROLE genbi_auth; SELECT count(*) FROM audit_log;" || true)"
if echo "$AUTH_AUDIT" | grep -q "permission denied"; then
  ok "auth role is denied on non-users tables (audit_log)"
else
```

```
  bad "auth role could read audit_log — its grants are too broad"
fi

# --- 6b. Auth: login issues a JWT (needed by the checks below) ---------------
LOGIN_RESP="$(curl -sf -X POST http://localhost:8000/api/v1/auth/login \
  -H 'Content-Type: application/json' \
  -d '{"email":"admin@genbi.local","password":"admin123"}' 2>/dev/null || true)"
if echo "$LOGIN_RESP" | grep -q '"access_token"'; then
  ok "auth — login issues a JWT for the seeded dev user"
else
  bad "auth — /auth/login failed (seed user only exists on fresh volumes; try make reset)"
fi

# --- 6c. Tenant-scoped metric query (Phase 10: JWT → Cube → GUC → RLS) --------
TOKEN="$(echo "$LOGIN_RESP" | sed -n 's/.*"access_token":"\([^"]*\)".*/\1/p')"
if [[ -n "$TOKEN" ]]; then
  METRIC_RESP="$(curl -s -o /dev/null -w '%{http_code}' -X POST http://localhost:8000/api/v1/metrics/query \
    -H "Authorization: Bearer $TOKEN" -H 'Content-Type: application/json' \
    -d '{"measures":["Sales.revenue_total"],"limit":5}' 2>/dev/null || echo '000')"
  METRIC_BODY="$(curl -s -X POST http://localhost:8000/api/v1/metrics/query \
    -H "Authorization: Bearer $TOKEN" -H 'Content-Type: application/json' \
    -d '{"measures":["Sales.revenue_total"],"limit":5}' 2>/dev/null || true)"
  if [[ "$METRIC_RESP" == "200" ]] && echo "$METRIC_BODY" | grep -q '"data":\[{'; then
    ok "tenant-scoped metric query returns rows (JWT→Cube→GUC→RLS chain live)"
  elif [[ "$METRIC_RESP" == "200" ]]; then
    echo "      metric query OK but 0 rows — tenant has no seed data (make seed)?"
  else
    bad "metric query failed (HTTP $METRIC_RESP) — Cube tenant path broken?"
  fi
else
  echo "      skipping metric query check (no token — login failed above)"
fi

# --- 6d. NL2SQL schema grounding (Phase 11) -----------------------------------
EMBED_COUNT="$(psql_exec 'SELECT count(*) FROM schema_embeddings;' || echo 0)"
if [[ "${EMBED_COUNT:-0}" =~ ^[0-9]+$ ]] && [[ "${EMBED_COUNT}" -ge 1 ]]; then
  ok "schema embeddings present ($EMBED_COUNT tables) — NL2SQL schema grounding armed"
else
  echo "      schema_embeddings empty — run scripts/embed_schema.py to arm schema grounding"
fi

# --- 6e. Audit trail (Phase 12) -----------------------------------------------
AUDIT_ROWS="$(psql_exec 'SELECT count(*) FROM audit_log;' || echo 0)"
if [[ "${AUDIT_ROWS:-0}" =~ ^[0-9]+$ ]] && [[ "${AUDIT_ROWS}" -ge 1 ]]; then
  ok "audit trail active ($AUDIT_ROWS LLM-call entries recorded)"
else
  echo "      audit_log empty — rows appear after the first LLM-backed chat query"
fi

# --- 8. Frontend (optional — requires Node/pnpm toolchain to build) ----------
HTTP_CODE="$(curl -s -o /dev/null -w '%{http_code}' http://localhost:3000 2>/dev/null || echo '000')"
if [[ "${HTTP_CODE}" =~ ^([23][0-9][0-9])$ ]]; then
  ok "frontend served on :3000 (HTTP $HTTP_CODE)"
else
  echo "      frontend not reachable on :3000 (requires Node/pnpm to build; optional)"
fi

# --- 9. Prometheus (optional) -------------------------------------------------
if curl -sf http://localhost:9090/-/healthy >/dev/null 2>&1; then
  ok "prometheus healthy on :9090"
else
  echo "      prometheus not reachable on :9090 (optional)"
fi

echo "————————————————————————"
echo "Result: $PASS passed, $FAIL failed"
exit $(( FAIL > 0 ? 1 : 0 ))
```

semantic/cube/.env.example

15 lines · 0.6 KB

```
# Driver type for Cube built-in resolver (refresh scheduler runs without a
# securityContext, where the per-tenant driverFactory does not apply).
CUBEJS_DB_TYPE=postgres
# Cube.dev — standard CUBEJS_DB_* driver env (Cube connects as the read-only
# cube_reader role created by infra/postgres/init.sql). Hosts here are for
# host-based dev; scripts/gen-env.sh rewrites them to compose service names.
CUBEJS_DB_HOST=localhost
CUBEJS_DB_PORT=5432
CUBEJS_DB_NAME=genbi
CUBEJS_DB_USER=cube_reader
CUBEJS_DB_PASS=changeme_cube_reader

CUBEJS_DEV_MODE=true
CUBEJS_API_SECRET=change-me-in-production
CUBEJS_WEB_SOCKETS=true
```

semantic/cube/cube.js

48 lines · 1.9 KB

```
// GenBI Cube project — data models live in ./model/*.yml (10 cubes over the
// seeded analytics tables). ADR 007 documents the pivot to Cube-native
// models; ADR 008 documents the per-tenant data path below.
//
// Tenant isolation on data queries (/load):
//   1. The backend mints JWTs carrying a `tenantId` claim (Cube surfaces
//      custom claims as securityContext).
//   2. contextToOrchestratorId keys the orchestrator — and therefore the
//      driver pool — by tenant, so tenants never share connections.
//   3. driverFactory passes the pg `options` connection parameter setting
//      app.current_tenant_id — the exact GUC the Postgres RLS policies
//      consume (Phase 8b).
//
// Failure mode is CLOSED: if the GUC is ever missing, RLS returns zero rows
// for that query — it can never return another tenant's data. Metadata
// requests (/meta, no tenant claim) use the plain driver.

const PostgresDriver = require("@cubejs-backend/postgres-driver");

function baseDbConfig() {
  return {
    host: process.env.CUBEJS_DB_HOST,
    port: process.env.CUBEJS_DB_PORT ? parseInt(process.env.CUBEJS_DB_PORT, 10) : 5432,
    user: process.env.CUBEJS_DB_USER,
    password: process.env.CUBEJS_DB_PASS,
    database: process.env.CUBEJS_DB_NAME,
  };
}

module.exports = {
  apiSecret: process.env.CUBEJS_API_SECRET,
  // devMode is env-only in current Cube (CUBEJS_DEV_MODE in .env);
  // passing it here is rejected: Invalid cube-server-core options.

  contextToOrchestratorId: ({ securityContext }) =>
    (securityContext && securityContext.tenantId) || "anonymous",

  driverFactory: ({ securityContext }) => {
    const tenantId = securityContext && securityContext.tenantId;
    const config = baseDbConfig();
    if (tenantId) {
      // Session GUC on every pooled connection — consumed by the
      // tenant_isolation RLS policies.
      config.options = `-c app.current_tenant_id=${tenantId}`;
    }
    return new PostgresDriver(config);
  },
};
```

semantic/cube/model/activity.yml

43 lines · 1.2 KB

```yaml
# Product engagement events (source: public.activity). Joins WebUsers.
cubes:
  - name: Activity
    sql_table: public.activity
    data_source: default

    joins:
      - name: WebUsers
        relationship: many_to_one
        sql: "{CUBE}.user_id = {WebUsers}.id"

    measures:
      - name: event_count
        title: Event Count
        description: Number of engagement events (logins, queries, exports, ...)
        type: count
        format: number
      - name: unique_active_users
        title: Unique Active Users
        description: Distinct users with at least one event
        type: count_distinct
        sql: user_id
        format: number

    dimensions:
      - name: id
        title: ID
        sql: id
        type: string
        primaryKey: true
      - name: event_type
        title: Event Type
        description: Engagement event type (login, query, export, dashboard_view, report_create)
        sql: event_type
        type: string
      - name: activity_date
        title: Activity Date
        description: Date of the event
        sql: activity_date
        type: time
      - name: tenant_id
        sql: tenant_id
        type: string
```

semantic/cube/model/customers.yml

49 lines · 1.2 KB

```yaml
# Customer records (source: public.customers). Joined from Orders.
cubes:
  - name: Customers
    sql_table: public.customers
    data_source: default

    joins:
      - name: Orders
        relationship: one_to_many
        sql: "{CUBE}.id = {Orders}.customer_id"

    measures:
      - name: customer_count
        title: Customer Count
        description: Number of customer records
        type: count
        format: number
      - name: active_customers
        title: Active Customers
        description: Customers whose status is active
        type: count
        format: number
        filters:
          - sql: "{CUBE}.status = 'active'"

    dimensions:
      - name: id
        title: ID
        sql: id
        type: string
        primaryKey: true
      - name: country
        title: Country
        description: Customer country
        sql: country
        type: string
      - name: status
        title: Customer Status
        description: Customer lifecycle status (active, inactive, churned)
        sql: status
        type: string
      - name: signup_date
        title: Signup Date
        description: Date the customer signed up
        sql: signup_date
        type: time
      - name: tenant_id
        sql: tenant_id
        type: string
```

semantic/cube/model/deals.yml

62 lines · 1.7 KB

```yaml
# Sales pipeline deals (source: public.deals). Joins SalesReps and Regions
# (both real FKs in the seed data).
cubes:
  - name: Deals
    sql_table: public.deals
    data_source: default

    joins:
      - name: SalesReps
        relationship: many_to_one
        sql: "{CUBE}.rep_id = {SalesReps}.id"
      - name: Regions
        relationship: many_to_one
        sql: "{CUBE}.region_id = {Regions}.id"

    measures:
      - name: pipeline_value
        title: Pipeline Value
        description: Sum of deal amounts across all stages
        type: sum
        sql: amount
        format: currency
        meta:
          currency: USD
      - name: deal_count
        title: Deal Count
        description: Number of deals
        type: count
        format: number
      - name: won_deals
        title: Won Deals
        description: Deals closed as won
        type: count
        format: number
        filters:
          - sql: "{CUBE}.stage = 'closed_won'"
      - name: win_rate
        title: Win Rate
        description: Share of deals closed as won
        sql: "100.0 * {Deals.won_deals} / NULLIF({Deals.deal_count}, 0)"
        type: number
        format: percent

    dimensions:
      - name: id
        title: ID
        sql: id
        type: string
        primaryKey: true
      - name: stage
        title: Deal Stage
        description: Pipeline stage (prospecting, negotiation, closed_won, closed_lost)
        sql: stage
        type: string
      - name: close_date
        title: Close Date
        description: Date the deal closed (or is expected to close)
        sql: close_date
        type: time
      - name: tenant_id
        sql: tenant_id
        type: string
```

semantic/cube/model/orders.yml

54 lines · 1.3 KB

```yaml
# Customer orders (source: public.orders). Joins Customers on customer_id
# (real FK in the seed data).
cubes:
  - name: Orders
    sql_table: public.orders
    data_source: default

    joins:
      - name: Customers
        relationship: many_to_one
        sql: "{CUBE}.customer_id = {Customers}.id"

    measures:
      - name: order_count
        title: Order Count
        description: Number of orders placed
        type: count
        format: number
      - name: order_value
        title: Total Order Value
        description: Sum of order amounts
        type: sum
        sql: amount
        format: currency
        meta:
          currency: USD
      - name: avg_order_value
        title: Average Order Value
        description: Mean order amount
        type: avg
        sql: amount
        format: currency
        meta:
          currency: USD

    dimensions:
      - name: id
        title: ID
        sql: id
        type: string
        primaryKey: true
      - name: status
        title: Order Status
        description: Order fulfillment status (completed, pending, cancelled)
        sql: status
        type: string
      - name: order_date
        title: Order Date
        description: Date the order was placed
        sql: order_date
        type: time
      - name: tenant_id
        sql: tenant_id
        type: string
```

semantic/cube/model/products.yml

43 lines · 1.1 KB

```yaml
# Product catalog (source: public.products). NOTE: sales.product_id does not
# reference this table in the seed data — no join to Sales.
cubes:
  - name: Products
    sql_table: public.products
    data_source: default

    joins: []

    measures:
      - name: product_count
        title: Product Count
        description: Number of products in the catalog
        type: count
        format: number
      - name: avg_price
        title: Average Price
        description: Mean product list price
        type: avg
        sql: price
        format: currency
        meta:
          currency: USD

    dimensions:
      - name: id
        title: ID
        sql: id
        type: string
        primaryKey: true
      - name: product_name
        title: Product Name
        description: Product name
        sql: product_name
        type: string
      - name: category
        title: Category
        description: Product category (Software, Hardware, Services, Support, Training)
        sql: category
        type: string
      - name: tenant_id
        sql: tenant_id
        type: string
```

semantic/cube/model/regions.yml

35 lines · 0.9 KB

```yaml
# Region lookup (source: public.regions). Joined from Deals and SalesReps.
cubes:
  - name: Regions
    sql_table: public.regions
    data_source: default

    joins:
      - name: Deals
        relationship: one_to_many
        sql: "{CUBE}.id = {Deals}.region_id"
      - name: SalesReps
        relationship: one_to_many
        sql: "{CUBE}.id = {SalesReps}.region_id"

    measures:
      - name: region_count
        title: Region Count
        description: Number of regions
        type: count
        format: number

    dimensions:
      - name: id
        title: ID
        sql: id
        type: string
        primaryKey: true
      - name: region_name
        title: Region Name
        description: Region name (North, South, East, West, Central)
        sql: region_name
        type: string
      - name: tenant_id
        sql: tenant_id
        type: string
```

semantic/cube/model/sales.yml

63 lines · 1.7 KB

```yaml
# Sales revenue analytics (source: public.sales).
# NOTE: sales.product_id / sales.rep_id are NOT foreign keys to
# products/sales_representatives in the seed data — no joins declared here.
cubes:
  - name: Sales
    sql_table: public.sales
    data_source: default

    joins: []

    measures:
      - name: revenue_total
        title: Total Revenue
        description: Sum of revenue across all sales transactions
        type: sum
        sql: revenue
        format: currency
        meta:
          currency: USD
      - name: units_sold
        title: Units Sold
        description: Total units sold across all sales transactions
        type: sum
        sql: units
        format: number
      - name: avg_revenue_per_sale
        title: Average Revenue per Sale
        description: Mean revenue per sales transaction
        type: avg
        sql: revenue
        format: currency
        meta:
          currency: USD
      - name: sale_count
        title: Sale Count
        description: Number of sales transactions
        type: count
        format: number

    dimensions:
      - name: id
        title: ID
        sql: id
        type: string
        primaryKey: true
      - name: region
        title: Region
        description: Sales region name
        sql: region
        type: string
      - name: product_name
        title: Product Name
        description: Product name as recorded on the sale
        sql: product_name
        type: string
      - name: transaction_date
        title: Transaction Date
        description: Date of the sales transaction
        sql: transaction_date
        type: time
      - name: tenant_id
        sql: tenant_id
        type: string
```

semantic/cube/model/sales_reps.yml

41 lines · 1.0 KB

```yaml
# Sales representatives (source: public.sales_representatives).
# Joined from Deals; joins Regions.
cubes:
  - name: SalesReps
    sql_table: public.sales_representatives
    data_source: default

    joins:
      - name: Deals
        relationship: one_to_many
        sql: "{CUBE}.id = {Deals}.rep_id"
      - name: Regions
        relationship: many_to_one
        sql: "{CUBE}.region_id = {Regions}.id"

    measures:
      - name: rep_count
        title: Sales Rep Count
        description: Number of sales representatives
        type: count
        format: number

    dimensions:
      - name: id
        title: ID
        sql: id
        type: string
        primaryKey: true
      - name: name
        title: Rep Name
        description: Sales representative full name
        sql: name
        type: string
      - name: hire_date
        title: Hire Date
        description: Date the representative was hired
        sql: hire_date
        type: time
      - name: tenant_id
        sql: tenant_id
        type: string
```

semantic/cube/model/transactions.yml

55 lines · 1.4 KB

```yaml
# Financial transactions (source: public.transactions).
cubes:
  - name: Transactions
    sql_table: public.transactions
    data_source: default

    joins: []

    measures:
      - name: transaction_volume
        title: Transaction Volume
        description: Net sum of transaction amounts (withdrawals are negative in seed data)
        type: sum
        sql: amount
        format: currency
        meta:
          currency: USD
      - name: transaction_count
        title: Transaction Count
        description: Number of transactions
        type: count
        format: number
      - name: avg_transaction
        title: Average Transaction
        description: Mean transaction amount
        type: avg
        sql: amount
        format: currency
        meta:
          currency: USD

    dimensions:
      - name: id
        title: ID
        sql: id
        type: string
        primaryKey: true
      - name: type
        title: Transaction Type
        description: Transaction type (deposit, withdrawal, transfer, payment)
        sql: type
        type: string
      - name: status
        title: Transaction Status
        description: Transaction status (completed, pending, failed)
        sql: status
        type: string
      - name: transaction_date
        title: Transaction Date
        description: Date of the transaction
        sql: transaction_date
        type: time
      - name: tenant_id
        sql: tenant_id
        type: string
```

semantic/cube/model/web_users.yml

55 lines · 1.4 KB

```yaml
# Web users (source: public.web_users) — analytics accounts, NOT app logins.
# Joined from Activity (user_id is a real FK in the seed data).
cubes:
  - name: WebUsers
    sql_table: public.web_users
    data_source: default

    joins:
      - name: Activity
        relationship: one_to_many
        sql: "{CUBE}.id = {Activity}.user_id"

    measures:
      - name: user_count
        title: Web User Count
        description: Number of web-user records
        type: count
        format: number
      - name: active_users
        title: Active Web Users
        description: Web users whose status is active
        type: count
        format: number
        filters:
          - sql: "{CUBE}.status = 'active'"

    dimensions:
      - name: id
        title: ID
        sql: id
        type: string
        primaryKey: true
      - name: country
        title: Country
        description: User country
        sql: country
        type: string
      - name: status
        title: User Status
        description: User status (active, inactive)
        sql: status
        type: string
      - name: signup_date
        title: Signup Date
        description: Date the user signed up
        sql: signup_date
        type: time
      - name: last_login
        title: Last Login
        description: Date of the user's most recent login
        sql: last_login
        type: time
      - name: tenant_id
        sql: tenant_id
        type: string
```

semantic/README.md

57 lines · 2.8 KB

```
# Semantic Layer

> **Source of truth: Cube-native data models** (`semantic/cube/model/*.yml`).
> See [ADR 007](../docs/adr/007-cube-native-semantic-layer.md) for why the
> dbt MetricFlow scaffold was removed.

## What lives here

```
semantic/cube/
├── cube.js          ← Cube project config (API secret, dev mode)
├── .env.example     ← CUBEJS_DB_* (cube_reader role) + API secret template
└── schema/          ← THE CATALOG — one YAML per cube over the seeded
                       analytics tables (10 cubes, 23 measures)
    ├── sales.yml        revenue_total, units_sold, avg_revenue_per_sale, sale_count
    ├── orders.yml       order_count, order_value, avg_order_value (joins Customers)
    ├── customers.yml    customer_count, active_customers
    ├── transactions.yml transaction_volume, transaction_count, avg_transaction
    ├── web_users.yml    user_count, active_users (joins Activity)
    ├── activity.yml     event_count, unique_active_users (joins WebUsers)
    ├── deals.yml        pipeline_value, deal_count, won_deals, win_rate
    ├── sales_reps.yml   rep_count (joins Deals, Regions)
    ├── products.yml     product_count, avg_price
    └── regions.yml      region_count (joins Deals, SalesReps)
```

Principles

1. **Metrics are defined ONCE here.** Never re-implement metric logic in
   agent prompts or service code — the backend consumes them via
   `CubeClient` (`/meta` for definitions, `/load` for data).
2. **Every cube carries `tenant_id`** as a (hidden) dimension — the hook for
   per-tenant query rewriting when data queries land (Phase 10).
3. **Joins follow real foreign keys only.** `sales.product_id`/`rep_id` are
   dangling in the seed data, so Sales deliberately declares no joins.
4. **Structural tests guard the catalog** (`backend/tests/semantic/test_catalog.py`):
   unique names, symmetric joins, tenant dimension, core metrics present.

Runtime

- The `cube` service (dev compose) mounts this directory at `/cube/conf`
  and connects to Postgres as the read-only `cube_reader` role.
- `scripts/gen-env.sh` writes `semantic/cube/.env` with the compose-network
  host and the shared `CUBEJS_API_SECRET` (aligned with the backend).
- Metric metadata (`/meta`) powers the NL2SQL agent's metric context and
  `GET /api/v1/metrics/list`. Data queries (`/metrics/query`) are Phase 10:
  analytics tables are RLS-enforced, so Cube needs per-tenant GUC driver
  work first (see ADR 006/007).

Adding a metric

1. Edit the cube YAML in `schema/` (add a measure, or a new cube file for a
   new table).
2. Run `uv run pytest tests/semantic/test_catalog.py` — structural invariants
   must hold.
3. Restart the cube service (`make restart`); `/meta` reflects the change
   (CubeClient caches for 5 min).
```

todo.md

1251 lines · 64.7 KB

```
# GenBI Platform — Build Progress

> **Last updated:** 2026-08-15 | **Stack tier:** Enterprise | **84/84 tasks complete (Phases 1–16 — full original vision shipped)**

---

## Phase 1 — Scaffolding (Tasks 1–10)

| # | Task | Status |
|---|---|---|
| 1 | Root project configuration (CLAUDE.md, .claude/rules, .claude/skills) |  |
| 2 | Backend directory structure (FastAPI, agents, services, models, connectors) |  |
| 3 | Frontend directory structure (Next.js 15 App Router, shadcn/ui base) |  |
| 4 | Semantic layer structure (dbt MetricFlow + Cube.dev) |  |
| 5 | Infrastructure config (Docker Compose, PostgreSQL init, CI) |  |
| 6 | .claude/ rules and skills (api-design, llm-patterns, sql-safety) |  |
| 7 | Documentation (ADR records, OpenAPI specs) |  |
| 8 | Flint Chart integration module (ChartAssemblyInput bridge) |  |
| 9 | Final structure verification (85 files across 7 top-level dirs) |  |
| 10 | Finished Phase 1 |  |

---

## Phase 2 — Core Pipeline (Tasks 11–18)

| # | Task | Status |
|---|---|---|
| 11 | NL2SQLAgent with pgvector schema embedding retrieval |  |
| 12 | ValidationAgent with SQL safety gate (read-only enforcement) |  |
| 13 | NarrativeAgent for data storytelling and insight generation |  |
| 14 | Centralized LLM client with timeout, retry, and audit logging |  |
| 15 | PostgreSQL connector with read-only enforcement and parameterized queries |  |
| 16 | Cube.dev semantic layer client — async REST client with metric formatting for LLM context |  |
| 17 | PII masking for column-level data protection (per-tenant, role-aware) |  |
| 18 | End-to-end chat pipeline (NL → SQL → Chart → Narrative → AuditLog) |  |

---

## Phase 3 — Infrastructure & Quality (Tasks 19–23)

| # | Task | Status |
|---|---|---|
| 19 | Schema embedding pipeline — nightly pgvector sync from information_schema |  |
| 20 | Golden NL2SQL evaluation suite — 20 queries, 3 tolerance levels, 90% threshold |  |
| 21 | LLM mock fixtures for all agent tests |  |
| 22 | OpenTelemetry + Langfuse observability (7 counters/histograms, Prometheus endpoint) |  |
| 23 | CI/CD pipeline — GitHub Actions with lint, test, eval gate, and Docker build |  |

---

## Phase 4 — Frontend & Streaming (Tasks 24–28)

| # | Task | Status |
|---|---|---|
| 24 | Frontend app shell with Next.js App Router, shadcn/ui, and auth guard |  |
| 25 | Zod validation, JWT auth flow, 11 shadcn/ui components, SSE streaming chat UI |  |
| 26 | Apache AGE graph database — 5 vertex labels, 6 edge types, openCypher lineage and impact analysis |  |
| 27 | Test data infrastructure — 10 tables with synthetic generators, multi-tenant seed script |  |
| 28 | Progressive SSE streaming pipeline — start → intent → sql → validation → data → chart → narrative → done |  |

---

## Phase 5 — Performance & Safety (Tasks 29–30)

| # | Task | Status |
|---|---|---|
| 29 | Redis caching layer — L1 (in-memory LRU) + L2 (Redis), typed get/set for schema, metrics, results, charts, LLM responses, sessions, and rate li
mits |  |
| 30 | Chart hallucination detection — 6 validation categories (field existence, type compatibility, data ranges, null handling, structural validity)
with auto-correction engine |  |

---

## Integration Map

```
User Query
  → RouterAgent (intent classification)
    → NL2SQLAgent (metric context from Cube.dev, cached at L2)
      → ValidationAgent (SQL safety gate)
        → Connector (read-only execution, results cached at L1+L2)
          → ChartGenAgent → ChartValidator (hallucination check + auto-correct, spec cached at L2)
            → NarrativeAgent (insight paragraph)
              → AuditLog (persist trace)
```

Key files delivered

File	Lines	Purpose
`backend/app/core/cache.py`	488	Dual-tier Redis + in-memory caching service
`backend/app/agents/validation/chart_validator.py`	487	Deterministic chart hallucination detector
`backend/app/services/chat_service.py`	593	Full pipeline orchestrator with caching and validation
`backend/app/semantic/cube_client.py`	731	Cube.dev REST API client with metric-to-LLM formatting
`backend/app/db/graph_schema.py`	447	Apache AGE lineage graph (openCypher)
`backend/app/agents/chart_gen_agent.py`	204	Flint MCP chart spec generator
`frontend/src/components/chat/chat-view.tsx`	—	Streaming chat UI with SSE consumption
`scripts/embed_schema.py`	—	Nightly pgvector schema sync
`scripts/seed_test_data.py`	509	10-table synthetic test data generator

```

---

## Phase 5b — Environment Provisioning & Bootstrap (new)

>   **STATUS: VERIFIED 2026-08-06.** Closed by Phase 7 Tasks 1–3. Build,
> test run, and `make verify` all green; see Phase 7 for details.

> **Added:** 2026-08-05. Goal: a single `make setup && make up` (or one shell
> script) takes a clean machine to a fully running platform — all services
> healthy, dependencies installed, DB migrated and seeded, secrets generated,
> smoke test green. Today this is impossible: the platform will not boot
> end-to-end as committed. Every defect below is verified against the tree.

| # | Task | Severity | Status |
|---|---|---|---|
| 1 | Author one-command bootstrap script (`scripts/setup.sh`) + Makefile | Blocker | |
| 2 | Fix the AGE-on-pgvector image problem so `init.sql` stops failing | Blocker | |
| 3 | Add the missing config/lock files the Dockerfiles require | Blocker | |
| 4 | Add `alembic.ini` + first migration; reconcile the three schema sources | Blocker | |
| 5 | Healthchecks + dependency ordering on every compose service | High | |
| 6 | Generated `.env` files with random secrets; `make secrets` helper | High | |
| 7 | Postgres role for Apache AGE + Cube dev config | Medium | |
| 8 | `make verify` smoke test (health endpoints + DB round-trip) | Medium | |

---

### Task 1 — One-command bootstrap (`scripts/setup.sh` + `Makefile`)

**Why it matters:** No bootstrap exists today (no `Makefile`, no `setup.sh`,
no `bootstrap.*`). A new contributor must read 5 docs and still hits failures.
CLAUDE.md §5 lists commands but assumes a working environment.

**Deliverable — `Makefile` at repo root with these targets** (each idempotent):
```make
.PHONY: setup up down logs ps verify clean secrets migrate seed reset help

setup:           ## First-time: check prereqs, gen secrets, install deps, build
 up:             ## Start dev stack (docker compose up -d)
 down:           ## Stop dev stack
 logs:           ## Tail logs
 ps:             ## Show service status
 verify:         ## Health + smoke checks (Task 8)
 secrets:        ## Regenerate .env secrets
 migrate:        ## Run alembic upgrade head inside backend
 seed:           ## Load synthetic test data
 reset:          ## down -v && up (NUKES data)
 clean:          ## down -v, prune images, remove node_modules/.venv
 help:           ## Self-documenting target list
```

Deliverable — `scripts/setup.sh` that `make setup` calls. Steps:
1. **Prereq check** — verify `docker`, `docker compose v2`, `uv`, `pnpm`,
   `node >=22`, `python >=3.12` are installed; fail with a clear message + install
   hint if any missing. Do NOT auto-install system packages (host differs).
2. **Generate `.env` files** from templates (Task 6) — `backend/.env`,
   `semantic/cube/.env` — overwriting only if absent (prompt before overwrite).
3. **Install host-side dev tooling** (needed for local non-Docker dev per
   CLAUDE.md §5): `cd backend && uv sync --dev`; `cd frontend && pnpm install`.
4. **Build images**: `docker compose -f infra/docker-compose.dev.yml build`.
5. **Start infra only** (`postgres redis`) and wait-healthy.
6. **Run migrations** (Task 4) + optional `make seed`.
7. **Print the access URLs**: backend `:8000/docs`, frontend `:3000`,
   prometheus `:9090`, grafana `:3001`, cube `:4000`, OTLP `:4318`.
8. **Run `make verify`** at the end; exit non-zero if it fails.

Idempotency rule: re-running `make setup` must be safe and fast — skip
steps whose outputs already exist (check for `backend/.env`, `backend/.venv`,
`frontend/node_modules`, built images).

Done when: on a machine with only Docker + uv + pnpm installed,
`git clone && make setup` reaches `make verify` green with no manual edits.

Task 2 — Resolve AGE-on-pgvector (the platform will not init without this)

Why it matters (BLOCKER): `init.sql:11` runs `CREATE EXTENSION age;` and
`:12` runs `LOAD 'age';`, but the image is `pgvector/pgvector:pg16`
(`docker-compose.dev.yml:6`, `docker-compose.yml:3`) which bundles **pgvector,
not Apache AGE**. The init script fails on first boot, the `genbi_graph` graph
in `graph_schema.py:83` can't be created, and `init_age_graph()` crashes every
tenant that touches lineage. This is the single most blocking provisioning
defect.

Options — pick one and document the choice in a new `docs/adr/005-age-and-pgvector-image.md`:

- **(A) Custom Postgres image with both extensions** *(recommended for dev)* —
  add `infra/postgres/Dockerfile`:
  ```dockerfile
  FROM pgvector/pgvector:pg16
  RUN apk add --no-cache --virtual .age-build build-base git clang llvm-dev \
      && git clone --depth 1 https://github.com/apache/age /age \
      && cd /age && make PGDIR=/usr/local/pgsql install \
      && apk del .age-build
  ```
  Point both compose files at `build: ./infra/postgres`. Pin the AGE git ref.

- **(B) Use `apache/age:PG16`** and add pgvector into it** — symmetric to (A);
  AGE upstream image + pgvector build. More work; AGE's image is less maintained.

- **(C) Make AGE optional for dev** — wrap `CREATE EXTENSION age` in a guarded
  block behind a `GENBI_ENABLE_AGE=true` env flag (default false in dev, true in
  prod image). `graph_schema.py` already degrades; make the failure a logged
  warning instead of a boot crash. This unblocks everyone immediately while (A)
  is built. **Recommended as the first commit**, then layer (A) on top.

Verify: `docker compose up postgres` exits 0 with init.sql applied;
`SELECT * FROM ag_catalog.cypher('genbi_graph', $$RETURN 1$$) AS (a agtype);`
returns a row.

Task 3 — Add the missing files the Dockerfiles/compose require

Why it matters (BLOCKER): the committed Dockerfiles reference files that
don't exist; builds fail. Verified missing:

File	Referenced at	Effect if absent
`backend/uv.lock`	`backend/Dockerfile:9` (`COPY pyproject.toml uv.lock*`) + `:10` (`uv sync --frozen`)	`--frozen` aborts: "lockfile not found". The `*` glob hides the absence at COPY time.
`frontend/pnpm-lock.yaml`	`frontend/Dockerfile:4` (`COPY package.json pnpm-lock.yaml*`) + `:5` (`pnpm install --frozen-lockfile`)	`--frozen-lockfile` errors with no lockfile.
`backend/.env`	`docker-compose.dev.yml:43` (`env_file: ../backend/.env`)	Compose **fails to start** backend ("env file not found").
`semantic/cube/.env`	`docker-compose.dev.yml:79`	Compose fails to start cube.
`alembic.ini`	CI runs `uv run alembic upgrade head` (`ci.yml`)	Alembic can't locate config. (Tracked under Task 4.)

Fix:
1. Generate `backend/uv.lock`: `cd backend && uv lock`. Commit it. Re-running
   `uv sync --frozen` in the Dockerfile will then succeed.
2. Generate `frontend/pnpm-lock.yaml`: `cd frontend && pnpm install`. Commit it.
3. Don't commit `.env` files (CLAUDE.md §6 rule). Instead: `setup.sh` generates
   them from `.env.example` / `semantic/cube/.env.example` on first run, and
   `.gitignore` already excludes `*.env` (verify). Provide a checked-in
   `backend/.env.example` mirroring root `.env.example` if one doesn't exist.

Note on `--frozen`: if `uv.lock`/`pnpm-lock.yaml` are absent, either drop
`--frozen`/`--frozen-lockfile` for dev or (better) commit real lockfiles.
Lockfiles must be committed — reproducible builds depend on it.

Done when: `docker compose -f infra/docker-compose.dev.yml build` succeeds
with no "file not found" or "lockfile missing" errors, cold.

Task 4 — `alembic.ini` + first migration; kill the three-source schema drift

Why it matters (BLOCKER): DB schema today is defined in **three places that
drift**: `infra/postgres/init.sql` (Docker entrypoint), `scripts/seed_test_data.py:246-369`
(hand-written DDL), and `backend/app/db/models.py` (ORM, only 3 of ~13 tables).
There is **no `alembic.ini`** and `migrations/versions/` holds only `.gitkeep`
— zero migrations. CI's `alembic upgrade head` is a no-op that can't locate
config. `embed_schema.py:176-190` upserts into columns (`table_schema`,
`full_name`, `columns_json`, `embedding_text`, `updated_at`) that don't exist in
`init.sql:83-91` — the nightly sync errors against the initialized schema.

Fix:
1. **Add `backend/alembic.ini`** with `[alembic] script_location = app/db/migrations`
   and `sqlalchemy.url =` empty (overridden by `migrations/env.py:18` from
   `settings.DATABASE_URL_SYNC`). `env.py` already exists and is correct.
2. **Generate the first migration** from the ORM as the source of truth:
   `cd backend && uv run alembic revision --autogenerate -m "initial schema"`.
   But first expand `db/models.py` to cover ALL tables in `init.sql`
   (`audit_log`, `conversations`, `schema_embeddings`, `agent_examples`,
   `tenants`, `users`) so autogenerate produces a complete baseline. Use
   `mapped_column` / SQLAlchemy 2.0 style per CLAUDE.md §7.
3. **Decide ownership:** `init.sql` becomes the Docker-only bootstrap
   (extensions + roles + the seed tenant only); Alembic owns all table DDL
   going forward. Move the `CREATE TABLE` block out of `init.sql` into the
   first migration. `seed_test_data.py` stops issuing DDL and only `INSERT`s.
4. **Fix `embed_schema.py:176-190`** to match the actual `schema_embeddings`
   columns (`table_name, column_name, description, embedding, tenant_id,
   created_at`) — or, better, expand the table schema to hold what the script
   needs and put that change in the migration.
5. Empty `migrations/script.py.mako` (0 bytes) — replace with the standard
   Alembic mako template so `alembic revision` works.

Done when: `make migrate` on a fresh DB (only extensions + tenant seed
applied by `init.sql`) creates every table; `init.sql` and `db/models.py` agree;
`embed_schema.py` runs without `UndefinedColumn` errors.

Task 5 — Healthchecks + ordering on every compose service

Why it matters: Dev compose has healthchecks only on `postgres` and `redis`
(`docker-compose.dev.yml:17-21, :29-32`). Backend depends on both via
`condition: service_healthy` (`:47-50`) — good. But `cube`, `prometheus`,
`grafana`, `otel_collector` have **no healthcheck and no dependency edges**;
they race the backend on boot. **Prod compose** (`docker-compose.yml`) has
zero healthchecks anywhere and `backend depends_on: [postgres, redis]`
without `condition` (`:32-34`) — backend can start before Postgres accepts
connections, causing boot loops on cold start.

Fix:
- Add healthchecks to `cube` (`/cubejs-api/v1/load` or a `/readyz`), `backend`
  (`curl http://localhost:8000/api/v1/health` — verify that route exists), and
  the prod `postgres`/`redis` services. Install `curl` in slim images or use
  `python -c "urllib.request..."` to avoid adding packages.
- Convert prod `depends_on` lists to the long form with
  `condition: service_healthy`.
- In dev, make `cube` `depends_on: postgres (healthy)` (Cube reads from PG).

Done when: `docker compose up -d` from a clean state reaches all-green
health with no restart loops; `docker compose ps` shows every service
`(healthy)`.

Task 6 — Generated `.env` files with random secrets + `make secrets`

Why it matters: CLAUDE.md §6 forbids committing `.env`, but nothing generates them — compose fails (Task 3). `.env.example` has placeholder secrets (`JWT_SECRET_KEY=change-me-in-production-use-openssl-rand-64`). Without a generator, people either commit real secrets or copy placeholders verbatim (both bad). This is the provisioning-side twin of Phase 6 Task 2's fail-fast.

Deliverable — `scripts/gen-env.sh`:
- Reads root `.env.example` and writes `backend/.env`, substituting:
  - `JWT_SECRET_KEY` → `openssl rand -hex 32`
  - `TENANT_ENCRYPTION_KEY` → `python -c "from cryptography.fernet import Fernet; print(Fernet.generate_key().decode())"`
  - `CUBE_API_SECRET` → `openssl rand -hex 32`
  - `ANTHROPIC_API_KEY` → leave placeholder, warn the user to fill it
  - `DATABASE_URL` → keep example value but ensure it matches the compose `POSTGRES_USER/PASSWORD/DB`
- Reads `semantic/cube/.env.example` → writes `semantic/cube/.env` with a matching `CUBESQL_DB_HOST=postgres`, the same `CUBE_API_SECRET`, and a generated `CUBEJS_DB_PASS` matching `POSTGRES_PASSWORD`.
- Refuses to overwrite an existing `.env` unless `--force`.
- `make secrets` re-runs it (alias for `scripts/gen-env.sh`).

Done when: after `make secrets`, every `env_file:` reference in both compose files resolves; no placeholder secret remains except `ANTHROPIC_API_KEY` (which is gated behind an explicit warning).

Task 7 — Postgres role for AGE + Cube dev config

Why it matters: Apache AGE requires a superuser (or a role granted `ag_catalog`) to create/load the extension and graphs. The `genbi` role in `init.sql` is the DB owner but not a superuser, so `init_age_graph()` (`graph_schema.py:75`) will fail with permission denied at runtime even after Task 2 fixes the image. Separately, Cube needs a DB role with read access to the analytic tables; none is provisioned.

Fix (in `init.sql` post-extensions block):
```sql
-- AGE: grant the genbi role access to the graph catalog
GRANT USAGE ON SCHEMA ag_catalog TO genbi;
GRANT ALL ON SCHEMA ag_catalog TO genbi;
-- (For dev only; in prod prefer a dedicated age_admin role.)
```
- Add a read-only `cube_reader` role for Cube: `CREATE ROLE cube_reader LOGIN PASSWORD '...'; GRANT USAGE ON SCHEMA public TO cube_reader; GRANT SELECT ON ALL TABLES IN SCHEMA public TO cube_reader;` and reference it from `semantic/cube/.env`. Wire the password via `gen-env.sh` (Task 6).

Done when: as the `genbi` role, `SELECT ag_catalog.create_graph('test');` succeeds; Cube can connect and run `/load` against a seeded table.

Task 8 — `make verify` smoke test

Why it matters: No way to confirm the stack is actually up and wired without manually hitting endpoints. Closes the provisioning loop.

Deliverable — `scripts/verify.sh` (called by `make verify`) that checks, exiting non-zero on first failure with a readable message:
1. `GET http://localhost:8000/api/v1/health` → 200 — **CONFIRMED working**; liveness probe returns `{"status":"ok","version":"0.1.0"}`. Route resolves via `main.py:41` (`/api/v1`) → `router.py:9` (health, no prefix) → `health.py:8`.
2. `GET http://localhost:8000/api/v1/health/ready` → 200 with real pings — **REMEDIATED** (was a stub returning hardcoded `"connected"` for DB/Redis even when down). Now executes `SELECT 1` on the async engine, pings Redis via `cache.ping()` (non-memoized), returns **503** if either is unreachable, and reports `"configured"`/`"unconfigured"` for Flint instead of a fake `"connected"`. Changes in `health.py`, `cache.py` (`RedisCache.ping`).
3. `GET http://localhost:4000/cubejs-api/v1/load` → responds (Cube reachable).
4. `redis-cli -h localhost ping` → PONG.
5. `psql` into postgres → `SELECT count(*) FROM tenants;` returns >= 1 (seed applied); `SELECT * FROM ag_catalog.cypher('genbi_graph', $$RETURN 1$$) AS (a agtype);` returns 1 (AGE working).
6. `curl http://localhost:3000` → 200 (frontend served).
7. Optional: `curl http://localhost:9090/-/healthy` (Prometheus).

Done when: `make verify` prints a green checklist and exits 0 on a freshly-provisioned stack; fails loudly on any broken wire. (The health-route sub-task is already closed — see Task 8 notes in the change log.)

Sequencing

Tasks **2 and 3 are the unblockers** — do them first (the platform won't boot without them). Then **4** (schema ownership), then **1/6/7/8** (the bootstrap glue that ties it together), then **5** (polish). Concretely:

1. Task 3 (lockfiles + .env generation) — fastest, unblocks all Docker builds.
2. Task 2 (AGE image) — unblocks DB init.
3. Task 4 (alembic + schema reconciliation) — unblocks migrations + embed pipeline.
4. Task 6 → Task 1 → Task 7 → Task 8 — the bootstrap experience.
5. Task 5 — hardening of the compose files.

After Phase 5b lands, Phase 6 hardening (RLS, masking, observability wiring) can be verified against a running stack rather than by static reading.

Phase 6 — Security & Integrity Hardening (post-review)

> **STATUS: VERIFIED 2026-08-06.** Closed by Phase 7 Tasks 1–3. Test suite green (57 passed), stack boots, readiness probes pass.

> **Added:** 2026-08-05, following a full codebase review. Each item closes a gap
> between a documented guarantee (CLAUDE.md) and the actual code. All findings
> have file:line citations verified against the current tree.
>
> Note: original task list had 5 numbered lines; line 2 ("plumbing that fulfill
> documented guarantees.") is a line-wrap continuation of line 1, so items 1+2
> are merged below.

#	Task	Severity	Status
1	Wire PII masking into the pipeline + attach observability in `main.py`	High	
2	Add RLS + `FORCE` to every tenant table; fail-fast on default JWT secret	High	
3	Route `ChartGenAgent` through `llm_client`; dereference `settings.LLM_*_MODEL`	Medium	
4	Rewrite the golden NL2SQL eval to actually invoke `NL2SQLAgent`	Medium	

Task 1 — PII masking + observability wiring (was review items #2, #6)

Why it matters: CLAUDE.md §7 (line 207) promises *"no PII ever enters LLM
context"*, and §12 promises OpenTelemetry + Langfuse + a Prometheus `/metrics`
endpoint. Neither is currently true.

Defect A — masking is dead code. `masking.py` is fully implemented
(`PIIMasker.mask_rows` at `:91`, `get_masker_for_tenant` at `:212`) but is
never imported outside its own module (grep-confirmed). Query results flow
from `PostgreSQLConnector.execute` → `ChatService._step_execute`
(`chat_service.py:383`) → unmodified into `_step_chart`
(`chart_gen_agent.py:91` ships `data[:10]` to the LLM) and `_step_narrative`
(`chat_service.py:528` ships `data[:5]` as `data_summary["head"]`).

Fix A — single chokepoint in `_step_execute`:
- In `chat_service._step_execute`, after `results = await connector.execute(sql)`
  and **before** `cache.set_query_result(...)`, apply:
  ```python
  from app.core.masking import get_masker_for_tenant
  results = get_masker_for_tenant(self.tenant_id).mask_rows(results)
  ```
- Masking before caching means **cached rows are also safe** — no PII persists
  in Redis/L1 either. This is the desired property for analytics.
- Add a unit test in `tests/services/` feeding a row with an `email` column and
  asserting the masked value reaches the chart/narrative inputs.

Defect B — observability never attached. `observability.py` defines
`init_tracing()` (`:60`), `init_metrics()` (`:124`), `instrument_app()` (`:95`),
but `main.py` `lifespan` calls only `setup_logging()` (`main.py:16`).

Fix B — wire in `main.py`:
- In `create_app()`, after `app.include_router(...)`, call `instrument_app(app)`
  (must run after routes are registered).
- In `lifespan` startup: `tracer_provider = init_tracing()`,
  `meter_provider = init_metrics()`; on shutdown: store providers and call
  `.shutdown()` on each, plus `get_langfuse_tracer().flush()`.
- Wrap each init in try/except so a missing collector doesn't crash boot.

** Gotcha — Prometheus scrape target mismatch.** `init_metrics` uses
`PrometheusMetricReader`, which serves metrics on its own port (default 9464),
not as a FastAPI `/metrics` route. But `infra/prometheus/prometheus.yml:5-7`
scrapes `backend:8000`. Either (a) add a FastAPI `/metrics` route using
`prometheus_client.generate_latest`, or (b) point `prometheus.yml` at `:9464`.
Pick (a) — it's the conventional shape and matches the existing "endpoint at
/metrics" comment. Flag this as part of Task 1.

Done when: a query result with a `phone` column is masked before reaching
both the chart-spec prompt and the narrative prompt; `/metrics` returns the 7
defined counters/histograms; OTel spans export when `OTEL_EXPORTER_OTLP_ENDPOINT`
is set.

Task 2 — RLS everywhere + JWT secret fail-fast (was review items #3, #4)

Why it matters: CLAUDE.md §8 (line 220) mandates *"RLS policy must exist
for every user-data table"*. Currently RLS is enabled on exactly **1 of 5**
tenant-scoped tables. And `JWT_SECRET_KEY` defaults to `"change-me"`
(`config.py:47`), so any deployment that forgets the env var signs tokens with
a publicly known secret.

Defect A — RLS coverage. `infra/postgres/init.sql` enables RLS only on
`users` (`:43`). Missing: `audit_log` (`:47`), `conversations` (`:68`),
`schema_embeddings` (`:83`), `agent_examples` (`:101`). `FORCE ROW LEVEL
SECURITY` is never set, so even the `users` policy is bypassed by the table
owner (the app's DB role).

Fix A — append to `init.sql` after the table definitions:
```sql
ALTER TABLE audit_log         ENABLE ROW LEVEL SECURITY;
ALTER TABLE conversations      ENABLE ROW LEVEL SECURITY;
ALTER TABLE schema_embeddings  ENABLE ROW LEVEL SECURITY;
ALTER TABLE agent_examples     ENABLE ROW LEVEL SECURITY;

-- Close the owner-bypass loophole on ALL tenant tables
ALTER TABLE users              FORCE ROW LEVEL SECURITY;
ALTER TABLE audit_log          FORCE ROW LEVEL SECURITY;
ALTER TABLE conversations      FORCE ROW LEVEL SECURITY;
ALTER TABLE schema_embeddings  FORCE ROW LEVEL SECURITY;
ALTER TABLE agent_examples     FORCE ROW LEVEL SECURITY;

CREATE POLICY tenant_isolation ON audit_log         USING (tenant_id = current_setting('app.current_tenant_id')::UUID);
CREATE POLICY tenant_isolation ON conversations      USING (tenant_id = current_setting('app.current_tenant_id')::UUID);
CREATE POLICY tenant_isolation ON schema_embeddings  USING (tenant_id = current_setting('app.current_tenant_id')::UUID);
CREATE POLICY tenant_isolation ON agent_examples     USING (tenant_id = current_setting('app.current_tenant_id')::UUID);
```

**   Gotcha 1 — nullable tenant_id.** `schema_embeddings` and `agent_examples`
declare `tenant_id UUID REFERENCES tenants(id)` **without `NOT NULL`**
(`init.sql:89, :108`). RLS compares `tenant_id = current_setting(...)`, so NULL
rows are invisible to everyone — which may silently hide shared/global schema
rows. Decide: either (a) `ALTER COLUMN tenant_id SET NOT NULL` (requires
backfill), or (b) adjust policy to `USING (tenant_id IS NULL OR tenant_id =
current_setting('app.current_tenant_id')::UUID)` if global/shared rows are
intended. Resolve before applying FORCE.

**   Gotcha 2 — the GUC is never set.** Even the existing `users` policy can't
fire: nothing in the app calls `SET app.current_tenant_id`. The connector must
issue `SET LOCAL app.current_tenant_id = '<jwt tenant_id>'` per transaction
inside `execute()`/`execute_raw()` (alongside the existing `SET TRANSACTION READ
ONLY`). Without this, RLS either blocks everything or (without FORCE) blocks
nothing. **This is a required companion change** — wire it from the JWT
`tenant_id` already extracted in `auth.py:44`.

**   Gotcha 3 — protected file.** `init.sql` is in the CLAUDE.md §13
DO-NOT-MODIFY list (`:302`). Adding RLS *strengthens* safety rather than
relaxing it, but call it out in the PR description for visibility.

Defect B — default JWT secret. `config.py:47`.

Fix B — pydantic v2 model_validator in `Settings`:
```python
@model_validator(mode="after")
def _enforce_production_secrets(self):
    insecure = self.APP_ENV not in ("development", "test") and self.JWT_SECRET_KEY == "change-me"
    if insecure:
        raise ValueError("JWT_SECRET_KEY must be set in non-development environments")
    return self
```
Consider the same pattern for `ANTHROPIC_API_KEY == ""` (optional, out of scope
unless desired).

Done when: a query through the connector sets the tenant GUC and RLS
enforces isolation (test: insert a row under tenant A, query as tenant B,
expect 0 rows); booting with `APP_ENV=production JWT_SECRET_KEY=change-me`
raises immediately.

Task 3 — Route ChartGenAgent through llm_client + kill hardcoded models (was #7, #8)

Why it matters: CLAUDE.md §6 (line 175) — *"NEVER call LLM APIs without
[llm_client]"*; line 174 — *"NEVER hardcode model names"*. Both are violated.

Defect A — ChartGenAgent bypasses the wrapper. `chart_gen_agent.py:79`
constructs `ChatAnthropic(...)` directly and calls `llm.ainvoke(prompt)`,
skipping retry, token-budget, JSON extraction, and **audit logging**.

Fix A: rewrite `_generate_chart_spec` to use the centralized client:
```python
from app.core.llm_client import LLMCallOptions, get_llm_client
client = get_llm_client()
result = await client.invoke(
    messages=prompt,
    use_reasoning=False,
    options=LLMCallOptions(temperature=0, max_tokens=2048, response_format="json"),
    tenant_id=tenant_id,
    user_id=user_id,
)
spec = result.parsed or {<existing fallback dict>}
```
- Use `result.parsed` (the client already extracts JSON) instead of
  `json.loads(response.content)`.
- **Signature change:** `_generate_chart_spec` and `execute` currently lack
  `user_id`. Thread `user_id` from the caller (`chat_service._step_chart`) so
  the audit record is attributed — `ChatService.process_query` already has it.

Defect B — hardcoded model literals in `AgentConfig`. Literals appear at
`chat_service.py:264` (`claude-haiku-4`), `:290` (`claude-opus-4`), `:417`,
`:520`, and `flint_operator.py:68`. Replace each with `settings.LLM_FAST_MODEL`
or `settings.LLM_REASONING_MODEL` (import `from app.core.config import settings`).

**   Gotcha — `AgentConfig.model_name` is decorative.** The actual model is
chosen inside `llm_client.invoke` from the `use_reasoning` flag
(`llm_client.py:127`), so `AgentConfig.model_name` is only a label/metadata
field, never used to select the call. The fix is still correct (don't hardcode
the literal), but note that swapping the string won't change which model is
called unless `invoke` is also bypassed (as in Defect A). Resolving A makes B
moot for correctness; B is still worth doing for grep-cleanliness.

Already correct: `chart_gen_agent.py:81` already references
`settings.LLM_FAST_MODEL` for the (soon-to-be-removed) direct call; and
`narrative_agent.py` / `nl2sql_agent.py` both correctly use `get_llm_client()`.
No change needed there.

Done when: `grep -rn "claude-opus-4\|claude-haiku-4" backend/app/` returns
only matches inside docstrings/comments; the chart-spec LLM call appears in the
audit trail.

Task 4 — Make the golden NL2SQL eval a real gate (was #14)

Why it matters: CLAUDE.md §9 (line 242) — *"Golden NL2SQL test suite must
pass before any merge to main (20 queries, 90% threshold)."* The threshold is
currently unenforced and the test is tautological.

Defect. `test_nl2sql_eval.py:521-558` `test_full_suite_accuracy` calls
`evaluate_case(case, case.expected_sql)` — it scores each golden query's **own
expected SQL** against itself and asserts `accuracy == 100.0`. It cannot detect
a model regression. The machinery to do it properly exists but is unused:
`MOCK_RESPONSES` (`:312`), `_build_mock_responses` (`:315`), and the

`NL2SQLAgent` import (`:26`) are all built then ignored. The 90% threshold
lives only in a `__main__` block (`:624`) that compares expected-to-expected.

Fix — instantiate the agent and mock only the LLM layer:
```python
# In TestEvalSuite.test_full_suite_accuracy:
from app.core import llm_client as llm_mod

class _FakeResult:
    def __init__(self, payload):  # payload = MOCK_RESPONSES[case.id]
        for k, v in payload.items(): setattr(self, k, v)

results = []
for case in golden_cases:
    fake = _FakeResult(MOCK_RESPONSES[case.id])
    with patch.object(llm_mod, "get_llm_client") as mock_get:
        # NL2SQLAgent calls get_llm_client().invoke(...); return fake regardless of args
        mock_get.return_value.invoke = AsyncMock(return_value=fake)
        agent = NL2SQLAgent(AgentConfig(model_name="test"))
        agent_result = await agent.execute(query=case.nl_query, tenant_id="default")
    results.append(evaluate_case(case, agent_result.output.get("sql", "")))

# Enforce the real threshold IN pytest, not just __main__:
suite_accuracy = sum(r.passed for r in results) / len(results) * 100
assert suite_accuracy >= 90.0, f"NL2SQL accuracy {suite_accuracy:.1f}% < 90% threshold"
```
This mocks the **LLM transport** only — the agent's prompt-building,
JSON-parsing, destructive-pattern checking, and the full scoring pipeline all
run for real.

** Gotcha — fix the broken fixture too.** `tests/fixtures/llm_mock.py` has
`import json` at line 247 (last line) but calls `json.dumps` at lines 92-168 →
`UnboundLocalError` on import; `create_mock_llm_client` (`:206-243`) has a
`pass` loop body so it registers no scenarios. The clean approach is to either
(a) fix `llm_mock.py` (move `import json` to top, implement the loop) and use
it here, or (b) delete it and use the inline `AsyncMock` pattern above. Option
(a) is preferred so other agent tests can reuse it.

Done when: `test_full_suite_accuracy` fails if `MOCK_RESPONSES[case.id].parsed.sql`
is mutated to be wrong for any case; the pytest run (not just `__main__`)
enforces the 90% bar; CI's `eval-gate` job
(`ci.yml:180-217`) actually blocks a merge on quality.

Sequencing recommendation

Tasks have no hard ordering dependency, but the lowest-risk high-impact order is:

1. **Task 2** (RLS + JWT) — pure infrastructure/config, no runtime behavior change.
2. **Task 1** (masking + observability) — small, additive, fulfills two guarantees at once.
3. **Task 3** (llm_client routing) — touches the hot path; pair with an integration check.
4. **Task 4** (eval rewrite) — test-only; do last so the new gate can guard the others.

Each task should be its own commit/PR. Run `uv run pytest tests/ -v -m "not e2e"`
after each.

Phase 7 — Build & Test Verification (new)

> **Added:** 2026-08-06. Phases 5b and 6 were implemented and statically verified
> (all files parse, integration points traced by hand), but the two things that
> require a live environment were NOT executed: the test suite and the Docker
> build (including the custom AGE image). These tasks close that loop. They are
> verification/validation work, not new features.

#	Task	Severity	Status
1	Run the backend test suite for real; fix what fails	High	
2	Build the stack via `make setup`; fix the AGE image + any boot failures	High	
3	Run `make verify` green end-to-end on a clean provision	Medium	
4	Add a CI job that builds the custom Postgres image	Medium	
5	Closeout: flip Phase 5b + Phase 6 from UNVERIFIED → verified	Medium	

Task 1 — Run the test suite for real and remediate failures

Why it matters: Phase 6 changed load-bearing code — the golden eval now
invokes `NL2SQLAgent` (was tautological), `ChartGenAgent` routes through
`llm_client`, and the connector gained a tenant GUC. None of it has actually
been executed. "It parses" ≠ "it works."

How to run:
```bash
make setup                                                  # provisions the stack
docker compose -f infra/docker-compose.dev.yml exec backend \
  uv run pytest tests/ -v -m "not e2e" --tb=short
```
Or locally without Docker (faster iteration):
```bash
cd backend && uv sync --dev
uv run pytest tests/ -v -m "not e2e" --tb=short
uv run pytest tests/evals/ -v                               # the rewritten gate
```

Known risks to watch for (traced, not yet confirmed):

1. **pytest-asyncio collection of the new async class methods.**
   `test_nl2sql_eval.py`'s `TestEvalSuite.test_full_suite_accuracy` and
   `test_suite_detects_corrupted_mock` are `async def` methods on a plain
   class. `asyncio_mode = "auto"` (`pyproject.toml`) auto-marks free async
   functions, but **class-based async tests may need an explicit marker or a
   `pytest_asyncio.fixture`** depending on the pytest-asyncio version. If they

```
   show as skipped/errored with "coroutine never awaited", add
   `@pytest.mark.asyncio` to the class or convert to module-level async
   functions. Verify first — don't assume.

2. **The golden eval imports `NL2SQLAgent` which imports `langchain_anthropic`
   at module load** (`nl2sql_agent.py:16`). If `uv sync --dev` didn't install
   it (it's in base deps, should be fine), collection errors. Confirm
   `langchain-anthropic>=0.3` resolves.

3. **`load_prompt("nl2sql-system")` path resolution at test time.**
   `llm_client.load_prompt` computes the path as
   `Path(__file__).parent.parent.parent.parent / ".claude" / "prompts"`
   (`llm_client.py:327`). From `backend/app/core/llm_client.py` that's
   `<repo>/.claude/prompts/` — correct when run from the repo, but inside the
   Docker container (`WORKDIR /app`, i.e. `backend/`) it resolves to
   `/app/.claude/prompts` which **does not exist**. The prompt returns `""` and
   tests still pass (agent builds its own user message), but it's a latent bug
   for runtime. File as a follow-up: make the prompt path configurable via
   `settings` (e.g. `PROMPT_DIR`) with a container-aware default.

4. **CI uses stock `pgvector/pgvector:pg16` — no AGE** (`ci.yml` services
   block). Any test that touches `graph_schema` / AGE will fail in CI even if
   it passes locally with the custom image. Either (a) skip AGE tests in CI
   with a marker, or (b) point CI at the custom image (Task 4). Decide which.

5. **No Redis service in CI.** `cache.py` degrades gracefully (returns None),
   so cache-dependent tests should still pass — but the readiness probe test
   (if one exists) would report Redis unreachable. Confirm no test asserts
   `cache.set` then `cache.get` round-trips without Redis available.

**Scope rule:** this task is *run + fix failures*. If a failure reveals a
design issue (not a bug), write a new Phase 7 task for it rather than
expanding scope here.

**Done when:** `uv run pytest tests/ -v -m "not e2e"` exits 0 locally AND in
CI, with the golden eval (`tests/evals/`) green and `test_suite_detects_corrupted_mock`
proving the gate can fail.

---

### Task 2 — Build the stack via `make setup`; fix the AGE image + boot failures

**Why it matters:** The #1 blocker from the original review was that the
platform could not boot. Phase 5b wrote the provisioning, but none of it has
been built. The custom AGE image is the highest-risk component (compiles AGE
from source against PG16 headers).

**How to run:**
```bash
make setup    # or, to isolate the image build:
docker compose -f infra/docker-compose.dev.yml build postgres
```

Known risks to watch for:

1. **AGE PG16 branch build failures (highest risk).**
   `infra/postgres/Dockerfile` clones `--branch PG16` from `apache/age` and
   runs `make PG_CONFIG=.../pg_config install`. AGE's PG16 support has
   historically lagged behind PG releases and occasionally needs:
   - A specific AGE release tag instead of `HEAD` of the branch (pin a tag
     like `v1.5.0` if `PG16` branch HEAD is broken).
   - The PG server compiled with `--with-llvm` (the pgvector image already
     has LLVM, but verify `llvm-dev`/`clang` are the versions AGE expects).
   - `llvm15` specifically on older Alpine (PG16 bundles LLVM 16; mismatched
     `llvm-dev` causes link errors).
   If the build fails, the fix is usually: pin the AGE tag, and match the
   `llvm-dev`/`clang` major version to what `pg_config --configure` reports.

2. **`CREATE EXTENSION age` requires superuser.**
   Docker's entrypoint makes `POSTGRES_USER` a superuser, so `genbi` can run
   `CREATE EXTENSION age` — but verify the `init.sql` guard's
   `pg_available_extensions` check actually sees AGE after the image build
   (a failed `.so` install would make AGE "available" per the catalog but
   unloadable). If the guard logs "AGE not available" despite the image
   building, the `.so` didn't land in `pkglibdir`.

3. **`gen-env.sh` Fernet dependency.** The script calls
   `python3 -c "from cryptography.fernet import Fernet..."` — if `cryptography`
   isn't installed on the host, it falls back to a raw base64 key, which is
   fine but worth confirming the fallback branch works (I wrote it
   defensively; it hasn't run).

4. **uv/pnpm lockfile generation in `setup.sh`.** The script runs `uv lock`
   and `pnpm install --lockfile-only` if lockfiles are missing, then warns to
   commit them. Confirm the generated `backend/uv.lock` actually satisfies
   `uv sync --frozen` in the Dockerfile (the whole point of Task 3's
   Dockerfile change).

5. **Alembic stamp path.** `setup.sh` runs
   `alembic stamp 0001_baseline` inside the backend container. Verify the
   `alembic.ini` `script_location = app/db/migrations` resolves from
   `WORKDIR /app` — should be fine, but a wrong cwd produces a silent "alembic
   not available" fallback message.

Done when: `make setup` completes, `docker compose ps` shows every service
`(healthy)`, and the AGE graph check in `verify.sh` reports
`genbi_graph` exists (or the graceful "AGE not available" warning if AGE is
disabled).

Task 3 — `make verify` green on a clean provision

Why it matters: `verify.sh` is the acceptance gate for the whole
provisioning effort. It checks all 7 wires (backend liveness/readiness, Cube,
```

Redis, Postgres seed+AGE, frontend, Prometheus). Getting it green proves the platform is actually up, not just that containers started.

How to run: `make verify` (after Task 2).

Known risks: This task is mostly "run it and triage." The most likely failure is the **backend readiness probe returning 503** — it now does real `SELECT 1` + Redis ping. If the backend boots before migrations complete (the `start_period: 30s` healthcheck should cover this, but verify), readiness will flap. Second likely failure: **Cube** — its healthcheck uses `wget` which may not exist in `cubejs/cube:latest`; if so, switch to `curl` or a TCP check.

Done when: `make verify` prints all and exits 0 on a machine that just ran `make setup` from clean.

Task 4 — CI builds the custom Postgres image

Why it matters: CI currently uses `pgvector/pgvector:pg16` as the test DB service (`ci.yml`). After Phase 5b, the project's own schema (`init.sql`) runs `CREATE EXTENSION age` (guarded, so it won't crash — but AGE-dependent code paths are untestable in CI). Also: the custom image itself has no CI coverage, so an AGE upstream breakage ships green.

Deliverable:
- Add a `postgres-image` job to `.github/workflows/ci.yml` that builds `infra/postgres/Dockerfile` and pushes to the registry (or just builds to validate it compiles on every PR).
- Point the `backend-lint-test` job's `services.postgres.image` at the custom image (either the registry tag from the build job, or build it inline via a `docker build` step before the services block — GitHub Actions services can't `build:`, so you need a pre-built tag or a different approach).
- Add a marker like `@pytest.mark.age` to AGE-dependent tests so they run in CI once the image is available, and are skipped otherwise.

** GitHub Actions limitation:** the `services:` block only accepts `image:`, not `build:`. Options: (a) build & push the image in a prior job and reference the tag, (b) use a `docker-compose`-based service setup instead of the `services:` block, or (c) keep CI on stock pgvector and accept that AGE tests are local-only (mark them `@pytest.mark.age` + skip in CI). Option (c) is the lowest-effort honest choice; (a) is the most correct.

Done when: CI builds the AGE image on every PR (failing the build if AGE upstream breaks), and AGE-marked tests either run in CI or are explicitly skipped with a documented reason.

Task 5 — Closeout: flip Phase 5b + Phase 6 from UNVERIFIED → verified

Why it matters: Phase 5b and Phase 6 both carry a `  STATUS: UNVERIFIED` banner at the top. That banner is the source of truth for whether the work in those phases can be relied on. It must not stay UNVERIFIED forever, and it must not be flipped prematurely. This task is the single, explicit gate that moves the status — it is NOT done automatically by Tasks 1–4, because status is a human-confirmed declaration, not a side effect.

Entry criteria (ALL must be met before flipping):
1. Task 1 complete — `uv run pytest tests/ -v -m "not e2e"` and `uv run pytest tests/evals/` both exit 0, locally and in CI.
2. Task 2 complete — `make setup` builds the stack including the custom AGE image; `docker compose ps` shows every service `(healthy)`.
3. Task 3 complete — `make verify` prints all and exits 0 on a clean provision.
4. Task 4 complete (or explicitly deferred with rationale) — CI builds the custom Postgres image, and AGE-marked tests are either running or documented-skipped.

The edit:
- Remove the `  STATUS: UNVERIFIED` blockquote from the Phase 5b header (lines ~103–107) and the Phase 6 header (lines ~413–417).
- Replace each with a `  STATUS: VERIFIED` line citing the date and the Phase 7 task numbers that closed it, e.g.:
  `>  STATUS: VERIFIED 2026-08-XX. Closed by Phase 7 Tasks 1–3.`
- Mark Task 5 in the Phase 7 table.

Anti-scope: if any of Tasks 1–4 surfaced a new bug that was filed as a follow-up task (rather than fixed inline), the status flip may proceed **only if** the follow-ups are non-blocking for the core guarantees those phases claim (RLS enforced, masking wired, eval gate real, stack boots). Blocking follow-ups must be resolved first.

Done when: both phase headers read `  STATUS: VERIFIED` with a date and citation, and Task 5 is .

Sequencing

1. **Task 2 first** (build the image + boot the stack) — because Task 1's test run benefits from a working environment, and Task 3 depends on the stack being up.
2. **Task 1** (run tests, fix failures) — against the now-bootable stack.
3. **Task 3** (verify green) — once tests pass and the stack is stable.
4. **Task 4** (CI image) — hardens the pipeline so regressions are caught before merge, not at the next provision.
5. **Task 5** (closeout) — the final, explicit gate. Run only after 1–4 are (or 4 is documented-deferred). This is what flips Phases 5b and 6 from UNVERIFIED to VERIFIED — nothing else does.

These five tasks convert "implemented and statically checked" into "verified working." Until Task 5 is , treat Phases 5b/6 as **unverified** — the code is written carefully but has never executed.

Phase 8 — Auth & Chat Completion (Tasks 31–37)

> **Status: VERIFIED 2026-08-14** — closes the product's missing login path
> end-to-end and repairs the never-green lint gates.

#	Task	Status
31	Password hashing (bcrypt) + `POST /auth/login` (JWT minting, tenant disambiguation)	
32	Dev bootstrap user seed (`init.sql`: `admin@genbi.local`) + README demo credentials	
33	Auth tests: `security.py` unit tests + DB-backed login integration tests (skip when Postgres down)	
34	Frontend auth slice: `/login` page, `/chat` route, `AuthGuard` redirect, landing page, `LoginForm` extraction	
35	Fix token storage key mismatch (`auth-storage.ts`; api-client previously read the wrong localStorage key)	
36	Repair never-green frontend gates: commit `pnpm-lock.yaml`, ESLint flat config (`eslint .`), shadcn dangling re-export cleanup	
37	Repair backend lint gate (ruff `extend` path + 136 pre-existing violations) + docs refresh + `verify.sh` login check	

Verified by

- `ruff check .` + `ruff format --check .` — clean (backend)
- `pytest -m "not e2e"` — 61 passed, 6 skipped (DB-backed auth tests skip without Postgres; run in CI)
- `pnpm typecheck` / `pnpm lint` / `pnpm build` — clean (standalone trace fails only on local non-symlink filesystems)
- `make verify` now includes an auth login smoke check

Follow-ups (out of scope, flagged)

- **RLS tenant isolation is not enforced**: the app connects as `POSTGRES_USER`
  (a superuser in the official image), so `FORCE ROW LEVEL SECURITY` policies
  never constrain it. Fix: dedicated non-superuser `genbi_app` role + grants +
  connector auth swap. The login request already supports optional `tenant_id`
  for that migration.
- `graph_schema.py` builds AGE Cypher with f-string interpolation — parameterize
  (AGE `$params` jsonb arg) as part of the RLS work.

Phase 8b — Enforced Tenant Isolation (Tasks 38–44)

> **Status: code complete 2026-08-15; runtime verification pending a Docker
> session (`make reset && make setup && make verify`).** Closes the gap where
> RLS was configured but never enforced (the app connected as a superuser).

#	Task	Status
38	Alembic 0002: `genbi_app` + `genbi_auth` roles, grants, `users_login_lookup` policy (mirrored in init.sql)	
39	Alembic 0003: analytics tables under FORCE RLS; seed `users` → `web_users` rename (collision fix)	
40	Runtime switch: `DATABASE_URL` → genbi_app, new `DATABASE_URL_AUTH` engine for login (`get_auth_db`)	
41	Admin scripts repair: `db_admin.py` owner connections + tenant GUC; seed/embed rewrite (both were broken through the read-only connector)	
42	Isolation tests: cross-tenant invisibility, forged-tenant INSERT rejection, no-GUC blindness, genbi_auth carve-out bounds (8 tests)	
43	`verify.sh`: RLS enforcement checks (genbi_app sees 0 users w/o GUC; genbi_auth users-only)	
44	Docs: ADR 006, README roles table + upgrade path	

Verified by

- ruff + full pytest locally (DB-backed tests skip without Postgres; they run
  in CI where the migration step creates the roles)
- CI exercises: migrations (0002+0003) → auth + isolation suites against the
  real roles
- Pending (needs Docker): `make reset && make setup && make verify`,
  login smoke via the genbi_auth path

Notes / follow-ups

- The AGE lineage module (`graph_schema.py`) remains dead code with
  f-string-interpolated Cypher — parameterize before wiring it to the
  runtime role.
- Login rate limiting (protects the genbi_auth cross-tenant read oracle) is
  an app-layer follow-up.
- Writing to tenant tables through the ORM requires setting the GUC on the
  session — copy the `set_tenant_guc` pattern when the first ORM writer
  (audit_log) lands.

Phase 9 — Real Semantic Layer: Cube-Native Catalog (Tasks 45–50)

> **Status: code complete 2026-08-15; live Cube boot pending a Docker
> session (`make reset && make setup && make verify` — as with 8b).**
> ADR 007 documents the pivot: the dbt tier was unrunnable scaffold (no dbt
> toolchain, Cloud-only bridge, invalid YAML) — Cube-native data models are
> now the single source of truth.

#	Task	Status
45	Catalog: `semantic/cube/schema/` — 10 cubes / 23 measures over the seeded tables, joins only on real FKs, tenant_id dimension everywhere	
46	Runtime wiring: cube.js rewrite (real CUBEJS_DB_* driver env, cube_reader role), compose mount `/cube/conf`, node-fetch healthcheck, gen-env.sh	
47	CubeClient fixes: JWT auth (was raw secret), raw-/meta cache + re-parse (was silently dropping parsed metrics), get_agent_context keyword ranking >20 metrics	
48	`GET /metrics/list` real implementation (503 CUBE_UNAVAILABLE); `/metrics/query` explicitly deferred to Phase 10 (needs per-tenant Cube GUC driver)	
49	Tests: catalog structural validation (9), client fixes (7), API list endpoint (3) — all offline	
50	Cleanup/docs: delete `semantic/dbt/`, ADR 007, rewrite semantic README + docs, README refresh	

Verified by

- ruff clean; full suite: **82 passed, 14 skipped** (21 new tests; skips are
  the DB-backed 8b suites that run in CI)
- Catalog tests parse every schema YAML and enforce invariants (unique
  names, symmetric joins, tenant dims, core metrics)
- Fixed latent CI bug found via new tests: error-shape assertions in
  `test_auth.py` expected top-level `code` (FastAPI nests under `detail`)

Deferred to Phase 10 (with /metrics/query + Explore)

- Per-tenant Cube data path: `contextToOrchestratorId` + connection-init
  `set_config('app.current_tenant_id', ...)` in the cube_reader driver
- `retrieve_schema_context` pgvector wiring in NL2SQLAgent (stub returns [])
- `scripts/embed_schema.py` uses an unverified embedding model
  (`claude-embeddings-20250219`) — validate before relying on it

Phase 10 — Metrics + Explore: Tenant-Scoped Cube Data (Tasks 51–56)

> **Status: code complete 2026-08-15; live chain (JWT→Cube→GUC→RLS) verified
> in code/CI mocks, live confirmation pending a Docker session via the new
> `make verify` metric-query check.** ADR 008 documents the design; the
> failure mode is closed (misconfiguration zero rows, never cross-tenant).

#	Task	Status
51	Per-tenant Cube data path: `tenantId` JWT claim → `contextToOrchestratorId` → driverFactory pg `options` GUC (`semantic/cube/cube.js`)	
52	CubeClient: per-tenant token cache, `query(tenant_id=...)`, metric_key no-op fix; first-ever query() tests	
53	`POST /metrics/query` real (catalog validation 400, 503 CUBE_UNAVAILABLE, tenant-scoped, 300s two-tier cache keyed on canonical query)	
54	`GET /datasources` real (cubes from /meta); `/datasources/test` stays a documented stub (different feature)	
55	Explore page: catalog cards, native-select query builder, results table, ChartCard via /charts/render; chat header toggle now navigates to /explore (was dead state)	
56	`verify.sh`: live tenant-scoped metric query check (login → query → expect non-empty rows)	

Verified by

- Backend: 17 new tests (tenant tokens, query() shape/flattening/total,
  API happy/tenant-threading/cache-hit/400/503/422/auth, datasources) —
  full suite green, ruff clean
- Frontend: typecheck + lint clean, build compiles all 7 pages (explore
  included)
- Pending live (Docker): `make verify` metric-query check proves the whole
  JWT→securityContext→per-tenant-driver→GUC→RLS chain end-to-end

Follow-ups

- Safety hardening (EXPLAIN-backed validation, >1M-row confirm,
  tests/services) and the small bundle (login rate limiting, audit_log
  writer, retrieve_schema_context pgvector wiring, embedding-model
  validation) remain queued.

Phase 11 — NL2SQL Context Wiring (Tasks 57–61)

> **Status: code complete 2026-08-15.** Closes the core-loop gap: the NL2SQL
> agent generated SQL with ZERO schema context and ZERO few-shot examples
> (both retrieval stubs returned [] and nothing called them — invisible to
> the golden eval because it mocks the LLM).

#	Task	Status
57	Embedding provider: `app/core/embeddings.py` (OpenAI text-embedding-3-small — 1536 dims match VECTOR(1536); the Anthropic embeddings call targeted a nonexistent API). Config: OPENAI_API_KEY/EMBEDDING_MODEL/EMBEDDING_DIMS + env templates	
58	Retrieval service: `app/services/schema_retrieval.py` — pgvector cosine top-k on schema_embeddings + agent_examples via the RLS-bound connector (tenant GUC); fail-open everywhere	
59	ChatService wiring: schema context (get/set_schema_context, TTL 86400 — previously unused methods) + few-shot (new cache methods, TTL 3600) flow into agent.execute	
60	embed_schema.py: shared embedding provider, cache invalidation after sync, `--examples` golden-set seeding (20 NL/SQL pairs → agent_examples, idempotent); prompt touch-up (dbt → Cube.dev)	
61	Tests: 13 new (provider dims/errors, retrieval contract/tenant-scoping/fail-open, service wiring incl. cache-hit and degradation paths)	

Verified by

- 13 new tests; full suite green; ruff clean
- Ops flow: `PYTHONPATH=backend uv run python scripts/embed_schema.py --examples`
  arms both retrieval sources (needs OPENAI_API_KEY), then invalidates cache

Remaining roadmap

- Governance completion (audit_log writer + login rate limiting)
- Validation hardening (EXPLAIN-backed gate, >1M-row confirm, pipeline tests)
- ivfflat index on agent_examples if the example set grows

Phase 12 — Governance Completion (Tasks 62–66)

> **Status: code complete 2026-08-15.** Closes the README's "audit tracing"
> headline claim (the audit_log table existed but nothing wrote to it) and
> throttles the genbi_auth login oracle.

#	Task	Status
62	Audit writer via the LLMClient callback hook (the documented "every LLM call" contract): `app/services/audit.py` — asyncpg on genbi_app + tenant GUC, parameterized INSERT, non-UUID rows skipped, fail-open; wired in `create_app()`	
63	`_audit()` fix: input_prompt_hash now SHA-256 of the actual prompt (was model+tokens)	
64	Login rate limiting: Redis INCR counter per normalized email (5 failures / 900s → 429 TOO_MANY_ATTEMPTS, reset on success, fail-open without Redis); every 401 path registers	
65	verify.sh: fixed Phase 10 ordering bug (login check ran AFTER the metric check consuming its token — the check always silently skipped); audit-trail info check	
66	Tests: 19 across audit writer (GUC+insert contract, UUID skip, fail-open), prompt-hash fix, limiter (threshold/TTL/reset/fail-open), 429 endpoint paths	

Verified by

- 19 new service tests; ruff clean; full suite green
- CI safety: test_auth accumulates only 3 failed logins on its shared email
  (verified) — under the 5-failure threshold
- Pending live (Docker): audit rows after one chat query; 429 after 5 bad logins

Remaining roadmap (Phase 13 candidate)

- Validation hardening: EXPLAIN-backed SQL gate (connector's explain() is

```
  implemented but unused), >1M-row confirmation, chat-pipeline tests

---

## Phase 13 — Validation Hardening (Tasks 67–70)

> **Status: code complete 2026-08-15.** The final planned phase — completes
> the original roadmap at 70/70 tasks. The SQL safety gate is now
> EXPLAIN-backed and the >1M-row confirmation contract is real end-to-end.

| # | Task | Status |
|---|---|---|
| 67 | EXPLAIN wiring: ValidationAgent `connector` kwarg (protected file — pure strengthening), real row estimates + plan text, fail-open both layers; ChatService passes a short-lived RLS-scoped connector |  |
| 68 | Confirmation contract: `_step_validate` → dict return; gate in sync + stream paths stops before execution until `confirm_large_query: true`; ChatRequest/ChatResponse fields; SSE `validation` payload + `done` status `confirmation_required` |  |
| 69 | Frontend: confirm panel ("Confirm and run" re-sends with the flag), schemas extended; fixed a latent onClick-arg bug in handleSend exposed by the new signature |  |
| 70 | Tests: 13 new — agent+fake connector (estimates flow, >1M flags, EXPLAIN error fails open, destructive SQL never EXPLAINed) + full pipeline orchestration (sync happy/no_sql/validation_failed/confirm-block-then-proceed; stream event order/confirmation_required/confirmed-resend) |  |

### Verified by

- 133 passed / 14 skipped (13 new); ruff clean; frontend typecheck/lint clean
- Orchestration tests run the REAL deterministic ValidationAgent with only
  LLM agents mocked — the confirmation gate is exercised end-to-end
- Doc debt fixed: agent-system.md stale checks table (is_valid key,
  timeout-injection claim, pattern counts), openapi.yaml fields, CLAUDE.md
  protected-file paths (validation agent + dbt-era metrics path)

### Roadmap status: COMPLETE (70/70)

Future candidates (new roadmap): conversation persistence + multi-turn
chat (conversations table still unwritten), reports/dashboards build-out,
AGE lineage wiring, feedback API (feedback_score column), per-role
validation permissions (user_roles plumbing exists, unused).

---

## Phase 14 — Conversations & Multi-Turn Chat (Tasks 71–75)

> **Status: code complete 2026-08-15.** The chat product finally has memory:
> conversations persist, history feeds the NL2SQL prompt, and the UI lists
> and resumes threads.

| # | Task | Status |
|---|---|---|
| 71 | Alembic 0004_messages: messages table (RLS FORCE + tenant_isolation + genbi_app grants) mirrored in init.sql; ORM Message model |  |
| 72 | `app/services/conversations.py`: create/list/append via asyncpg on genbi_app + tenant GUC (Phase 12 pattern); per-user scoping predicate; fail-open writes, raise-on-read |  |
| 73 | `GET /conversations` + `GET /conversations/{id}/messages` (JWT-scoped; 400/503 paths); SSE `start` event now carries conversation_id |  |
| 74 | History injection: ChatService resolves/creates the conversation, loads last 6 turns into the NL2SQL prompt ("## Conversation History" section), persists user + assistant turns at every pipeline exit (both modes) |  |
| 75 | Frontend: conversations sidebar (list, new-chat, active highlight), replay-on-select, send carries conversation_id; 18 new backend tests (service contract, endpoints, agent prompt section, wiring incl. fail-open paths) |  |

### Verified by

- 151 passed / 14 skipped (18 new); ruff clean; frontend typecheck/lint clean
- Pipeline tests extended to mock conversation persistence (fail-open held —
  the earlier unmocked run passed, just slowly)
- Live behavior (Docker session): send two follow-up questions in one
  conversation; sidebar lists/resumes threads

### Follow-ups

- Message pagination UI (limit param exists), conversation rename/delete,
- reports/dashboards build-out; feedback API; AGE wiring (unchanged queue)

---

## Phase 15 — Governance Bundle (Tasks 76–78)

> **Status: code complete 2026-08-15.** Three quick governance wins: feedback
> loop, role-based query policy, and AGE lineage parameterization.

| # | Task | Status |
|---|---|---|
| 76 | Feedback API: `POST /chat/feedback` (session_id + score −1/0/+1) → `audit_log.feedback_score` via the GUC writer pattern (updates all rows of the session); 503 FEEDBACK_UNAVAILABLE on failure |  |
| 77 | Per-role validation: ValidationAgent `_check_role_restrictions` — QUERY_ROLES gate (admin/analyst/user/viewer; unrecognized roles hard-rejected), viewer role restricted to single-table lookups (JOIN/UNION/CTE rejected); empty roles unrestricted (connector + RLS remain the hard gates) |  |
| 78 | AGE lineage parameterized: all cypher in graph_schema.py converted from f-string interpolation to inline literals + AGE `$name` parameters bound via the third `ag_catalog.cypher` argument; no behavior change (module still has zero callers — wiring is future work) |  |

### Verified by

- 160 passed / 14 skipped (9 new tests: role matrix, feedback happy/503/422/auth);
  ruff clean; frontend untouched
- Live (Docker session): thumbs-up a chat response then check
  `SELECT feedback_score FROM audit_log`; AGE statements exercise on `make
  reset` (init graph path)

### Follow-ups

- Frontend thumbs up/down buttons on assistant messages (writes via the new
  endpoint) — small UI addition, deferred
- AGE sync scheduler (`sync_semantic_layer_to_graph` still has no caller)

---

## Phase 16 — Reports & Dashboards: The Capstone (Tasks 79–84)

> **Status: code complete 2026-08-15.** The last original-scaffold stub is
> real: multi-chart reports from natural language, persisted and browsable.
```

```
| # | Task | Status |
|---|---|---|
| 79 | Alembic 0005_reports: reports + report_sections tables (RLS FORCE + tenant_isolation + genbi_app grants + indexes), init.sql mirror, ORM models
 |   |
| 80 | Report service (`app/services/reports.py`): LLM-planned pipeline (2 LLM calls/report — planner over the metric catalog at temp 0 + one summary
call), deterministic per-section execution (tenant-scoped Cube query → ChartAssemblyInput → FlintChartBridge SVG), per-section fail-open; persistence
via the conversations GUC-writer pattern (writes fail-open, reads raise) |   |
| 81 | API: POST /reports/generate (full report incl. sections), GET /reports (list, declared before /{id}), GET /reports/{report_id} (400/404/503 pat
hs) |   |
| 82 | New prompt `.claude/prompts/report-planner-system.md` (JSON contract: title + sections with exact catalog metric names, dimension/granularity r
ules) |   |
| 83 | Frontend /reports page: prompt box + section-count select + Generate; sidebar list of past reports; report rendering with per-section ChartCard
s (reusing the chart stack, no new deps); chat header FileText nav |   |
| 84 | Tests: 21 new (12 service — planner sanitization, deterministic chart specs, happy path with tenant-threading + total computation, partial fail
ure, persistence-failure warning, GUC+insert contract, fail-open; 9 API — generate/list/get matrix) |   |

### Verified by

- 181 passed / 14 skipped (21 new); ruff clean; frontend typecheck/lint clean;
  build compiles all 8 pages (reports included); chain verified → 0005_reports
- Live (Docker session): generate "Q3 revenue and pipeline report", view it
  from the list, re-open it after reload (persistence)

### Follow-ups

- `/dashboards` frontend dir still empty (report list is the surface for now)
- Scheduled/regenerating reports, PDF export, dashboard pinning, AGE
  DASHBOARD_USES lineage wiring — unchanged queue
```

uv.lock

4090 lines · 757.6 KB

```
version = 1
revision = 3
requires-python = ">=3.12"
resolution-markers = [
    "python_full_version >= '3.15' and sys_platform == 'win32'",
    "python_full_version >= '3.15' and sys_platform == 'emscripten'",
    "python_full_version >= '3.15' and sys_platform != 'emscripten' and sys_platform != 'win32'",
    "python_full_version == '3.14.*' and sys_platform == 'win32'",
    "python_full_version == '3.14.*' and sys_platform == 'emscripten'",
    "python_full_version == '3.14.*' and sys_platform != 'emscripten' and sys_platform != 'win32'",
    "python_full_version < '3.14' and sys_platform == 'win32'",
    "python_full_version < '3.14' and sys_platform == 'emscripten'",
    "python_full_version < '3.14' and sys_platform != 'emscripten' and sys_platform != 'win32'",
]

[manifest]
members = [
    "genbi",
    "genbi-backend",
]

[[package]]
name = "aiohappyeyeballs"
version = "2.7.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/ce/f4/eec0465c2f67b2664688d0240b3212d5196fd89e741df67ddb81f8d35658/aiohappyeyeballs-2.7.1.tar
.gz", hash = "sha256:065665c041c42a5938ed220bdcd7230f22527fbec085e1853d2402c8a3615d9d", size = 24757, upload-time = "2026-07-01T17:11:55.501Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/71/43/1947f06babed6b3f1d7f38b0c767f52df66bfb2bc10b468c4a7de9eceff2/aiohappyeyeballs-2.7.1-py3-non
e-any.whl", hash = "sha256:9243213661e29250eb41368e5daa826fc017156c3b8a11440826b2e3ed376472", size = 15038, upload-time = "2026-07-01T17:11:54.055Z" }
,
]

[[package]]
name = "aiohttp"
version = "3.14.3"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "aiohappyeyeballs" },
    { name = "aiosignal" },
    { name = "attrs" },
    { name = "frozenlist" },
    { name = "multidict" },
    { name = "propcache" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
    { name = "yarl" },
]
sdist = { url = "https://files.pythonhosted.org/packages/58/d9/22ce5786ac0c1653ae8b6c23bded02c1686d11f0dbb45b31ce128e0df985/aiohttp-3.14.3.tar.gz", ha
sh = "sha256:9491196535a88924a60afd5b5f434b5b203b6cc616250878dbdb223a8f7844bc", size = 7971213, upload-time = "2026-07-23T01:57:27.037Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/18/d4/eb96299230e20acf2efae207cb8d69051f1f68e357e5ea5e479bf6fb097a/aiohttp-3.14.3-cp312-cp312-mac
osx_10_13_universal2.whl", hash = "sha256:39aded8c7f3b935b54aab1d8d73c70ec0ee2d3ec3b943e0e86611bc150ba47f5", size = 754690, upload-time = "2026-07-23T
01:53:47.332Z" },
    { url = "https://files.pythonhosted.org/packages/88/11/e7a70a209eb9a067c0d3212b518a0134e3484f5178c7533878b6b514d469/aiohttp-3.14.3-cp312-cp312-mac
osx_10_13_x86_64.whl", hash = "sha256:5bcb6ff3fdab1258a192679ff1a05d44f59626430aa05cd1a9d2447423599228", size = 509484, upload-time = "2026-07-23T01:5
3:51.159Z" },
    { url = "https://files.pythonhosted.org/packages/30/07/4bbc222cc8dbe31d4c3e8a5baad2286e4d42026ac0c570027b89afce6344/aiohttp-3.14.3-cp312-cp312-mac
osx_11_0_arm64.whl", hash = "sha256:617105e2c3018ee38d0c8ce5ee3c84f621a6d8b9f723202aacaff28449ca91ee", size = 511949, upload-time = "2026-07-23T01:53:
55.083Z" },
    { url = "https://files.pythonhosted.org/packages/54/b9/42e74c46b7b7c794b995bbc1f573fb48950c38b19d8600c62a6804ee2d67/aiohttp-3.14.3-cp312-cp312-man
ylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:f631fe87a6f30df5fbe6d79640b25e4cffb38c31c7fb6f10871517b84b0f8c1a
", size = 1765282, upload-time = "2026-07-23T01:53:59.662Z" },
    { url = "https://files.pythonhosted.org/packages/6b/ed/62bc4d74363ad346d518e0720363a949f63e2e23439a79eb5813d4d29bb3/aiohttp-3.14.3-cp312-cp312-man
ylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:a94dbaae5ae27bd849c93570669bff91e0510f33a80805738e3de72a7be0447b",
size = 1741511, upload-time = "2026-07-23T01:54:04.063Z" },
    { url = "https://files.pythonhosted.org/packages/d0/9f/181e8a8bc79e47d13c7fc4540bd7a3b729d9505609c61f392a8dd2fbfe55/aiohttp-3.14.3-cp312-cp312-man
ylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:8f2f1c4c032c7cedd7d8da6f54c97b70266c6570c3108d3fdffee7188bb70529
", size = 1810680, upload-time = "2026-07-23T01:54:09.882Z" },
    { url = "https://files.pythonhosted.org/packages/5c/9a/dec94d6ad694552fe3424e3f1928d7a606a5d9d9433a04e7ecdd9d38ae7f/aiohttp-3.14.3-cp312-cp312-man
ylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:ea05e1f97ceea523942d9b2a7d7c0359d781d683d6b043f5943a602b14da4787", siz
e = 1905646, upload-time = "2026-07-23T01:54:13.475Z" },
    { url = "https://files.pythonhosted.org/packages/52/b7/7cd31f29d6055bd711ae6e669367fba6f5ae9de463910a793e30556a8db7/aiohttp-3.14.3-cp312-cp312-man
ylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:543906c127fb1d929b95076db19b83fa2d46751006ff1e23b093aa5ac4d8db42",
size = 1792122, upload-time = "2026-07-23T01:54:15.752Z" },
    { url = "https://files.pythonhosted.org/packages/66/73/10b1ef93afa61f4963c746257b70ced619cf31a4798671de5fdb2608501d/aiohttp-3.14.3-cp312-cp312-man
ylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:0a5ff2dfbb9ce645fa5b8ef3e02c6c0b9cc3f6030ff863d0c51fffc50cb5541b", size = 1591127, upl
oad-time = "2026-07-23T01:54:19.489Z" },
    { url = "https://files.pythonhosted.org/packages/49/ed/3b203fa6de1b338c14acdc06bf6ca9b043b7944f005966958c2ced932cde/aiohttp-3.14.3-cp312-cp312-mus
llinux_1_2_aarch64.whl", hash = "sha256:041badb8f84396357c4d3ad26de6afd7a32b112f43d3c63045c0c8278cfd2043", size = 1725210, upload-time = "2026-07-23T0
1:54:24.129Z" },
    { url = "https://files.pythonhosted.org/packages/28/b7/1c2aab8c706436dcc28598452488ac9cd7c409da815237c28c27d58993e6/aiohttp-3.14.3-cp312-cp312-mus
llinux_1_2_armv7l.whl", hash = "sha256:530125ee1163c4219af35dc3aa1206e541e7b31b6efc1a3f93b70a136f65d427", size = 1764848, upload-time = "2026-07-23T01
:54:27.973Z" },
    { url = "https://files.pythonhosted.org/packages/54/50/94c28f08b131c4bf10984ea2c7a536c9920608bb2d6e7f95642c30cc87b7/aiohttp-3.14.3-cp312-cp312-mus
llinux_1_2_ppc64le.whl", hash = "sha256:c8653fd547c93a61aadc612007790f5555cdd18946fa48cf45e26d8ea4ea473d", size = 1777102, upload-time = "2026-07-23T0
1:54:31.775Z" },
    { url = "https://files.pythonhosted.org/packages/13/d4/e7d09ba7d345fb2d74440fd2fa033c5e079fac05552927705986f41a364f/aiohttp-3.14.3-cp312-cp312-mus
llinux_1_2_riscv64.whl", hash = "sha256:89176250f686cb9853c0fb7ead90e639e915b84a6f43eedc2a4e7ec21f1037f0", size = 1580205, upload-time = "2026-07-23T0
1:54:34.518Z" },
    { url = "https://files.pythonhosted.org/packages/a3/84/072a91d68e1e1eb587985b54baab94221277f877e8ef274fc213a0ceae28/aiohttp-3.14.3-cp312-cp312-mus
llinux_1_2_s390x.whl", hash = "sha256:3a26434dafe408229ff3403458ca58de24fb51936504decac49ce6755f77e59d", size = 1797219, upload-time = "2026-07-23T01:
54:36.995Z" },
    { url = "https://files.pythonhosted.org/packages/e0/eb/aad34e897e668424d6e995da5dff8a4a09af93363d3392488772957a63aa/aiohttp-3.14.3-cp312-cp312-mus
llinux_1_2_x86_64.whl", hash = "sha256:d1558173930a5a8d3069cee5c92fc91c87c4dbcb099debbb3622053717145a19", size = 1768629, upload-time = "2026-07-23T01
:54:40.103Z" },
    { url = "https://files.pythonhosted.org/packages/b6/2b/6bb88ddba0fecd9122aa3ebcad25996cf6c083a4a7040dbb3a4f97972af6/aiohttp-3.14.3-cp312-cp312-win
32.whl", hash = "sha256:16100ad3ab8d649fdfbee87602d9d2dcdca9df0b9eda8a1b5fdc0d41f96da559", size = 451481, upload-time = "2026-07-23T01:54:42.547Z" },
    { url = "https://files.pythonhosted.org/packages/76/9b/f2f8f108da17ecef2cc3efc424e8b7ad3782b1a8360f7b8eae8ced84f6ea/aiohttp-3.14.3-cp312-cp312-win
_amd64.whl", hash = "sha256:33a2d7c28d33797a2e99923dffa63f83d908a19b6bf26cfe80fa790aa5e1a75a", size = 476845, upload-time = "2026-07-23T01:54:44.853Z"
 },
```

```
    { url = "https://files.pythonhosted.org/packages/3e/44/28dac80a8941b604f4da10ce21097614ca1bf905ce93dca28d8d7de9c1e7/aiohttp-3.14.3-cp312-cp312-win_arm64.whl", hash = "sha256:362a3fd481769cac1a824514bcd86fda51c65e8fe6e051099e008fddde6db17c", size = 448050, upload-time = "2026-07-23T01:54:47.087Z" },
    { url = "https://files.pythonhosted.org/packages/57/be/5afd201cc0ab139029aadb75392efe85a293403d9dd3a3226161c21ce00c/aiohttp-3.14.3-cp313-cp313-android_21_arm64_v8a.whl", hash = "sha256:2e9878ae68e4a5f1c0abe4dd497dbc3d51946f5837b56759e2a02e78fa90ef86", size = 506269, upload-time = "2026-07-23T01:54:49.075Z" },
    { url = "https://files.pythonhosted.org/packages/22/09/dec8189d62b45ade009f6792a2264b942a90cb88aeaf181239933cd72c3c/aiohttp-3.14.3-cp313-cp313-android_21_x86_64.whl", hash = "sha256:f3d2669fe7dec7fc359ecdb5984b29b50d85d5d00f8c1cb61de4f4a24ee42627", size = 515166, upload-time = "2026-07-23T01:54:51.894Z" },
    { url = "https://files.pythonhosted.org/packages/28/24/2854869d29ed8a8b19d74f9ec6629515f7e04d02dd329d9d179201e58e47/aiohttp-3.14.3-cp313-cp313-ios_13_0_arm64_iphoneos.whl", hash = "sha256:cc7cb243a68167172f48c1fd43cee91ec4b1d40cefd190edd43369d1a6bc9c82", size = 486263, upload-time = "2026-07-23T01:54:54.223Z" },
    { url = "https://files.pythonhosted.org/packages/d4/dd/57187c8be2a35aea65eaee3bd2c3dcbbcf0204f5106c89637e3610380cd1/aiohttp-3.14.3-cp313-cp313-ios_13_0_arm64_iphonesimulator.whl", hash = "sha256:78253b573e6ffab5028924fc98bc281aae05445969982a10864bc360dea2016c", size = 492299, upload-time = "2026-07-23T01:54:56.236Z" },
    { url = "https://files.pythonhosted.org/packages/b9/11/06ae6ed8f0d414edf4068861e233d8fe23ee699bfd4b3ceb8663db948a62/aiohttp-3.14.3-cp313-cp313-ios_13_0_x86_64_iphonesimulator.whl", hash = "sha256:7041d52c3a7fa20c9e8c182b534704abb19502c8bdcbde7ab23bfda6f642394f", size = 502235, upload-time = "2026-07-23T01:54:58.377Z" },
    { url = "https://files.pythonhosted.org/packages/7e/a3/559639c34a345d2cf7c52dff6838119f2eaf29eb508227b5b83f573af813/aiohttp-3.14.3-cp313-cp313-macosx_10_13_universal2.whl", hash = "sha256:ac74facc01463f138b0da5580329cfcc82818dea5656e83ddcd11268fc12ff80", size = 750883, upload-time = "2026-07-23T01:55:00.65Z" },
    { url = "https://files.pythonhosted.org/packages/91/cd/41e131f13afd1e7b0172a9d9eda085ef90eb8439f41f0d279db81ed3ae60/aiohttp-3.14.3-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:d6218d92e450824e9b4881f44e8c09f1853b490f9a64130801024a4793b1b3b0", size = 508473, upload-time = "2026-07-23T01:55:02.945Z" },
    { url = "https://files.pythonhosted.org/packages/bc/6b/e7f13410d391c6e55b4c007a8de024355389d7d459e3d64c42b2d33617e5/aiohttp-3.14.3-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:11fb37ef075669eee52ab1928fbf6e1741fada40409fa309ebde9607a962aebf", size = 509190, upload-time = "2026-07-23T01:55:05.173Z" },
    { url = "https://files.pythonhosted.org/packages/97/21/6464573e53d69672cc1eada3e5c5cb2d2efa82701e8305a0f2047a576967/aiohttp-3.14.3-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:55bdcc472aafe2de4a253045cc128007a64f1e0264fb675791e132ea5edaa3bd", size = 1761478, upload-time = "2026-07-23T01:55:07.383Z" },
    { url = "https://files.pythonhosted.org/packages/1a/81/d217043a4c17fbce360905e3b2bdd20139ebc9a2de836d035d179c4da006/aiohttp-3.14.3-cp313-cp313-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:c39846c3aad97a8530c89d7a3869a8f8e9e3762c6ac0504481e5c80948f7e807", size = 1735092, upload-time = "2026-07-23T01:55:09.803Z" },
    { url = "https://files.pythonhosted.org/packages/a1/66/e13a02d0eeb1a9a502402a977abb4e4abff9fe4051c26f80558c57a7c975/aiohttp-3.14.3-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:5895ef58c4620afe02fa16044f023dc4dafec08158f9d08874a46a7dbc0341b8", size = 1800546, upload-time = "2026-07-23T01:55:12.012Z" },
    { url = "https://files.pythonhosted.org/packages/26/5e/57d42fca1d18cb5acc1cad945d017fabc5d6ae71d8a08ad66be8dc3ee544/aiohttp-3.14.3-cp313-cp313-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:fa9467a8113aa69d3d7c55a70ef0b7c636010a40993f3df9d9d0d73b3eb7ef24", size = 1895250, upload-time = "2026-07-23T01:55:14.357Z" },
    { url = "https://files.pythonhosted.org/packages/ca/1c/7da8d08e74d56f00070822f9638ff3f1c563f8ad87d1efa996c87bfc8644/aiohttp-3.14.3-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:d7d2deec16eeedf55f2c7cf75b521ea3856a5177e123844f8fd0f114ce252cb5", size = 1789289, upload-time = "2026-07-23T01:55:16.668Z" },
    { url = "https://files.pythonhosted.org/packages/cd/0f/cf16bcf56896981c1a0319f5d5db9337994b5165730c48a8fa07e9b34be6/aiohttp-3.14.3-cp313-cp313-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:dd54d0e8717de95939766febac482ac0474d8ac3b048115f9f2b1d23a16e7db4", size = 1586706, upload-time = "2026-07-23T01:55:18.913Z" },
    { url = "https://files.pythonhosted.org/packages/fe/6f/76eac12a7f2480e1e304f842efdb07db33256b0d9165b866b6ef0806c202/aiohttp-3.14.3-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:df82f3787c940c94986b34222d59c9e38843fba85139f36e85255a82ad5355a9", size = 1724652, upload-time = "2026-07-23T01:55:21.296Z" },
    { url = "https://files.pythonhosted.org/packages/39/b6/19c8c592baeeb94b75f966547d40c02ac7590902306ec5863d5c027cf506/aiohttp-3.14.3-cp313-cp313-musllinux_1_2_armv7l.whl", hash = "sha256:42a67efc36300d052fb4508a53e8b6901b9284b599ae63945c377569c5fcc1e1", size = 1756239, upload-time = "2026-07-23T01:55:23.705Z" },
    { url = "https://files.pythonhosted.org/packages/dc/c9/4e9383150296f97f873b680c4de8fb2cd88608fb9f48c79edcb111611abc/aiohttp-3.14.3-cp313-cp313-musllinux_1_2_ppc64le.whl", hash = "sha256:7a75aa63cbf9b21cfaf60dc2657e19df2c2867d91707d653fee171ffeedd1371", size = 1769161, upload-time = "2026-07-23T01:55:26.082Z" },
    { url = "https://files.pythonhosted.org/packages/aa/1e/147bdc6cc5de5f3ab011be8bf5d6e786633249f22c20bae06f85e45f5387/aiohttp-3.14.3-cp313-cp313-musllinux_1_2_riscv64.whl", hash = "sha256:e92eb8acc45eb6a9f4935071a77edf5b85cc6f8dfad5cd99e97653c26593cdde", size = 1578759, upload-time = "2026-07-23T01:55:28.846Z" },
    { url = "https://files.pythonhosted.org/packages/fd/31/78388a9d6040ece2e11df62ea229a822cf5e52d238374b220ae9975b2623/aiohttp-3.14.3-cp313-cp313-musllinux_1_2_s390x.whl", hash = "sha256:b014a6ed7cf912e787149fdc529166d3ceabac23f26efeea3158c9aba2354e7e", size = 1792025, upload-time = "2026-07-23T01:55:31.457Z" },
    { url = "https://files.pythonhosted.org/packages/03/51/a3d29fdf2c25d796746af8ad6fe56a45d6256c38b0a8a2ed752e1160b3a2/aiohttp-3.14.3-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:3d4f72af88ac2474bb5bca640030320e3d38a0163a1d7533500e87be458eef71", size = 1768477, upload-time = "2026-07-23T01:55:33.87Z" },
    { url = "https://files.pythonhosted.org/packages/29/a6/442e18b5afeade534d877a2dc3c3e392aff8d49787890b0cf84790410267/aiohttp-3.14.3-cp313-cp313-win32.whl", hash = "sha256:5f08ec777f35ee70720233b8b9811d3bb5d728137f30ac91b7457709c3261ac0", size = 451069, upload-time = "2026-07-23T01:55:36.121Z" },
    { url = "https://files.pythonhosted.org/packages/9d/69/3d876ac02659f271cf7f6769f14a8e3de5b6e888ed8b5a7e998086a4cec8/aiohttp-3.14.3-cp313-cp313-win_amd64.whl", hash = "sha256:dff9461ec275f22135650d5ba4b4931a11f3958df7dfbb8db630000d4dee0883", size = 476518, upload-time = "2026-07-23T01:55:38.303Z" },
    { url = "https://files.pythonhosted.org/packages/b2/0e/50d6e6471cd31edce8b282bdec59375a3a69124d8a989a0b1313355cae52/aiohttp-3.14.3-cp313-cp313-win_arm64.whl", hash = "sha256:ddcac3c6b382e81f1dd0499199d4136b877beb4cb5ef770bbbfba56c4b8f55d2", size = 447676, upload-time = "2026-07-23T01:55:40.451Z" },
    { url = "https://files.pythonhosted.org/packages/c8/20/887fdcf832326571b370ffc347b3e70abe101096f3720126aac161b1d872/aiohttp-3.14.3-cp314-cp314-android_24_arm64_v8a.whl", hash = "sha256:49f7325beb0f85ef4aef5f48f490269575f83e6e2acad00a1d80b807eb027062", size = 509067, upload-time = "2026-07-23T01:55:42.618Z" },
    { url = "https://files.pythonhosted.org/packages/ad/a3/92cec936f78cc4bf0fa5554ebe593b73459d94e3c62303e1902a4cccb6f7/aiohttp-3.14.3-cp314-cp314-android_24_x86_64.whl", hash = "sha256:e3be98a7c30b8c25d573dafba7171d66dfb05ee6a9070fc46535464ff97700a6", size = 514774, upload-time = "2026-07-23T01:55:44.937Z" },
    { url = "https://files.pythonhosted.org/packages/29/ba/2a0c38df3fc557620b6a5acd98364af050053b6285b4dc7ee74100c63c18/aiohttp-3.14.3-cp314-cp314-ios_13_0_arm64_iphoneos.whl", hash = "sha256:614c61d478b83953e261d02bb2df750f17227cd33ef8002945bf5aebbde21919", size = 488134, upload-time = "2026-07-23T01:55:47.135Z" },
    { url = "https://files.pythonhosted.org/packages/48/d6/d51b7d4bf309af3693940d8ffd2b9ed0b682434ef85959b7c9c137f60cf8/aiohttp-3.14.3-cp314-cp314-ios_13_0_arm64_iphonesimulator.whl", hash = "sha256:1caa7b0d05f3e3a36f87788c59e970a7ee1cefcfcbb924a9f138c4a6551c9cb7", size = 494201, upload-time = "2026-07-23T01:55:49.451Z" },
    { url = "https://files.pythonhosted.org/packages/3f/5a/8f624384e5f1efabb5229b94157eb966b021e97bdb188c62860c2ae243c2/aiohttp-3.14.3-cp314-cp314-ios_13_0_x86_64_iphonesimulator.whl", hash = "sha256:dfa68deb2a443bdaa3ea5297b0699c1464f08aef3812b486d1348eee61b07dc0", size = 502766, upload-time = "2026-07-23T01:55:51.656Z" },
    { url = "https://files.pythonhosted.org/packages/a6/26/4ff0164370deec18fb19254ee4ab10b7a73304ac0c860b13f5f84663759b/aiohttp-3.14.3-cp314-cp314-macosx_10_15_universal2.whl", hash = "sha256:e72ee89e28d907a18f46959b4eb0bb06701cc7f8cf4366e00029e2ccfaaf5924", size = 756557, upload-time = "2026-07-23T01:55:53.964Z" },
    { url = "https://files.pythonhosted.org/packages/97/a3/7056b86dc0d9ec709ea9777eae3b0161428f943372f8b98c01c11593b682/aiohttp-3.14.3-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:ad4c8b7488d745d2ca4838ebd8ae5ba9b56341d30b1da43640e4ce87f9f49646", size = 510168, upload-time = "2026-07-23T01:55:56.22Z" },
    { url = "https://files.pythonhosted.org/packages/85/ed/0357a015892fd68058bf2d39d3fd1958e459b997a7db30aaa6aaa434ae96/aiohttp-3.14.3-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:db332af25642007330fca8be5c4d194caf2bea7a7fc84415aff3497af5dfee6b", size = 512957, upload-time = "2026-07-23T01:55:58.437Z" },
    { url = "https://files.pythonhosted.org/packages/47/d1/8aba53f15ccb2238405f5e9d30e2a8ca44f93878c26e7165ade00d374b1c/aiohttp-3.14.3-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:25bd2708db6bdf6a6630dd37bdcdfcb47c4434d22ac69c64665b802910140b30", size = 1750149, upload-time = "2026-07-23T01:56:00.856Z" },
    { url = "https://files.pythonhosted.org/packages/49/bd/40c3fee32752928437 5c6701cbb0fa4600cc2e8432af1378f897e2ef7d3a/aiohttp-3.14.3-cp314-cp314-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:cef89a58e628c4efcac3275c2d68083f82426dcdc89c1492a6f654f9f7ea6ab9", size = 1707685, upload-time = "2026-07-23T01:56:03.371Z" },
    { url = "https://files.pythonhosted.org/packages/2a/a3/ca0cc6724cca8114b05694abd916060758c79894c3aa5b012cdadc1bc28e/aiohttp-3.14.3-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:c23ec8ee9d5ab2f5421f9c7fffce208435607af27fd46d4a44e031954352838f", size = 1803911, upload-time = "2026-07-23T01:56:05.817Z" },
    { url = "https://files.pythonhosted.org/packages/95/b5/85b099c299c3ffd38ad9b3e43694c8a346934e4a30c88c4fd5a841234f77/aiohttp-3.14.3-cp314-cp314-man
```

```
ylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:e2667f0bbe7eb6c74eae5e9691441ad186e5845ca3cff63230fc09c4e7514f5d", siz
e = 1876929, upload-time = "2026-07-23T01:56:08.413Z" },
    { url = "https://files.pythonhosted.org/packages/d5/b7/1da684a04175473fa4cddbf9a2f572e79514c3fd27a74597f43057d4f3da/aiohttp-3.14.3-cp314-cp314-man
ylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:18cb43369747b2ae007bd2655fb8e63a099c2ff1d207962943636dac989b3147",
size = 1761112, upload-time = "2026-07-23T01:56:10.918Z" },
    { url = "https://files.pythonhosted.org/packages/d1/16/bc4b55e3e5cb175fd69c53c90d60d2f47797cb343da5106e23863dc4dba4/aiohttp-3.14.3-cp314-cp314-man
ylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:d77640cc618c1d99fc4f8589c0f24a730adfa54eb1e57ef7bf0c8dfb78da898c", size = 1583500, upl
oad-time = "2026-07-23T01:56:13.613Z" },
    { url = "https://files.pythonhosted.org/packages/2a/e8/13a9d957a1ee40837f46aa30f0f4c657e673ad86a2e6362a9f9be20d26d9/aiohttp-3.14.3-cp314-cp314-mus
llinux_1_2_aarch64.whl", hash = "sha256:53e5179d8abb5710f8e83ba207c41c8d1261fcffd4616500e15ca2b7a33be10a", size = 1713940, upload-time = "2026-07-23T0
1:56:15.969Z" },
    { url = "https://files.pythonhosted.org/packages/38/05/d33c680c1bcf1c7e130f9cbfc1fc02fe8bb0c4af2a94a53dd5fb56131e5c/aiohttp-3.14.3-cp314-cp314-mus
llinux_1_2_armv7l.whl", hash = "sha256:cd817772b2fcf2b8c0905795318485f9ec16eae60b29feb7f4c77085311637f0", size = 1724413, upload-time = "2026-07-23T01
:56:18.591Z" },
    { url = "https://files.pythonhosted.org/packages/85/1d/af798d306f7a74b6a632dbcabcf62a4c91391b7582d2a8c6d7712e2cc54e/aiohttp-3.14.3-cp314-cp314-mus
llinux_1_2_ppc64le.whl", hash = "sha256:4e3ac92d90e92773b2362d506068e9a948192bd553e743c5b2429e28527c8661", size = 1770748, upload-time = "2026-07-23T0
1:56:21.074Z" },
    { url = "https://files.pythonhosted.org/packages/a8/92/ad720d472556a995049206867765e9410969684f86ee09423ff9969044c1/aiohttp-3.14.3-cp314-cp314-mus
llinux_1_2_riscv64.whl", hash = "sha256:3f42e9b78301f11c8f861746175d8b9c1ccef713fcad9eab396e2f6db8ed4a22", size = 1577564, upload-time = "2026-07-23T0
1:56:23.475Z" },
    { url = "https://files.pythonhosted.org/packages/60/ad/0ed7586cbef7a884e23a752fa2bb987a122e6a5dd50dab109258d0a95193/aiohttp-3.14.3-cp314-cp314-mus
llinux_1_2_s390x.whl", hash = "sha256:9d9edccfe496b476db5f398d97b865e9a6752bcf8aec4eef8390ce20fb64bb41", size = 1782080, upload-time = "2026-07-23T01:
56:25.994Z" },
    { url = "https://files.pythonhosted.org/packages/97/ea/dbaed0d73e8a69aad653b045dab451c67c2454bb731a37b45a86593e9422/aiohttp-3.14.3-cp314-cp314-mus
llinux_1_2_x86_64.whl", hash = "sha256:1c5ec8fb1bcc31a8466f74aaf26c345d5c386fa4bd08a3f0eb9c7a4a3fe8b5bf", size = 1745813, upload-time = "2026-07-23T01
:56:28.604Z" },
    { url = "https://files.pythonhosted.org/packages/81/1b/6893d4bc57e434fc93a6c9217c637d967a0b651d989f6e3265179375754a/aiohttp-3.14.3-cp314-cp314-win
32.whl", hash = "sha256:38901a84da3ce22249f6e860bf8f90d141bcab7da090cc398f8bb58c0e44b7da", size = 455872, upload-time = "2026-07-23T01:56:31.031Z" },
    { url = "https://files.pythonhosted.org/packages/f5/8b/c7baa1ba1eda4db6989baefe5de6d99834921b84ebd7918624febcb9f290/aiohttp-3.14.3-cp314-cp314-win
_amd64.whl", hash = "sha256:8b3b60de05f3dcb6f6a00f818bb2ec781cee4de0645f59ccaf99b1d1823b6100", size = 481030, upload-time = "2026-07-23T01:56:33.365Z"
 },
    { url = "https://files.pythonhosted.org/packages/22/8c/c29d067df825a2df88ca432db848aa2fe8199598359cc06c12b09320cac9/aiohttp-3.14.3-cp314-cp314-win
_arm64.whl", hash = "sha256:1576145bdceeb92382d899751e12743a3a5b8e460a841e3e50543859e54864dc", size = 453669, upload-time = "2026-07-23T01:56:35.731Z"
 },
    { url = "https://files.pythonhosted.org/packages/6a/a4/9c033beb355d39b6147980597ec9645e4729243f686ee4dc73945de72030/aiohttp-3.14.3-cp314-cp314t-ma
cosx_10_15_universal2.whl", hash = "sha256:8800c996b01c2772a783e3e46f3e1abd5823029adca0df54231960de9bfefa5b", size = 791403, upload-time = "2026-07-23
T01:56:37.972Z" },
    { url = "https://files.pythonhosted.org/packages/80/ca/87c32a0a7704583cfc49660bd817889bae5b830bf53b5dcb4e92145ac2da/aiohttp-3.14.3-cp314-cp314t-ma
cosx_10_15_x86_64.whl", hash = "sha256:ebe8e504f058fe91223351cecd2d9d6946c9d241bb0250d898ffbdf584cc72b0", size = 526413, upload-time = "2026-07-23T01:
56:40.523Z" },
    { url = "https://files.pythonhosted.org/packages/9e/d8/8ec0e471248c500acdce2be3f46db8fb62b5eb60efef072529cc85ee1d26/aiohttp-3.14.3-cp314-cp314t-ma
cosx_11_0_arm64.whl", hash = "sha256:30402d03a7c0ff52bce290b57e564e9079fd9d0cb545c8aba73f86a103162d2e", size = 532135, upload-time = "2026-07-23T01:56
:42.876Z" },
    { url = "https://files.pythonhosted.org/packages/fe/45/f8919fd936e8b79fcd9bda7b6d8e62613462a713f4f17987fd7c34399142/aiohttp-3.14.3-cp314-cp314t-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:9fc7b5bfec6573f3ae844f457fdde5adeb713f8b8e4a81ad64fc207b4938371
6", size = 1922742, upload-time = "2026-07-23T01:56:45.528Z" },
    { url = "https://files.pythonhosted.org/packages/f6/ec/9ca76b28a27525b0cc53e20842e0228b022f301ce1f436b7d814b4aaf2df/aiohttp-3.14.3-cp314-cp314t-ma
nylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:8a5fd34f7f7410d1730d5c2ba873cacb2eed3fede366feb268a70ba22581ed8f",
 size = 1787371, upload-time = "2026-07-23T01:56:48.045Z" },
    { url = "https://files.pythonhosted.org/packages/b1/04/6acdbf17315f7b55f1937e3387acb89a3cddeb4995689553d064af8e92ab/aiohttp-3.14.3-cp314-cp314t-ma
nylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:270d3dace9ca2f10f0da5d8ebe519b7a310fc6112ed916e32df5866df088855
3", size = 1912623, upload-time = "2026-07-23T01:56:50.605Z" },
    { url = "https://files.pythonhosted.org/packages/86/e6/438b0c79ca6f45eb9fd9817dd4c01a91919a38c0de5ee9e05e2b4dc0ece7/aiohttp-3.14.3-cp314-cp314t-ma
nylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:3ae5b3a59436d089b5395d910121a390feed4d00578eb95a0fd1a329fe963100", si
ze = 2005515, upload-time = "2026-07-23T01:56:53.153Z" },
    { url = "https://files.pythonhosted.org/packages/bb/6b/62cbd6577758699525f5c712d1ddef57d9875fbab0ae8d5f5a202fd598f8/aiohttp-3.14.3-cp314-cp314t-ma
nylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:2498f0fe69ead802f9675beca44a7c21c62fdaa4ec5145ea1c3ad6edbee29f85",
 size = 1879906, upload-time = "2026-07-23T01:56:55.818Z" },
    { url = "https://files.pythonhosted.org/packages/00/95/18bcbf830a21dc3aae24d8f6b6feaf3db1d2090242d00a7868db2ffb0b67/aiohttp-3.14.3-cp314-cp314t-ma
nylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:a0dc483c00da8b673abbb367eb6f8d8f4bcec30eb58529ea13cb42e7fd2dfa33", size = 1675849, up
load-time = "2026-07-23T01:56:58.861Z" },
    { url = "https://files.pythonhosted.org/packages/a9/19/47f4968659c5e23606c3790c80fc624e691c153d036148449ee84d31b287/aiohttp-3.14.3-cp314-cp314t-mu
sllinux_1_2_aarch64.whl", hash = "sha256:c7d3a97c678d34fc5b59da671ee9cd630096ddc643e7b5a30d54a2a6f3574d3f", size = 1843496, upload-time = "2026-07-23T
01:57:01.591Z" },
    { url = "https://files.pythonhosted.org/packages/64/af/38c33c4dd82fddcb4e56c4653b6f1072a8edbc6b7fa15809f14932c41e2d/aiohttp-3.14.3-cp314-cp314t-mu
sllinux_1_2_armv7l.whl", hash = "sha256:f8fb78a83c9e5f741ca3a68cfb455c1f5bb83b4e7249a3848b3cd78d0a8563b0", size = 1827746, upload-time = "2026-07-23T0
1:57:05.131Z" },
    { url = "https://files.pythonhosted.org/packages/a1/9d/0537cda4885ac8f5b7053d164dd06312f4c483a4edcb8ee5b8aaf2a989bf/aiohttp-3.14.3-cp314-cp314t-mu
sllinux_1_2_ppc64le.whl", hash = "sha256:74ab5b6a9fb13e873e5a90946588baecaf488745e1db1a4a5c433f971f035098", size = 1853810, upload-time = "2026-07-23T
01:57:08.043Z" },
    { url = "https://files.pythonhosted.org/packages/19/fe/26f9c5e6458385aa86497836b0dea6fb2f027827d63f37c7856cce9286ee/aiohttp-3.14.3-cp314-cp314t-mu
sllinux_1_2_riscv64.whl", hash = "sha256:bd52f811e65f6fb634b1047159657c98f52b407f8efec907bcfc09da9a4c0a25", size = 1668895, upload-time = "2026-07-23T
01:57:10.837Z" },
    { url = "https://files.pythonhosted.org/packages/ec/4c/618b1db9b9ba079b8875d2cdf78e7c4a3bf72903bd5850fee7dd9544600a/aiohttp-3.14.3-cp314-cp314t-mu
sllinux_1_2_s390x.whl", hash = "sha256:f0f177d1b195b9e06376cfd7d308d8a1b920909a609d03ac82a8c73bbb16d3b9", size = 1883833, upload-time = "2026-07-23T01
:57:13.672Z" },
    { url = "https://files.pythonhosted.org/packages/94/c6/bd959bd1e4771f9fd944e9e436224c48c77b018b73b519b5aad346335bcc/aiohttp-3.14.3-cp314-cp314t-mu
sllinux_1_2_x86_64.whl", hash = "sha256:498c6c623134f8e09a3c4e60bcd607a0b4590dd7dbf08dd40851b27cbb520ccb", size = 1844251, upload-time = "2026-07-23T0
1:57:16.593Z" },
    { url = "https://files.pythonhosted.org/packages/5e/19/08d41839658bdd44a0ed2480f3891705ecb487ce28c0dde62c9040c997e0/aiohttp-3.14.3-cp314-cp314t-wi
n32.whl", hash = "sha256:b304db572b4368edd8dda8a2274f73156fe15558fca4a917cb8a09fc47af5963", size = 474180, upload-time = "2026-07-23T01:57:19.306Z" },
    { url = "https://files.pythonhosted.org/packages/99/5d/3cd6ef0a2b2851f7ab913b5b079334781bd50ff56a323e4454063377a080/aiohttp-3.14.3-cp314-cp314t-wi
n_amd64.whl", hash = "sha256:b20032766aedf6261c7a566585a40867d092ac03a0d81592d5370ef9b054f99b", size = 500528, upload-time = "2026-07-23T01:57:21.762Z
" },
    { url = "https://files.pythonhosted.org/packages/a4/37/cfd1ed540a4d318da025590d96b728e63713c09e9377950fc655dadeb856/aiohttp-3.14.3-cp314-cp314t-wi
n_arm64.whl", hash = "sha256:2e1161602f45a54de2ce0905243a95f58cb42dcd378402f3697f5e0b21e9d2e7", size = 469280, upload-time = "2026-07-23T01:57:24.241Z
" },
]

[[package]]
name = "aiosignal"
version = "1.4.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "frozenlist" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/61/62/06741b579156360248d1ec624842ad0edf697050bbaf7c3e46394e106ad1/aiosignal-1.4.0.tar.gz", h
ash = "sha256:f47eecd9468083c2029cc99945502cb7708b082c232f9aca65da147157b251c7", size = 25007, upload-time = "2025-07-03T22:54:43.528Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/fb/76/641ae371508676492379f16e2fa48f4e2c11741bd63c48be4b12a6b09cba/aiosignal-1.4.0-py3-none-any.w
hl", hash = "sha256:053243f8b92b990551949e63930a839ff0cf0b0ebbe0597b0f3fb19e1a0fe82e", size = 7490, upload-time = "2025-07-03T22:54:42.156Z" },
]

[[package]]
```

```
name = "aiosqlite"
version = "0.22.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/4e/8a/64761f4005f17809769d23e518d915db74e6310474e733e3593cfc854ef1/aiosqlite-0.22.1.tar.gz",
hash = "sha256:043e0bd78d32888c0a9ca90fc788b38796843360c855a7262a532813133a0650", size = 14821, upload-time = "2025-12-23T19:25:43.997Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/00/b7/e3bf5133d697a08128598c8d0abc5e16377b51465a33756de24fa7dee953/aiosqlite-0.22.1-py3-none-any.
whl", hash = "sha256:21c002eb13823fad740196c5a2e9d8e62f6243bd9e7e4a1f87fb5e44ecb4fceb", size = 17405, upload-time = "2025-12-23T19:25:42.139Z" },
]

[[package]]
name = "alembic"
version = "1.19.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "mako" },
    { name = "sqlalchemy" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/fb/01/a48dab7827ac4421272399f7ed9a2ec17edd12c8bcde4417bd7b6821b71a/alembic-1.19.0.tar.gz", ha
sh = "sha256:6487c612fc719dcfa22b17d2dd5b2b458929641e6aa2f0b65b135727f5e6d501", size = 2069906, upload-time = "2026-08-04T18:57:04.599Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/34/79/59ab6f1fe72eee229de91aa393225389a85df12a0fbbbfa264efcf6d7872/alembic-1.19.0-py3-none-any.wh
l", hash = "sha256:cf839d3849116aab3cc047e09c6968b9bd6b2fde61b6bb7c1e97352fe5503580", size = 265738, upload-time = "2026-08-04T18:57:06.219Z" },
]

[[package]]
name = "annotated-doc"
version = "0.0.5"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/5a/8e/38aa427ed5402449e226975b649c5dc73ccadfefeb95e6aecb8f8ea4b6b6/annotated_doc-0.0.5.tar.gz
", hash = "sha256:c7e58ce09192557605d8bbd92836d7e1d520ac9580096042c0bfd197efacf1bb", size = 10758, upload-time = "2026-07-28T13:50:58.129Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/3e/30/e900b21425a860e195f32e37657aa1f7c7f2b1bfb26f03ca209b90933c06/annotated_doc-0.0.5-py3-none-a
ny.whl", hash = "sha256:117bac03a25ede5df5440e855b32d556049ca169ead221505badf432fed4b101", size = 5302, upload-time = "2026-07-28T13:50:57.239Z" },
]

[[package]]
name = "annotated-types"
version = "0.8.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/5f/56/a8120250d128bed162cd73c76d45f6ef9991f3e068f62a8ee060afa3104a/annotated_types-0.8.0.tar.
gz", hash = "sha256:13b2beaad985e05e2d6407ee4c4f35590b11f8d693a258a561055cac8f64cab7", size = 15893, upload-time = "2026-07-23T20:16:13.995Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/99/91/8acff4f5e50511b911bbccb72b8628a49c68ce14148cd9f6431094859a90/annotated_types-0.8.0-py3-none
-any.whl", hash = "sha256:f072f4d804ea359e4eaf198b1af7a8b0943881a87f31bb764f8bf219bb9419e0", size = 13427, upload-time = "2026-07-23T20:16:12.938Z" },
]

[[package]]
name = "anthropic"
version = "0.120.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "distro" },
    { name = "docstring-parser" },
    { name = "httpx" },
    { name = "jiter" },
    { name = "pydantic" },
    { name = "sniffio" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/d7/10/4ca013cb166f226bd89e0aeb0fcaff94f45ddf716d4925ce89475d3c587b/anthropic-0.120.2.tar.gz",
 hash = "sha256:9722efc10c27a30a69f5338ddacdb35bc6a64297a4e4ba729bf83af873d5fb3a", size = 1008421, upload-time = "2026-07-28T17:38:26.986Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/63/af/0f5db57b9397a0f3b7fc204cbef143401a7cadaf982330f97f1ce3d39f34/anthropic-0.120.2-py3-none-any
.whl", hash = "sha256:0f0bc2b381dc0eb41c8d886b815d79c2041cd2374f83aed36f574b6dc9c579c1", size = 1022851, upload-time = "2026-07-28T17:38:25.466Z" },
]

[[package]]
name = "anyio"
version = "4.14.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "idna" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/61/cc/a381afa6efea9f496eff839d4a6a1aed3bfafc7b3ab4b0d1b243a12573dd/anyio-4.14.2.tar.gz", hash
 = "sha256:cfa139f3ed1a23ee8f88a145ddb5ac7605b8bbfd8592baacd7ce3d8bb4313c7f", size = 260176, upload-time = "2026-07-12T20:29:07.082Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/da/35/f2287558c17e29fafc8ef3daf819bb9834061cfa43bff8014f7df7f63bdc/anyio-4.14.2-py3-none-any.whl"
, hash = "sha256:9f505dda5ac9f0c8309b5e8bd445a8c2bf7246f3ce950121e45ea15bc41d1494", size = 125813, upload-time = "2026-07-12T20:29:05.763Z" },
]

[[package]]
name = "asgiref"
version = "3.12.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/e6/26/3b59f2bdae5f640389becb1f673cded775287f5fc4f816309d9ca9a3f93d/asgiref-3.12.1.tar.gz", ha
sh = "sha256:59dcb51c272ad209d59bed5708a64a333083e86017d7fcdd67498eeab7784340", size = 42378, upload-time = "2026-07-14T09:56:18.087Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/c0/1b/54f4ad77cd8a584fa70746c47df988e002cf1ee1eba43364d46f87803647/asgiref-3.12.1-py3-none-any.wh
l", hash = "sha256:fe386d1c2bff7259ea95929266d12a8cf9a8b5a1c2598402967d8792e7a7c094", size = 25478, upload-time = "2026-07-14T09:56:16.926Z" },
]

[[package]]
name = "ast-serialize"
version = "0.6.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/58/ad/0d70a3a2d6e01968d985415259e8ec7ad3f777903f9b1c1f3c8c44642c60/ast_serialize-0.6.0.tar.gz
", hash = "sha256:aadd3ffcf4858c9726bf3515f7b199c7eadbe504f96028e4a87172c0da65a8fe", size = 61489, upload-time = "2026-06-30T20:02:55.555Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/3f/12/3e5f575f156555547c250a8b0d1347517a3a20fc7f4492e9703a69d4f45e/ast_serialize-0.6.0-cp314-cp31
4t-macosx_10_12_x86_64.whl", hash = "sha256:a7520b672827885bafeae7501f684d14d47d17e5f45256f9df547686cca52264", size = 1177640, upload-time = "2026-06-
30T20:02:06.708Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/a2/a4/921a9e27951627983b0f368859ea00f8330a551dc0bf4c2fdcb11855a98b/ast_serialize-0.6.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:a14191beec7e0c078d2fc1f6edc0aee88bcd4db9f18e1bc9f8052b559c22dddc", size = 1168111, upload-time = "2026-06-30T20:02:08.366Z" },
    { url = "https://files.pythonhosted.org/packages/00/69/950cf404de7b8782cf95e5c1237e25e2aa46177b287f39f9eeddf481fd6f/ast_serialize-0.6.0-cp314-cp314-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:32ef62ec34cf6be20ad77d4799556638fbdf187f3ae10698dfb20ef9f2c89516", size = 1227656, upload-time = "2026-06-30T20:02:09.843Z" },
    { url = "https://files.pythonhosted.org/packages/4c/a8/46f8f6a6479d9d2273980957bb091a506c55f5b95d3c029ee58518a78407/ast_serialize-0.6.0-cp314-cp314-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:13b7769970a39983b0adf2f38917b1cd3b8946f76df045756c3d741bc689f089", size = 1227706, upload-time = "2026-06-30T20:02:11.367Z" },
    { url = "https://files.pythonhosted.org/packages/b7/b9/9ac415bda0a40e49eab8fea3b2741c19c98bb84d57d62c4cfc6230eb67be/ast_serialize-0.6.0-cp314-cp314-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:6f7a408601bb3edaefb3bc67a4c01f5235e3253653b6a5729a2ee2382b35341c", size = 1431705, upload-time = "2026-06-30T20:02:12.737Z" },
    { url = "https://files.pythonhosted.org/packages/e5/06/8807115d441444879f7561b5eede5ac18fc80392f11826d61ccf31f503b1/ast_serialize-0.6.0-cp314-cp314-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:8670bfa51208a2c0c8d138928e40e998fab158f9200d53bb80c088b5b8eda7b8", size = 1249533, upload-time = "2026-06-30T20:02:14.571Z" },
    { url = "https://files.pythonhosted.org/packages/3e/c0/c2ba82ef9618650357d9421a1fdb27ffec862a7f57e8e2de82a3ccd11e12/ast_serialize-0.6.0-cp314-cp314-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:a4826809eb8597a8cd59fd924b6d7c285b8969a1e0007e2cb652cab62376270f", size = 1252619, upload-time = "2026-06-30T20:02:16.219Z" },
    { url = "https://files.pythonhosted.org/packages/0f/a7/fa31d52dd4102cede29fb9634e98d214129b2783b4f95528c6dc6a8f6587/ast_serialize-0.6.0-cp314-cp314-manylinux_2_31_riscv64.whl", hash = "sha256:577a6c189068686869f5f1ddc38363f3ae1808a4753b577266f9202071a7bb66", size = 1242983, upload-time = "2026-06-30T20:02:17.813Z" },
    { url = "https://files.pythonhosted.org/packages/b1/20/ddf742b5ad3c4bafd3466f2265037cfd99bc1b9a5ee46a5d58c90d523242/ast_serialize-0.6.0-cp314-cp314-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:085de7f62dc9cc247eb01e965a362707d1d90b1d89a82c5bf78301a60a3c417b", size = 1296148, upload-time = "2026-06-30T20:02:19.146Z" },
    { url = "https://files.pythonhosted.org/packages/24/cb/9f6f217cce8b3b632c5568b478d195a35e79dce4dbe309438cb89ba6ea4f/ast_serialize-0.6.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:9f8a8b78b13173de6a9ec22111d9be674874cd5bdccda04f14ae5ebc2bef403a", size = 1403826, upload-time = "2026-06-30T20:02:20.696Z" },
    { url = "https://files.pythonhosted.org/packages/2d/f8/9d16d4f0107a183924425cc0e7618d8bf76f96b45afa9ff19f924ed1ad57/ast_serialize-0.6.0-cp314-cp314-musllinux_1_2_armv7l.whl", hash = "sha256:f2ff3baffc3a29c1f15bc9098aa0c09763410262d5e6cef42116f7356c184554", size = 1502943, upload-time = "2026-06-30T20:02:22.034Z" },
    { url = "https://files.pythonhosted.org/packages/80/dd/bbc1c38756350dddf7e24acae1c9482ef42051c267417e019aecc1ed4075/ast_serialize-0.6.0-cp314-cp314-musllinux_1_2_i686.whl", hash = "sha256:0067b25fce104eaae5b88383de9ab803faeb671831e14ca698b771b356e2600f", size = 1497632, upload-time = "2026-06-30T20:02:23.517Z" },
    { url = "https://files.pythonhosted.org/packages/42/7e/9daffefcf5b97e6bb4c3e0b3c024c1aee9722f23d3cf7cd2ff80d6fb4a40/ast_serialize-0.6.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:c617417f9cbb0cb144f6283c3cbe0d2e0f01beaf9f608f662b21191058a626ec", size = 1448858, upload-time = "2026-06-30T20:02:24.889Z" },
    { url = "https://files.pythonhosted.org/packages/e5/1f/f9baaab81a677ea0af7d2458cac2f94ebcc85958f8a3c15ba9d9e5dab653/ast_serialize-0.6.0-cp314-cp314-win32.whl", hash = "sha256:5337cb256dcea3df9288205213d1601581536526b8f4da44b6974f1180f3252a", size = 1052600, upload-time = "2026-06-30T20:02:26.263Z" },
    { url = "https://files.pythonhosted.org/packages/9e/1f/41b535866519512d8cf6669cb2cff7823b7672bb6279c0333b4ff89d7d9f/ast_serialize-0.6.0-cp314-cp314-win_amd64.whl", hash = "sha256:2d947e45cafc4b09bd7528917fa84c517654a43de173c79785574b7b3068ac24", size = 1095570, upload-time = "2026-06-30T20:02:27.639Z" },
    { url = "https://files.pythonhosted.org/packages/50/64/e472fe3e3a2d33d874b987e8518aedf24562919e3b6161a4fa1797e89c0f/ast_serialize-0.6.0-cp314-cp314-win_arm64.whl", hash = "sha256:6e15ec740436e1a0d62de848641abe5f3a2f89a7f94907d534795ac91bbacf14", size = 1067267, upload-time = "2026-06-30T20:02:28.949Z" },
    { url = "https://files.pythonhosted.org/packages/52/19/ac8348ae8711c9b5ae834634f635780cab62a0f5e6f988882e048b89c2ae/ast_serialize-0.6.0-cp39-abi3-macosx_10_12_x86_64.whl", hash = "sha256:093cb8bb91b720d8523580498d031791bb1bbaa048599c3d21085d380e11a596", size = 1185367, upload-time = "2026-06-30T20:02:30.427Z" },
    { url = "https://files.pythonhosted.org/packages/c1/f6/ec7ec652c51db77c2f61d8573338e13e4704303265ccc658cb4031d9f354/ast_serialize-0.6.0-cp39-abi3-macosx_11_0_arm64.whl", hash = "sha256:e61580a69faf47e3689795367ed211f2a10fd741478cc0f36a0f128793360aad", size = 1178657, upload-time = "2026-06-30T20:02:31.964Z" },
    { url = "https://files.pythonhosted.org/packages/6f/02/613a7534a41d0122f37d1e0c64aa8ac78bfb831f8c92f6db057a311abb3c/ast_serialize-0.6.0-cp39-abi3-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:305802f2ce2a7c4e87835078ea85c58b586ddda8095b92fe2ead9364ae19c80a", size = 1238620, upload-time = "2026-06-30T20:02:33.664Z" },
    { url = "https://files.pythonhosted.org/packages/4d/21/087957bba486242afc52f49b2d9e21c9dad00289356cf9efe67084015a9d/ast_serialize-0.6.0-cp39-abi3-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:c7b8b8f0c42f752ea00b2b7d7c090b3f80d9c1c5c75cadf16423790a0cc74081", size = 1236075, upload-time = "2026-06-30T20:02:34.936Z" },
    { url = "https://files.pythonhosted.org/packages/82/04/78128bbb170071c2c72a210a181f1c00e11cc1cec60a8beef747b07f9201/ast_serialize-0.6.0-cp39-abi3-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:cd5b91b9e6f2356ace3a556963b0cd783b395fbbb0bb17b4defc283415466e77", size = 1441348, upload-time = "2026-06-30T20:02:36.245Z" },
    { url = "https://files.pythonhosted.org/packages/64/64/62fb99d6faf199b4c3e5b08a07136e9a0d7664bb249c6de3670e5b63e9b6/ast_serialize-0.6.0-cp39-abi3-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:4d6ef91590258ada18909b9caea344dac4de2013906b035473cd674a43f4b790", size = 1258580, upload-time = "2026-06-30T20:02:37.53Z" },
    { url = "https://files.pythonhosted.org/packages/ca/87/b4d6c38e0ccd5e85dc54cecdf933a152c60b28fe5d993a6d8a72fa6d5896/ast_serialize-0.6.0-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:dcbed41e9386059fc0261d602445ede0976c2ecec2939688bcbcb9ed0b6f28b7", size = 1261693, upload-time = "2026-06-30T20:02:39.123Z" },
    { url = "https://files.pythonhosted.org/packages/0e/4b/3676ca2191f39bafb75f93f99b2f429ec464586158fece2165f3572805dc/ast_serialize-0.6.0-cp39-abi3-manylinux_2_31_riscv64.whl", hash = "sha256:cdc4e6f930b9090c2f92c9036ad12ffb8e6e44d4a5ba06f1458a05d60f203f7b", size = 1252517, upload-time = "2026-06-30T20:02:40.511Z" },
    { url = "https://files.pythonhosted.org/packages/f3/58/494ef8c4b4acb2f4a265ac934caf45f792a08fe27d6b853de35ad991941a/ast_serialize-0.6.0-cp39-abi3-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:897ac47b5637be41c0c07061c8a912fafa967ef1dc73fa115e4bfa70882a093b", size = 1304843, upload-time = "2026-06-30T20:02:41.961Z" },
    { url = "https://files.pythonhosted.org/packages/b1/f2/13736d920ab3d49bbee80ef1a277dd7b7aaf3b3545efd9d2a8114fe05525/ast_serialize-0.6.0-cp39-abi3-musllinux_1_2_aarch64.whl", hash = "sha256:c4af9a1386166e40ed01464991806f89038a2d89782576c7774876fa77034e32", size = 1413698, upload-time = "2026-06-30T20:02:44.179Z" },
    { url = "https://files.pythonhosted.org/packages/a8/5a/e046f3899e2acba4677d7427b76431443a1aa1a0e583dfb05b55b69d55cf/ast_serialize-0.6.0-cp39-abi3-musllinux_1_2_armv7l.whl", hash = "sha256:c901adbd750029b9ac4ad3d6aa56853e0ad4875119fbf52b7b8298afc223828b", size = 1512209, upload-time = "2026-06-30T20:02:45.584Z" },
    { url = "https://files.pythonhosted.org/packages/cc/c7/e42aaca7bb2d22a7c06d5a8c7930086c5a334e93d716e6fa5e6647a4515f/ast_serialize-0.6.0-cp39-abi3-musllinux_1_2_i686.whl", hash = "sha256:3ae22a366b752ab4496191525b78b097b5b72d531752e3c1dd7e383a8f2c8a1a", size = 1508464, upload-time = "2026-06-30T20:02:46.942Z" },
    { url = "https://files.pythonhosted.org/packages/95/93/5524a3dc6c3f593de3228ed9cbef73afa047625b7000ec21b7f58e6eb4d4/ast_serialize-0.6.0-cp39-abi3-musllinux_1_2_x86_64.whl", hash = "sha256:4ed29121da8b3fdc291002801a1de0f76248fa07dce89157a5f277842cf6126e", size = 1457164, upload-time = "2026-06-30T20:02:48.294Z" },
    { url = "https://files.pythonhosted.org/packages/4f/c0/36a6ffb4d653cf621427b4c4928671f53ad800c453474de2b82564a44ad9/ast_serialize-0.6.0-cp39-abi3-pyemscripten_2026_0_wasm32.whl", hash = "sha256:b1dac4e09d341c1300ba69cdcbe62867b32a8c75d90db9bf4d083bec3b039f0b", size = 863014, upload-time = "2026-06-30T20:02:49.742Z" },
    { url = "https://files.pythonhosted.org/packages/09/c7/7d5ad8b49e1278e1c2a1e0274bd7850560b3f09313aa00c13bc8d5544792/ast_serialize-0.6.0-cp39-abi3-win32.whl", hash = "sha256:82c312a7844d2fdeb4d5c48bd3d215bf940dafd4704e1a9bcf252a99010a99b1", size = 1063165, upload-time = "2026-06-30T20:02:50.98Z" },
    { url = "https://files.pythonhosted.org/packages/47/ae/6710c14ecb276031cf10249f6adf5a59e2d3fdb3b5183bd59f70524067ee/ast_serialize-0.6.0-cp39-abi3-win_amd64.whl", hash = "sha256:113b58346f9ceb664352032770caca817d4a3c86f611c6088e6ef65ddaa70f0e", size = 1101444, upload-time = "2026-06-30T20:02:52.554Z" },
    { url = "https://files.pythonhosted.org/packages/66/40/c53deb2cd0c9b0fb636d24d9f40924cf2e65028e6b20b10cd5c1eeb2c730/ast_serialize-0.6.0-cp39-abi3-win_arm64.whl", hash = "sha256:ccd132fe8db56f61fe743b1f644d01b8d65b83248a8da506f3132bda86d6ed5e", size = 1072965, upload-time = "2026-06-30T20:02:54.097Z" },
]

[[package]]
name = "asyncpg"
version = "0.31.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/fe/cc/d18065ce2380d80b1bcce927c24a2642efd38918e33fd724bc4bca904877/asyncpg-0.31.0.tar.gz", hash = "sha256:c989386c83940bfbd787180f2b1519415e2d3d6277a70d9d0f0145ac73500735", size = 993667, upload-time = "2025-11-24T23:27:00.812Z" }
wheels = [
```

```
    { url = "https://files.pythonhosted.org/packages/2a/a6/59d0a146e61d20e18db7396583242e32e0f120693b67a8de43f1557033e2/asyncpg-0.31.0-cp312-cp312-mac
osx_10_13_x86_64.whl", hash = "sha256:b44c31e1efc1c15188ef183f287c728e2046abb1d26af4d20858215d50d91fad", size = 662042, upload-time = "2025-11-24T23:2
5:49.578Z" },
    { url = "https://files.pythonhosted.org/packages/36/01/ffaa189dcb63a2471720615e60185c3f6327716fdc0fc04334436fbb7c65/asyncpg-0.31.0-cp312-cp312-mac
osx_11_0_arm64.whl", hash = "sha256:0c89ccf741c067614c9b5fc7f1fc6f3b61ab05ae4aaa966e6fd6b93097c7d20d", size = 638504, upload-time = "2025-11-24T23:25:
51.501Z" },
    { url = "https://files.pythonhosted.org/packages/9f/62/3f699ba45d8bd24c5d65392190d19656d74ff0185f42e19d0bbd973bb371/asyncpg-0.31.0-cp312-cp312-man
ylinux_2_28_aarch64.whl", hash = "sha256:12b3b2e39dc5470abd5e98c8d3373e4b1d1234d9fbdedf538798b2c13c64460a", size = 3426241, upload-time = "2025-11-24T
23:25:53.278Z" },
    { url = "https://files.pythonhosted.org/packages/8c/d1/a867c2150f9c6e7af6462637f613ba67f78a314b00db220cd26ff559d532/asyncpg-0.31.0-cp312-cp312-man
ylinux_2_28_x86_64.whl", hash = "sha256:aad7a33913fb8bcb5454313377cc330fbb19a0cd5faa7272407d8a0c4257b671", size = 3520321, upload-time = "2025-11-24T2
3:25:54.982Z" },
    { url = "https://files.pythonhosted.org/packages/7a/1a/cce4c3f246805ecd285a3591222a2611141f1669d002163abef999b60f98/asyncpg-0.31.0-cp312-cp312-mus
llinux_1_2_aarch64.whl", hash = "sha256:3df118d94f46d85b2e434fd62c84cb66d5834d5a890725fe625f498e72e4d5ec", size = 3316685, upload-time = "2025-11-24T2
3:25:57.43Z" },
    { url = "https://files.pythonhosted.org/packages/40/ae/0fc961179e78cc579e138fad6eb580448ecae64908f95b8cb8ee2f241f67/asyncpg-0.31.0-cp312-cp312-mus
llinux_1_2_x86_64.whl", hash = "sha256:bd5b6efff3c17c3202d4b37189969acf8927438a238c6257f66be3c426beba20", size = 3471858, upload-time = "2025-11-24T23
:25:59.636Z" },
    { url = "https://files.pythonhosted.org/packages/52/b2/b20e09670be031afa4cbfabd645caece7f85ec62d69c312239de568e058e/asyncpg-0.31.0-cp312-cp312-win
32.whl", hash = "sha256:027eaa61361ec735926566f995d959ade4796f6a49d3bde17e5134b9964f9ba8", size = 527852, upload-time = "2025-11-24T23:26:01.084Z" },
    { url = "https://files.pythonhosted.org/packages/b5/f0/f2ed1de154e15b107dc692262395b3c17fc34eafe2a78fc2115931561730/asyncpg-0.31.0-cp312-cp312-win
_amd64.whl", hash = "sha256:72d6bdcbc93d608a1158f17932de2321f68b1a967a13e014998db87a72ed3186", size = 597175, upload-time = "2025-11-24T23:26:02.564Z"
 },
    { url = "https://files.pythonhosted.org/packages/95/11/97b5c2af72a5d0b9bc3fa30cd4b9ce22284a9a943a150fdc768763caf035/asyncpg-0.31.0-cp313-cp313-mac
osx_10_13_x86_64.whl", hash = "sha256:c204fab1b91e08b0f47e90a75d1b3c62174dab21f670ad6c5d0f243a228f015b", size = 661111, upload-time = "2025-11-24T23:2
6:04.467Z" },
    { url = "https://files.pythonhosted.org/packages/1b/71/157d611c791a5e2d0423f09f027bd499935f0906e0c2a416ce712ba51ef3/asyncpg-0.31.0-cp313-cp313-mac
osx_11_0_arm64.whl", hash = "sha256:54a64f91839ba59008eccf7aad2e93d6e3de688d796f35803235ea1c4898ae1e", size = 636928, upload-time = "2025-11-24T23:26:
05.944Z" },
    { url = "https://files.pythonhosted.org/packages/2e/fc/9e3486fb2bbe69d4a867c0b76d68542650a7ff1574ca40e84c3111bb0c6e/asyncpg-0.31.0-cp313-cp313-man
ylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:c0e0822b1038dc7253b337b0f3f676cadc4ac31b126c5d42691c39691962e403
", size = 3424067, upload-time = "2025-11-24T23:26:07.957Z" },
    { url = "https://files.pythonhosted.org/packages/12/c6/8c9d076f73f07f995013c791e018a1cd5f31823c2a3187fc8581706aa00f/asyncpg-0.31.0-cp313-cp313-man
ylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:bef056aa502ee34204c161c72ca1f3c274917596877f825968368b2c33f585f4",
size = 3518156, upload-time = "2025-11-24T23:26:09.591Z" },
    { url = "https://files.pythonhosted.org/packages/ae/3b/60683a0baf50fbc546499cfb53132cb6835b92b529a05f6a81471ab60d0c/asyncpg-0.31.0-cp313-cp313-mus
llinux_1_2_aarch64.whl", hash = "sha256:0bfbcc5b7ffcd9b75ab1558f00db2ae07db9c80637ad1b2469c43df79d7a5ae2", size = 3319636, upload-time = "2025-11-24T2
3:26:11.168Z" },
    { url = "https://files.pythonhosted.org/packages/50/dc/8487df0f69bd398a61e1792b3cba0e47477f214eff085ba0efa7eac9ce87/asyncpg-0.31.0-cp313-cp313-mus
llinux_1_2_x86_64.whl", hash = "sha256:22bc525ebbdc24d1261ecbf6f504998244d4e3be1721784b5f64664d61fbe602", size = 3472079, upload-time = "2025-11-24T23
:26:13.164Z" },
    { url = "https://files.pythonhosted.org/packages/13/a1/c5bbeeb8531c05c89135cb8b28575ac2fac618bcb60119ee9696c3faf71c/asyncpg-0.31.0-cp313-cp313-win
32.whl", hash = "sha256:f890de5e1e4f7e14023619399a471ce4b71f5418cd67a51853b9910fdfa73696", size = 527606, upload-time = "2025-11-24T23:26:14.78Z" },
    { url = "https://files.pythonhosted.org/packages/91/66/b25ccb84a246b470eb943b0107c07edcae51804912b824054b3413995a10/asyncpg-0.31.0-cp313-cp313-win
_amd64.whl", hash = "sha256:dc5f2fa9916f292e5c5c8b2ac2813763bcd7f58e130055b4ad8a0531314201ab", size = 596569, upload-time = "2025-11-24T23:26:16.189Z"
 },
    { url = "https://files.pythonhosted.org/packages/3c/36/e9450d62e84a13aea6580c83a47a437f26c7ca6fa0f0fd40b6670793ea30/asyncpg-0.31.0-cp314-cp314-mac
osx_10_15_x86_64.whl", hash = "sha256:f6b56b91bb0ffc328c4e3ed113136cddd9deefdf5f79ab448598b9772831df44", size = 660867, upload-time = "2025-11-24T23:2
6:17.631Z" },
    { url = "https://files.pythonhosted.org/packages/82/4b/1d0a2b33b3102d210439338e1beea616a6122267c0df459ff0265cd5807a/asyncpg-0.31.0-cp314-cp314-mac
osx_11_0_arm64.whl", hash = "sha256:334dec28cf20d7f5bb9e45b39546ddf247f8042a690bff9b9573d00086e69cb5", size = 638349, upload-time = "2025-11-24T23:26:
19.689Z" },
    { url = "https://files.pythonhosted.org/packages/41/aa/e7f7ac9a7974f08eff9183e392b2d62516f90412686532d27e196c0f0eeb/asyncpg-0.31.0-cp314-cp314-man
ylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:98cc158c53f46de7bb677fd20c417e264fc02b36d901cc2a43bd6cb0dc6dbfd2
", size = 3410428, upload-time = "2025-11-24T23:26:21.275Z" },
    { url = "https://files.pythonhosted.org/packages/6f/de/bf1b60de3dede5c2731e6788617a512bc0ebd9693eac297ee74086f101d7/asyncpg-0.31.0-cp314-cp314-man
ylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9322b563e2661a52e3cdbc93eed3be7748b289f792e0011cb2720d278b366ce2",
size = 3471678, upload-time = "2025-11-24T23:26:23.627Z" },
    { url = "https://files.pythonhosted.org/packages/46/78/fc3ade003e22d8bd53aaf8f75f4be48f0b460fa73738f0391b9c856a9147/asyncpg-0.31.0-cp314-cp314-mus
llinux_1_2_aarch64.whl", hash = "sha256:19857a358fc811d82227449b7ca40afb46e75b33eb8897240c3839dd8b744218", size = 3313505, upload-time = "2025-11-24T2
3:26:25.235Z" },
    { url = "https://files.pythonhosted.org/packages/bf/e9/73eb8a6789e927816f4705291be21f2225687bfa97321e40cd23055e903a/asyncpg-0.31.0-cp314-cp314-mus
llinux_1_2_x86_64.whl", hash = "sha256:ba5f8886e850882ff2c2ace5732300e99193823e8107e2c53ef01c1ebfa1e85d", size = 3434744, upload-time = "2025-11-24T23
:26:26.944Z" },
    { url = "https://files.pythonhosted.org/packages/08/4b/f10b880534413c65c5b5862f79b8e81553a8f364e5238832ad4c0af71b7f/asyncpg-0.31.0-cp314-cp314-win
32.whl", hash = "sha256:cea3a0b2a14f95834cee29432e4ddc399b95700eb1d51bbc5bfee8f31fa07b2b", size = 532251, upload-time = "2025-11-24T23:26:28.404Z" },
    { url = "https://files.pythonhosted.org/packages/d3/2d/7aa40750b7a19efa5d66e67fc06008ca0f27ba1bd082e457ad82f59aba49/asyncpg-0.31.0-cp314-cp314-win
_amd64.whl", hash = "sha256:04d19392716af6b029411a0264d92093b6e5e8285ae97a39957b9a9c14ea72be", size = 604901, upload-time = "2025-11-24T23:26:30.34Z"
},
    { url = "https://files.pythonhosted.org/packages/ce/fe/b9dfe349b83b9dee28cc42360d2c86b2cdce4cb551a2c2d27e156bcac84d/asyncpg-0.31.0-cp314-cp314t-ma
cosx_10_15_x86_64.whl", hash = "sha256:bdb957706da132e982cc6856bb2f7b740603472b54c3ebc77fe60ea3e57e1bd2", size = 702280, upload-time = "2025-11-24T23:
26:32Z" },
    { url = "https://files.pythonhosted.org/packages/6a/81/e6be6e37e560bd91e6c23ea8a6138a04fd057b08cf63d3c5055c98e81c1d/asyncpg-0.31.0-cp314-cp314t-ma
cosx_11_0_arm64.whl", hash = "sha256:6d11b198111a72f47154fa03b85799f9be63701e068b43f84ac25da0bda9cb31", size = 682931, upload-time = "2025-11-24T23:26
:33.572Z" },
    { url = "https://files.pythonhosted.org/packages/a6/45/6009040da85a1648dd5bc75b3b0a062081c483e75a1a29041ae63a0bf0dc/asyncpg-0.31.0-cp314-cp314t-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:18c83b03bc0d1b23e6230f5bf8d4f217dc9bc08644ce0502a9d91dc9e634a9c
7", size = 3581608, upload-time = "2025-11-24T23:26:35.638Z" },
    { url = "https://files.pythonhosted.org/packages/7e/06/2e3d4d7608b0b2b3adbee0d0bd6a2d29ca0fc4d8a78f8277df04e2d1fd7b/asyncpg-0.31.0-cp314-cp314t-ma
nylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:e009abc333464ff18b8f6fd146addffd9aaf63e79aa3bb40ab7a4c332d0c5e9e",
 size = 3498738, upload-time = "2025-11-24T23:26:37.275Z" },
    { url = "https://files.pythonhosted.org/packages/7d/aa/7d75ede780033141c51d83577ea23236ba7d3a23593929b32b49db8ed36e/asyncpg-0.31.0-cp314-cp314t-mu
sllinux_1_2_aarch64.whl", hash = "sha256:3b1fbcb0e396a5ca435a8826a87e5c2c2cc0c8c68eb6fadf82168056b0e53a8c", size = 3401026, upload-time = "2025-11-24T
23:26:39.423Z" },
    { url = "https://files.pythonhosted.org/packages/ba/7a/15e37d45e7f7c94facc1e9148c0e455e8f33c08f0b8a0b1deb2c5171771b/asyncpg-0.31.0-cp314-cp314t-mu
sllinux_1_2_x86_64.whl", hash = "sha256:8df714dba348efcc162d2adf02d213e5fab1bd9f557e1305633e851a61814a7a", size = 3429426, upload-time = "2025-11-24T2
3:26:41.032Z" },
    { url = "https://files.pythonhosted.org/packages/13/d5/71437c5f6ae5f307828710efbe62163974e71237d5d46ebd2869ea052d10/asyncpg-0.31.0-cp314-cp314t-wi
n32.whl", hash = "sha256:1b41f1afb1033f2b44f3234993b15096ddc9cd71b21a42dbd87fc6a57b43d65d", size = 614495, upload-time = "2025-11-24T23:26:42.659Z" },
    { url = "https://files.pythonhosted.org/packages/3c/d7/8fb3044eaef08a310acfe23dae9a8e2e07d305edc29a53497e52bc76eca7/asyncpg-0.31.0-cp314-cp314t-wi
n_amd64.whl", hash = "sha256:bd4107bb7cdd0e9e65fae66a62afd3a249663b844fa34d479f6d5b3bef9c04c3", size = 706062, upload-time = "2025-11-24T23:26:44.086Z
" },
]

[[package]]
name = "attrs"
version = "26.1.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/9a/8e/82a0fe20a541c03148528be8cac2408564a6c9a0cc7e9171802bc1d26985/attrs-26.1.0.tar.gz", hash
 = "sha256:d03ceb89cb322a8fd706d4fb91940737b6642aa36998fe130a9bc96c985eff32", size = 952055, upload-time = "2026-03-19T14:22:25.026Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/64/b4/17d4b0b2a2dc85a6df63d1157e028ed19f90d4cd97c36717afef2bc2f395/attrs-26.1.0-py3-none-any.whl"
, hash = "sha256:c647aa4a12dfbad9333ca4e71fe62ddc36f4e63b2d260a37a8b83d2f043ac309", size = 67548, upload-time = "2026-03-19T14:22:23.645Z" },
]
```

```
[[package]]
name = "backoff"
version = "2.2.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/47/d7/5bbeb12c44d7c4f2fb5b56abce497eb5ed9f34d85701de869acedd602619/backoff-2.2.1.tar.gz", has
h = "sha256:03f829f5bb192318082164 3f8753b0502c3b682293992485b0eef2807afa5cba", size = 17001, upload-time = "2022-10-05T19:19:32.061Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/df/73/b6e24bd22e6720ca8ee9a85a0c4a2971af8497d8f3193fa05390cbd46e09/backoff-2.2.1-py3-none-any.whl
", hash = "sha256:63579f9a0628e06278f7e47b7d7d5b6ce20dc65c5e96a6f3ca99a6adca0396e8", size = 15148, upload-time = "2022-10-05T19:19:30.546Z" },
]

[[package]]
name = "banks"
version = "2.4.5"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "deprecated" },
    { name = "filetype" },
    { name = "griffe" },
    { name = "jinja2" },
    { name = "platformdirs" },
    { name = "pydantic" },
]
sdist = { url = "https://files.pythonhosted.org/packages/a1/b5/4784ee9518b97f9f69c714a4303f9a6186a7e4ff2349f89e24767e9754d9/banks-2.4.5.tar.gz", hash
= "sha256:ff575732fc67d5493a73c21e0d7268bc49e86fff02b0b8735e8efb9fcb9af3a4", size = 190822, upload-time = "2026-07-07T08:14:12.055Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/50/b0/4d34cb2fe538aeb6d4b0723b3339cb932120db084a5c9a3c6966a26bbe1b/banks-2.4.5-py3-none-any.whl",
 hash = "sha256:ac2e0091b4c79379d4773c9d04a138a0d937ee27c5803bf0142acc6d6769eea1", size = 36145, upload-time = "2026-07-07T08:14:10.974Z" },
]

[[package]]
name = "bcrypt"
version = "5.0.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/d4/36/3329e2518d70ad8e2e5817d5a4cac6bba05a47767ec416c7d020a965f408/bcrypt-5.0.0.tar.gz", hash
 = "sha256:f748f7c2d6fd375cc93d3fba7ef4a9e3a092421b8dbf34d8d4dc06be9492dfdd", size = 25386, upload-time = "2025-09-25T19:50:47.829Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/13/85/3e65e01985fddf25b64ca67275bb5bdb4040bd1a53b66d355c6c37c8a680/bcrypt-5.0.0-cp313-cp313t-maco
sx_10_12_universal2.whl", hash = "sha256:f3c08197f3039bec79cee59a606d62b96b16669cff3949f21e74796b6e3cd2be", size = 481806, upload-time = "2025-09-25T1
9:49:05.102Z" },
    { url = "https://files.pythonhosted.org/packages/44/dc/01eb79f12b177017a726cbf78330eb0eb442fae0e7b3dfd84ea2849552f3/bcrypt-5.0.0-cp313-cp313t-many
linux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:200af71bc25f22006f4069060c88ed36f8aa4ff7f53e67ff04d2ab3f1e79a5b2", size = 268626, upload
-time = "2025-09-25T19:49:06.723Z" },
    { url = "https://files.pythonhosted.org/packages/8c/cf/e82388ad5959c40d6afd94fb4743cc077129d45b952d46bdc3180310e2df/bcrypt-5.0.0-cp313-cp313t-many
linux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:baade0a5657654c2984468efb7d6c110db87ea63ef5a4b54732e7e337253e44f", size = 271853, upload-t
ime = "2025-09-25T19:49:08.028Z" },
    { url = "https://files.pythonhosted.org/packages/ec/86/7134b9dae7cf0efa85671651341f6afa695857fae172615e960fb6a466fa/bcrypt-5.0.0-cp313-cp313t-many
linux_2_28_aarch64.whl", hash = "sha256:c58b56cdfb03202b3bcc9fd8daee8e8e9b6d7e3163aa97c631dfcfcc24d36c86", size = 269793, upload-time = "2025-09-25T19
:49:09.727Z" },
    { url = "https://files.pythonhosted.org/packages/cc/82/6296688ac1b9e503d034e7d0614d56e80c5d1a08402ff856a4549cb59207/bcrypt-5.0.0-cp313-cp313t-many
linux_2_28_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:4bfd2a34de661f34d0bda43c3e4e79df586e4716ef401fe31ea39d69d581ef23", size = 289930, upload-
time = "2025-09-25T19:49:11.204Z" },
    { url = "https://files.pythonhosted.org/packages/d1/18/884a44aa47f2a3b88dd09bc05a1e40b57878ecd111d17e5bba6f09f8bb77/bcrypt-5.0.0-cp313-cp313t-many
linux_2_28_x86_64.whl", hash = "sha256:ed2e1365e31fc73f1825fa830f1c8f8917ca1b3ca6185773b349c20fd606cec2", size = 272194, upload-time = "2025-09-25T19:
49:12.524Z" },
    { url = "https://files.pythonhosted.org/packages/0e/8f/371a3ab33c6982070b674f1788e05b656cfbf5685894acbfef0c65483a59/bcrypt-5.0.0-cp313-cp313t-many
linux_2_34_aarch64.whl", hash = "sha256:83e787d7a84dbbfba6f250dd7a5efd689e935f03dd83b0f919d39349e1f23f83", size = 269381, upload-time = "2025-09-25T19
:49:14.308Z" },
    { url = "https://files.pythonhosted.org/packages/b1/34/7e4e6abb7a8778db6422e88b1f06eb07c47682313997ee8a8f9352e5a6f1/bcrypt-5.0.0-cp313-cp313t-many
linux_2_34_x86_64.whl", hash = "sha256:137c5156524328a24b9fac1cb5db0ba618bc97d11970b39184c1d87dc4bf1746", size = 271750, upload-time = "2025-09-25T19:
49:15.584Z" },
    { url = "https://files.pythonhosted.org/packages/c0/1b/54f416be2499bd72123c70d98d36c6cd61a4e33d9b89562c22481c81bb30/bcrypt-5.0.0-cp313-cp313t-musl
linux_1_1_aarch64.whl", hash = "sha256:38cac74101777a6a7d3b3e3cfefa57089b5ada650dce2baf0cbdd9d65db22a9e", size = 303757, upload-time = "2025-09-25T19:
49:17.244Z" },
    { url = "https://files.pythonhosted.org/packages/13/62/062c24c7bcf9d2826a1a843d0d605c65a755bc98002923d01fd61270705a/bcrypt-5.0.0-cp313-cp313t-musl
linux_1_1_x86_64.whl", hash = "sha256:d8d65b564ec849643d9f7ea05c6d9f0cd7ca23bdd4ac0c2dbef1104ab504543d", size = 306740, upload-time = "2025-09-25T19:4
9:18.693Z" },
    { url = "https://files.pythonhosted.org/packages/d5/c8/1fdbfc8c0f20875b6b4020f3c7dc447b8de60aa0be5faaf009d24242aec9/bcrypt-5.0.0-cp313-cp313t-musl
linux_1_2_aarch64.whl", hash = "sha256:741449132f64b3524e95cd30e5cd3343006ce146088f074f31ab26b94e6c75ba", size = 334197, upload-time = "2025-09-25T19:
49:20.523Z" },
    { url = "https://files.pythonhosted.org/packages/a6/c1/8b84545382d75bef226fbc6588af0f7b7d095f7cd6a670b42a86243183cd/bcrypt-5.0.0-cp313-cp313t-musl
linux_1_2_x86_64.whl", hash = "sha256:212139484ab3207b1f0c00633d3be92fef3c5f0af17cad155679d03ff2ee1e41", size = 352974, upload-time = "2025-09-25T19:4
9:22.254Z" },
    { url = "https://files.pythonhosted.org/packages/10/a6/ffb49d4254ed085e62e3e5dd05982b4393e32fe1e49bb1130186617c29cd/bcrypt-5.0.0-cp313-cp313t-win3
2.whl", hash = "sha256:9d52ed507c2488eddd6a95bccee4e808d3234fa78dd370e24bac65a21212b861", size = 148498, upload-time = "2025-09-25T19:49:24.134Z" },
    { url = "https://files.pythonhosted.org/packages/48/a9/259559edc85258b6d5fc5471a62a3299a6aa37a6611a169756bf4689323c/bcrypt-5.0.0-cp313-cp313t-win_
amd64.whl", hash = "sha256:f6984a24db30548fd39a44360532898c33528b74aedf81c26cf29c51ee47057e", size = 145853, upload-time = "2025-09-25T19:49:25.702Z"
},
    { url = "https://files.pythonhosted.org/packages/2d/df/9714173403c7e8b245acf8e4be8876aac64a209d1b392af457c79e60492e/bcrypt-5.0.0-cp313-cp313t-win_
arm64.whl", hash = "sha256:9fffdb387abe6aa775af36ef16f55e318dcda4194ddbf82007a6f21da29de8f5", size = 139626, upload-time = "2025-09-25T19:49:26.928Z"
},
    { url = "https://files.pythonhosted.org/packages/f8/14/c18006f91816606a4abe294ccc5d1e6f0e42304df5a33710e9e8e95416e1/bcrypt-5.0.0-cp314-cp314t-maco
sx_10_12_universal2.whl", hash = "sha256:4870a52610537037adb382444fefd3706d96d663ac44cbb2f37e3919dca3d7ef", size = 481862, upload-time = "2025-09-25T1
9:49:28.365Z" },
    { url = "https://files.pythonhosted.org/packages/67/49/dd074d831f00e589537e07a0725cf0e220d1f0d5d8e85ad5bbff251c45aa/bcrypt-5.0.0-cp314-cp314t-many
linux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:48f753100931605686f74e27a7b49238122aa761a9aefe9373265b8b7aa43ea4", size = 268544, upload
-time = "2025-09-25T19:49:30.39Z" },
    { url = "https://files.pythonhosted.org/packages/f5/91/50ccba088b8c474545b034a1424d05195d9fcbaaf802ab8bfe2be5a4e0d7/bcrypt-5.0.0-cp314-cp314t-many
linux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:f70aadb7a809305226daedf75d90379c397b094755a710d7014b8b117df1ebbf", size = 271787, upload-t
ime = "2025-09-25T19:49:32.144Z" },
    { url = "https://files.pythonhosted.org/packages/aa/e7/d7dba133e02abcda3b52087a7eea8c0d4f64d3e593b4fffc10c31b7061f3/bcrypt-5.0.0-cp314-cp314t-many
linux_2_28_aarch64.whl", hash = "sha256:744d3c6b164caa658adcb72cb8cc9ad9b4b75c7db507ab4bc2480474a51989da", size = 269753, upload-time = "2025-09-25T19
:49:33.885Z" },
    { url = "https://files.pythonhosted.org/packages/33/fc/5b145673c4b8d01018307b5c2c1fc87a6f5a436f0ad56607aee389de8ee3/bcrypt-5.0.0-cp314-cp314t-many
linux_2_28_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:a28bc05039bdf3289d757f49d616ab3efe8cf40d8e8001ccdd621cd4f98f4fc9", size = 289587, upload-
time = "2025-09-25T19:49:35.144Z" },
    { url = "https://files.pythonhosted.org/packages/27/d7/1ff22703ec6d4f90e62f1a5654b8867ef96bafb8e8102c2288333e1a6ca6/bcrypt-5.0.0-cp314-cp314t-many
linux_2_28_x86_64.whl", hash = "sha256:7f277a4b3390ab4bebe597800a90da0edae882c6196d3038a73adf446c4f969f", size = 272178, upload-time = "2025-09-25T19:
49:36.793Z" },
    { url = "https://files.pythonhosted.org/packages/c8/88/815b6d558a1e4d40ece04a2f84865b0fef233513bd85fd0e40c294272d62/bcrypt-5.0.0-cp314-cp314t-many
linux_2_34_aarch64.whl", hash = "sha256:79cfa161eda8d2ddf29acad370356b47f02387153b11d46042e93a0a95127493", size = 269295, upload-time = "2025-09-25T19
:49:38.164Z" },
    { url = "https://files.pythonhosted.org/packages/51/8c/e0db387c79ab4931fc89827d37608c31cc57b6edc08ccd2386139028dc0d/bcrypt-5.0.0-cp314-cp314t-many
linux_2_34_x86_64.whl", hash = "sha256:a5393eae5722bcef046a990b84dff02b954904c36a194f6cfc817d7dca6c6f0b", size = 271700, upload-time = "2025-09-25T19:
49:39.917Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/06/83/1570eddd150f572dbe9fc00f6203a89fc7d4226821f67328a85c330f239/bcrypt-5.0.0-cp314-cp314t-musl
linux_1_2_aarch64.whl", hash = "sha256:7f4c94dec1b5ab5d522750cb059bb9409ea8872d4494fd152b53cca99f1ddd8c", size = 334034, upload-time = "2025-09-25T19:
49:41.227Z" },
    { url = "https://files.pythonhosted.org/packages/c9/f2/ea64e51a65e56ae7a8a4ec236c2bfbdd4b23008abd50ac33fbb2d1d15424/bcrypt-5.0.0-cp314-cp314t-musl
linux_1_2_x86_64.whl", hash = "sha256:0cae4cb350934dfd74c020525eeae0a5f79257e8a201c0c176f4b84fdbf2a4b4", size = 352766, upload-time = "2025-09-25T19:4
9:43.08Z" },
    { url = "https://files.pythonhosted.org/packages/d7/d4/1a388d21ee66876f27d1a1f41287897d0c0f1712ef97d395d708ba93004c/bcrypt-5.0.0-cp314-cp314t-win3
2.whl", hash = "sha256:b17366316c654e1ad0306a6858e189fc835eca39f7eb2cafd6aaca8ce0c40a2e", size = 152449, upload-time = "2025-09-25T19:49:44.971Z" },
    { url = "https://files.pythonhosted.org/packages/3f/61/3291c2243ae0229e5bca5d19f4032cecad5dfb05a2557169d3a69dc0ba91/bcrypt-5.0.0-cp314-cp314t-win_
amd64.whl", hash = "sha256:92864f54fb48b4c718fc92a32825d0e42265a627f956bc0361fe869f1adc3e7d", size = 149310, upload-time = "2025-09-25T19:49:46.162Z"
},
    { url = "https://files.pythonhosted.org/packages/3e/89/4b01c52ae0c1a681d4021e5dd3e45b111a8fb47254a274fa9a378d8d834b/bcrypt-5.0.0-cp314-cp314t-win_
arm64.whl", hash = "sha256:dd19cf5184a90c873009244586396a6a884d591a5323f0e8a5922560718d4993", size = 143761, upload-time = "2025-09-25T19:49:47.345Z"
},
    { url = "https://files.pythonhosted.org/packages/84/29/6237f151fbfe295fe3e074ecc6d44228faa1e842a81f6d34a02937ee1736/bcrypt-5.0.0-cp38-abi3-macosx_
10_12_universal2.whl", hash = "sha256:fc746432b951e92b58317af8e0ca746efe93e66555f1b40888865ef5bf56446b", size = 494553, upload-time = "2025-09-25T19:4
9:49.006Z" },
    { url = "https://files.pythonhosted.org/packages/45/b6/4c1205dde5e464ea3bd88e8742e19f899c16fa8916fb8510a851fae985b5/bcrypt-5.0.0-cp38-abi3-manylin
ux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:c2388ca94ffee269b6038d48747f4ce8df0ffbea43f31abfa18ac72f0218effb", size = 275009, upload-ti
me = "2025-09-25T19:49:50.581Z" },
    { url = "https://files.pythonhosted.org/packages/3b/71/427945e6ead72ccffe77894b2655b695ccf14ae1866cd977e185d606dd2f/bcrypt-5.0.0-cp38-abi3-manylin
ux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:560ddb6ec730386e7b3b26b8b4c88197aaed924430e7b74666a586ac997249ef", size = 278029, upload-time
 = "2025-09-25T19:49:52.533Z" },
    { url = "https://files.pythonhosted.org/packages/17/72/c344825e3b83c5389a369c8a8e58ffe1480b8a699f46c127c34580c4666b/bcrypt-5.0.0-cp38-abi3-manylin
ux_2_28_aarch64.whl", hash = "sha256:d79e5c65dcc9af213594d6f7f1fa2c98ad3fc10431e7aa53c176b441943efbdd", size = 275907, upload-time = "2025-09-25T19:49
:54.709Z" },
    { url = "https://files.pythonhosted.org/packages/0b/7e/d4e47d2df1641a36d1212e5c0514f5291e1a956a7749f1e595c07a972038/bcrypt-5.0.0-cp38-abi3-manylin
ux_2_28_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:2b732e7d388fa22d48920baa267ba5d97cca38070b69c0e2d37087b381c681fd", size = 296500, upload-tim
e = "2025-09-25T19:49:56.013Z" },
    { url = "https://files.pythonhosted.org/packages/0f/c3/0ae57a68be2039287ec28bc463b82e4b8dc23f9d12c0be331f4782e19108/bcrypt-5.0.0-cp38-abi3-manylin
ux_2_28_x86_64.whl", hash = "sha256:0c8e093ea2532601a6f686edbc2c6b2ec24131ff5c52f7610dd64fa4553b5464", size = 278412, upload-time = "2025-09-25T19:49:
57.356Z" },
    { url = "https://files.pythonhosted.org/packages/45/2b/77424511adb11e6a99e3a00dcc7745034bee89036ad7d7e255a7e47be7d8/bcrypt-5.0.0-cp38-abi3-manylin
ux_2_34_aarch64.whl", hash = "sha256:5b1589f4839a0899c146e8892efe320c0fa096568abd9b95593efac50a87cb75", size = 275486, upload-time = "2025-09-25T19:49
:59.116Z" },
    { url = "https://files.pythonhosted.org/packages/43/0a/405c753f6158e0f3f14b00b462d8bca31296f7ecfc8fc8bc7919c0c7d73a/bcrypt-5.0.0-cp38-abi3-manylin
ux_2_34_x86_64.whl", hash = "sha256:89042e61b5e808b67daf24a434d89bab164d4de1746b37a8d173b6b14f3db9ff", size = 277940, upload-time = "2025-09-25T19:50:
00.869Z" },
    { url = "https://files.pythonhosted.org/packages/62/83/b3efc285d4aadc1fa83db385ec64dcfa1707e890eb42f03b127d66ac1b7b/bcrypt-5.0.0-cp38-abi3-musllin
ux_1_1_aarch64.whl", hash = "sha256:e3cf5b2560c7b5a142286f69bde914494b6d8f901aaa71e453078388a50881c4", size = 310776, upload-time = "2025-09-25T19:50:
02.393Z" },
    { url = "https://files.pythonhosted.org/packages/95/7d/47ee337dacecde6d234890fe929936cb03ebc4c3a7460854bbd9c97780b8/bcrypt-5.0.0-cp38-abi3-musllin
ux_1_1_x86_64.whl", hash = "sha256:f632fd56fc4e61564f78b46a2269153122db34988e78b6be8b32d28507b7eaeb", size = 312922, upload-time = "2025-09-25T19:50:0
4.232Z" },
    { url = "https://files.pythonhosted.org/packages/d6/3a/43d494dfb728f55f4e1cf8fd435d50c16a2d75493225b54c8d06122523c6/bcrypt-5.0.0-cp38-abi3-musllin
ux_1_2_aarch64.whl", hash = "sha256:801cad5ccb6b87d1b430f183269b94c24f248dddbbc5c1f78b6ed231743e001c", size = 341367, upload-time = "2025-09-25T19:50:
05.559Z" },
    { url = "https://files.pythonhosted.org/packages/55/ab/a0727a4547e383e2e22a630e0f908113db37904f58719dc48d4622139b5c/bcrypt-5.0.0-cp38-abi3-musllin
ux_1_2_x86_64.whl", hash = "sha256:3cf67a804fc66fc217e6914a5635000259fbbbb12e78a99488e4d5ba445a71eb", size = 359187, upload-time = "2025-09-25T19:50:0
6.916Z" },
    { url = "https://files.pythonhosted.org/packages/1b/bb/461f352fdca663524b4643d8b09e8435b4990f17fbf4fea6bc2a90aa0cc7/bcrypt-5.0.0-cp38-abi3-win32.w
hl", hash = "sha256:3abeb543874b2c0524ff40c57a4e14e5d3a66ff33fb423529c88f180fd756538", size = 153752, upload-time = "2025-09-25T19:50:08.515Z" },
    { url = "https://files.pythonhosted.org/packages/41/aa/4190e60921927b7056820291f56fc57d00d04757c8b316b2d3c0d1d6da2c/bcrypt-5.0.0-cp38-abi3-win_amd
64.whl", hash = "sha256:35a77ec55b541e5e583eb3436ffbbf53b0ffa1fa16ca6782279daf95d146dcd9", size = 150881, upload-time = "2025-09-25T19:50:09.742Z" },
    { url = "https://files.pythonhosted.org/packages/54/12/cd77221719d0b39ac0b55dbd39358db1cd1246e0282e104366ebbfb8266a/bcrypt-5.0.0-cp38-abi3-win_arm
64.whl", hash = "sha256:cde08734f12c6a4e28dc6755cd11d3bdfea608d93d958fffbe95a7026ebe4980", size = 144931, upload-time = "2025-09-25T19:50:11.016Z" },
    { url = "https://files.pythonhosted.org/packages/5d/ba/2af136406e1c3839aea9ecadc2f6be2bcd1eff255bd451dd39bcf302c47a/bcrypt-5.0.0-cp39-abi3-macosx_
10_12_universal2.whl", hash = "sha256:0c418ca99fd47e9c59a301744d63328f17798b5947b0f791e9af3c1c499c2d0a", size = 495313, upload-time = "2025-09-25T19:5
0:12.309Z" },
    { url = "https://files.pythonhosted.org/packages/ac/ee/2f4985dbad090ace5ad1f7dd8ff94477fe089b5fab2040bd784a3d5f187b/bcrypt-5.0.0-cp39-abi3-manylin
ux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:ddb4e1500f6efdd402218ffe34d040a1196c072e07929b9820f363a1fd1f4191", size = 275290, upload-ti
me = "2025-09-25T19:50:13.673Z" },
    { url = "https://files.pythonhosted.org/packages/e4/6e/b77ade812672d15cf50842e167eead80ac3514f3beacac8902915417f8b7/bcrypt-5.0.0-cp39-abi3-manylin
ux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:7aeef54b60ceddb6f30ee3db090351ecf0d40ec6e2abf41430997407a46d2254", size = 278253, upload-time
 = "2025-09-25T19:50:15.089Z" },
    { url = "https://files.pythonhosted.org/packages/36/c4/ed00ed32f1040f7990dac7115f82273e3c03da1e1a1587a778d8cea496d8/bcrypt-5.0.0-cp39-abi3-manylin
ux_2_28_aarch64.whl", hash = "sha256:f0ce778135f60799d89c9693b9b398819d15f1921ba15fe719acb3178215a7db", size = 276084, upload-time = "2025-09-25T19:50
:16.699Z" },
    { url = "https://files.pythonhosted.org/packages/e7/c4/fa6e16145e145e87f1fa351bbd54b429354fd72145cd3d4e0c5157cf4c70/bcrypt-5.0.0-cp39-abi3-manylin
ux_2_28_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:a71f70ee269671460b37a449f5ff26982a6f2ba493b3eabdd687b4bf35f875ac", size = 297185, upload-tim
e = "2025-09-25T19:50:18.525Z" },
    { url = "https://files.pythonhosted.org/packages/24/b4/11f8a31d8b67cca3371e046db49baa7c0594d71eb40ac8121e2fc0888db0/bcrypt-5.0.0-cp39-abi3-manylin
ux_2_28_x86_64.whl", hash = "sha256:f8429e1c410b4073944f03bd778a9e066e7fad723564a52ff91841d278dfc822", size = 278656, upload-time = "2025-09-25T19:50:
19.809Z" },
    { url = "https://files.pythonhosted.org/packages/ac/31/79f11865f8078e192847d2cb526e3fa27c200933c982c5b2869720fa5fce/bcrypt-5.0.0-cp39-abi3-manylin
ux_2_34_aarch64.whl", hash = "sha256:edfcdcedd0d0f05850c52ba3127b1fce70b9f89e0fe5ff16517df7e81fa3cbb8", size = 275662, upload-time = "2025-09-25T19:50
:21.567Z" },
    { url = "https://files.pythonhosted.org/packages/d4/8d/5e43d9584b3b3591a6f9b68f755a4da879a59712981ef5ad2a0ac1379f7a/bcrypt-5.0.0-cp39-abi3-manylin
ux_2_34_x86_64.whl", hash = "sha256:611f0a17aa4a25a69362dcc299fda5c8a3d4f160e2abb3831041feb77393a14a", size = 278240, upload-time = "2025-09-25T19:50:
23.305Z" },
    { url = "https://files.pythonhosted.org/packages/89/48/44590e3fc158620f680a978aafe8f87a4c4320da81ed11552f0323aa9a57/bcrypt-5.0.0-cp39-abi3-musllin
ux_1_1_aarch64.whl", hash = "sha256:db99dca3b1fdc3db87d7c57eac0c82281242d1eabf19dcb8a6b10eb29a2e72d1", size = 311152, upload-time = "2025-09-25T19:50:
24.597Z" },
    { url = "https://files.pythonhosted.org/packages/5f/85/e4fbfc46f14f47b0d20493669a625da5827d07e8a88ee460af6cd9768b44/bcrypt-5.0.0-cp39-abi3-musllin
ux_1_1_x86_64.whl", hash = "sha256:5feebf85a9cefda32966d8171f5db7e3ba964b77fdfe31919622256f80f9cf42", size = 313284, upload-time = "2025-09-25T19:50:2
6.268Z" },
    { url = "https://files.pythonhosted.org/packages/25/ae/479f81d3f4594456a01ea2f05b132a519eff9ab5768a70430fa1132384b1/bcrypt-5.0.0-cp39-abi3-musllin
ux_1_2_aarch64.whl", hash = "sha256:3ca8a166b1140436e058298a34d88032ab62f15aae1c598580333dc21d27ef10", size = 341643, upload-time = "2025-09-25T19:50:
28.02Z" },
    { url = "https://files.pythonhosted.org/packages/df/d2/36a086dee1473b14276cd6ea7f61aef3b2648710b5d7f1c9e032c29b859f/bcrypt-5.0.0-cp39-abi3-musllin
ux_1_2_x86_64.whl", hash = "sha256:61afc381250c3182d9078551e3ac3a41da14154fbff647ddf52a769f588c4172", size = 359698, upload-time = "2025-09-25T19:50:3
1.347Z" },
    { url = "https://files.pythonhosted.org/packages/c0/f6/688d2cd64bfd0b14d805ddb8a565e11ca1fb0fd6817175d58b10052b6d88/bcrypt-5.0.0-cp39-abi3-win32.w
hl", hash = "sha256:64d7ce196203e468c457c37ec22390f1a61c85c6f0b8160fd752940ccfb3a683", size = 153725, upload-time = "2025-09-25T19:50:34.384Z" },
    { url = "https://files.pythonhosted.org/packages/9f/b9/9d9a641194a730bda138b3dfe53f584d61c58cd5230e37566e83ec2ffa0d/bcrypt-5.0.0-cp39-abi3-win_amd
64.whl", hash = "sha256:64ee8434b0da054d830fa8e89e1c8bf30061d539044a39524ff7dec90481e5c2", size = 150912, upload-time = "2025-09-25T19:50:35.69Z" },
    { url = "https://files.pythonhosted.org/packages/27/44/d2ef5e87509158ad2187f4dd0852df80695bb1ee0cfe0a684727b01a69e0/bcrypt-5.0.0-cp39-abi3-win_arm
64.whl", hash = "sha256:f2347d3534e76bf50bca5500989d6c1d05ed64b440408057a37673282c654927", size = 144953, upload-time = "2025-09-25T19:50:37.32Z" },
]

[[package]]
name = "certifi"
version = "2026.7.22"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a3/c2/24167ea9858356b47a87a50d39908bfdb72ceeefe0041586e704e5376b3a/certifi-2026.7.22.tar.gz",
```

```
 hash = "sha256:741e2c3b351ddf169a738da9f2c048608ff7f2c5cc02f1ebc6b118bb090d5d55", size = 138112, upload-time = "2026-07-22T03:35:12.644Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/0b/a7/71ac2cff56fec219ed242bb11b8efb69fcc4bec75db06fb7bfe35de520e6/certifi-2026.7.22-py3-none-any.whl", hash = "sha256:62f22742b58a1a33014a2b6b706588a8d7e2a88ae7bd1a6ebe8c992928483775", size = 136983, upload-time = "2026-07-22T03:35:11.276Z" },
]

[[package]]
name = "cffi"
version = "2.1.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "pycparser", marker = "implementation_name != 'PyPy'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/9e/ef/008a1939e372c06329a3fce4279c02f328488f3526744906eeec3da7ad5f/cffi-2.1.1.tar.gz", hash = "sha256:dd31f52ea1086513bb9df30f8fcee9b8918323ae067a3d5b78bc826a000712be", size = 530807, upload-time = "2026-08-03T21:21:18.939Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/10/69/43965eccfdead3b9220015fd1320e117be8c6ed01a62ffab76eeb752f5d5/cffi-2.1.1-cp312-cp312-macosx_10_15_x86_64.whl", hash = "sha256:c8c69575568085ba0b1b10c0249d779a214aea6f6522e949a0fc9fb0fcb449d0", size = 184821, upload-time = "2026-08-03T21:19:44.887Z" },
    { url = "https://files.pythonhosted.org/packages/54/7d/16e5a096677b5e313ca80cd5e5170efa3ea44624a82bb111925522da64b1/cffi-2.1.1-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:f81b3b8f3d4e343550fa4baa0e479bba9f2d29ce9c2e9b51d1ce1718d7442fcf", size = 184719, upload-time = "2026-08-03T21:19:46.129Z" },
    { url = "https://files.pythonhosted.org/packages/56/e6/8941622732edec876dd17d0453dce07317ae96db34f2ec1436c9d3785986/cffi-2.1.1-cp312-cp312-manylinux1_i686.manylinux2014_i686.manylinux_2_17_i686.manylinux_2_5_i686.whl", hash = "sha256:811bd1e21d32de12efca32393a0ab3f5133b54fce9bd44b8bd77ab07da14bf6a", size = 214799, upload-time = "2026-08-03T21:19:47.218Z" },
    { url = "https://files.pythonhosted.org/packages/44/de/f98430906df1545ffde0d543dd124a7a439bc2cd32b36b9c53f805df7333/cffi-2.1.1-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:68e62fe11f30d5ca8289242866f0a5291402d8529ca2178ab8afc5c9694ae890", size = 222389, upload-time = "2026-08-03T21:19:48.331Z" },
    { url = "https://files.pythonhosted.org/packages/6a/5b/717f1526b9957b34456313c31645c5b82b8fb5c3fe9e4752999be7128bfc/cffi-2.1.1-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:4a7c934f7360e8cd64fe9efadcbd10c7c6364f531e432b9a4bf5ccbc9e0e8b50", size = 210249, upload-time = "2026-08-03T21:19:49.543Z" },
    { url = "https://files.pythonhosted.org/packages/64/b3/f8aa4f3e34986c7e4ec45072d1b1b9dd295b6b18007b45518d79726dd725/cffi-2.1.1-cp312-cp312-manylinux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:3143d81e29e1e20a9ce10901ec369012947876596f75a222235965f2b7ae832e", size = 208775, upload-time = "2026-08-03T21:19:50.918Z" },
    { url = "https://files.pythonhosted.org/packages/b1/db/dceb9dd5b231e1da801793f8acc9f3c52a7e1afe40bb1aae37e02b0faad5/cffi-2.1.1-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:c1453022f490d2459a11819d83ad1d586e9ff65a12ac3e705ffebd46d3685dcf", size = 221822, upload-time = "2026-08-03T21:19:52.054Z" },
    { url = "https://files.pythonhosted.org/packages/a0/d2/6cd24ae3be000a634109c247d1475d62e5616d0dc78c82770942ec384248/cffi-2.1.1-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:208f941bb9d18e768138677f0a6d2ce01f590df56043dda1df1535ac57c88517", size = 225232, upload-time = "2026-08-03T21:19:53.109Z" },
    { url = "https://files.pythonhosted.org/packages/cb/52/3fa190537004dd7f0ab860a6dc7c0175b8667f68d1e618a46f5498d30250/cffi-2.1.1-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:210019b6c7cf07f081b4c54635c8cf744377001350e29cc0f81c4377b4797735", size = 223597, upload-time = "2026-08-03T21:19:54.515Z" },
    { url = "https://files.pythonhosted.org/packages/80/fb/0bb75b7039588c074b37ae99f40d9bfddf990ecb2fbc346ebccd2e56b9be/cffi-2.1.1-cp312-cp312-win32.whl", hash = "sha256:046bfc24911b37851ee1b51aab8bffe713d89c68c6a057b09484ce9fd5f69b4e", size = 175292, upload-time = "2026-08-03T21:19:55.566Z" },
    { url = "https://files.pythonhosted.org/packages/d9/79/615cc094e2fb508cade7de88d3b4f6c4ec2bab695c97bce9153dc65aadf5/cffi-2.1.1-cp312-cp312-win_amd64.whl", hash = "sha256:f53e442b08449d42821fa4a4fba000095af9f62742a500f978a9f557ec44339a", size = 185919, upload-time = "2026-08-03T21:19:56.89Z" },
    { url = "https://files.pythonhosted.org/packages/70/c6/d0ea84713fe46b243a436a18fcd47d639732747e21635c8a27191b06dc30/cffi-2.1.1-cp312-cp312-win_arm64.whl", hash = "sha256:7bde5e4cc5c10140859842b9d383af292b22639a4dffb725314baf45968cef80", size = 180093, upload-time = "2026-08-03T21:19:58.155Z" },
    { url = "https://files.pythonhosted.org/packages/9d/f4/035513d4117049066b4779dc3b7c0c0fdad175fa13731c9f4003f1cd1478/cffi-2.1.1-cp313-cp313-ios_13_0_arm64_iphoneos.whl", hash = "sha256:b5bdfd1c873d4e093aabc0ca84c4ca6dbc4f752afb5c86f146d9742580c9da2e", size = 194248, upload-time = "2026-08-03T21:19:59.399Z" },
    { url = "https://files.pythonhosted.org/packages/76/af/2aeb4dbb5fc41a04161ae9ff1518de7cec08e164f44a8ce6a4cf7fd2cd1d/cffi-2.1.1-cp313-cp313-ios_13_0_arm64_iphonesimulator.whl", hash = "sha256:31348097ff5bbe827ccc41795d4dd099d9f0625e7def00ee653c137a490c2a6c", size = 196908, upload-time = "2026-08-03T21:20:00.746Z" },
    { url = "https://files.pythonhosted.org/packages/a7/46/2e5fdde8555706dd98139a910ca11be02809f3f605ce956f655d0214e100/cffi-2.1.1-cp313-cp313-macosx_10_15_x86_64.whl", hash = "sha256:9d2055050ea716bd38b7f7f1579c275386646b4894c155a3e2f3cd62ed41b7c6", size = 184805, upload-time = "2026-08-03T21:20:02.02Z" },
    { url = "https://files.pythonhosted.org/packages/55/41/4c7042f317b9217502988f0873af87e16ad606dc20f84e546e3e6ce9764c/cffi-2.1.1-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:19ee6127ee34de7d83ce3d371ebc5ed91addbdcc39f9ab15ce4eb35a4e534971", size = 184764, upload-time = "2026-08-03T21:20:03.141Z" },
    { url = "https://files.pythonhosted.org/packages/43/1f/1c3d90d91811c8f86ced9ed637956c54bfe5b79ca98fe976d7f8c8979f6b/cffi-2.1.1-cp313-cp313-manylinux1_i686.manylinux2014_i686.manylinux_2_17_i686.manylinux_2_5_i686.whl", hash = "sha256:6a8dddef476fab96d066d578fc88526767b836ab5ab21754e1d5bf3879c31c7c", size = 214722, upload-time = "2026-08-03T21:20:04.377Z" },
    { url = "https://files.pythonhosted.org/packages/37/6f/3b5ce4c3b2192d250f04908f2bfd91ef34552ec8f7716a5d4abdb8d67bb2/cffi-2.1.1-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:f16c709686a78c727bbbf059f92b0bf41c6fc60deec706d2dc19f529175a6125", size = 222369, upload-time = "2026-08-03T21:20:05.544Z" },
    { url = "https://files.pythonhosted.org/packages/02/10/4b3c75dde3d9663c9e02ba05c2668b954f671d4bbe346413ca8c696b295a/cffi-2.1.1-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:fcd22650c908d7b7da162bbfaab594a1227a15d1643a98c68b122ac642fa2264", size = 210175, upload-time = "2026-08-03T21:20:06.75Z" },
    { url = "https://files.pythonhosted.org/packages/df/62/14f74b9543e605d17701dc797b815958b8bb70b7624ce1b832ddad48ed6c/cffi-2.1.1-cp313-cp313-manylinux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:aa9511c62d14da7aacc9b4bf51f3f697a621e83b2d6919008243c3aad168eea3", size = 208670, upload-time = "2026-08-03T21:20:08.04Z" },
    { url = "https://files.pythonhosted.org/packages/95/95/86342356ff5953b3fb06f7ef7c5bee212d45e770abc7218d451b9148313c/cffi-2.1.1-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:a931079504ecc49efed7744c476a5c343a92fabf66dec2db95edb1b2fdc770e2", size = 221824, upload-time = "2026-08-03T21:20:09.274Z" },
    { url = "https://files.pythonhosted.org/packages/eb/ff/7b3429ff53aafe931ed8a5fc69f481bbef7ba6de87ddcbb63d08f483f613/cffi-2.1.1-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:a2d7755bef5a12ed488f4ef1f1b69ee9191d7396083b755a5d2295f6edb4768b", size = 225148, upload-time = "2026-08-03T21:20:10.7Z" },
    { url = "https://files.pythonhosted.org/packages/34/34/a95870b9221e09cf4f2ce3178b1a210abdfe63a1bd357da940418d7b8d15/cffi-2.1.1-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:e0bcb7e0f677f543555d2adff3bf19c05f66cdb4796e5ff602442ab2fe3c4ef7", size = 223564, upload-time = "2026-08-03T21:20:12.165Z" },
    { url = "https://files.pythonhosted.org/packages/70/ea/839b50531021a647fb5e929f72cf97bc1ff702b5472166164b5b6e76b851/cffi-2.1.1-cp313-cp313-win32.whl", hash = "sha256:334644fbac4eff73d985a17a91226df55d0f394160c4cfb880e084c8f7161cac", size = 175263, upload-time = "2026-08-03T21:20:13.559Z" },
    { url = "https://files.pythonhosted.org/packages/60/a6/8b149b2c3f2e11aaa1618ef64500b45f50f22c57a977a4dff1aff1f91042/cffi-2.1.1-cp313-cp313-win_amd64.whl", hash = "sha256:1aa5645c30469b09530c4ebca77ebf8f17618293c58f8549cb1a543a50236e7d", size = 185688, upload-time = "2026-08-03T21:20:14.69Z" },
    { url = "https://files.pythonhosted.org/packages/01/9a/11f687cb39d6a3504060d5242f04f48c735afb4d3d533958a20594890cb2/cffi-2.1.1-cp313-cp313-win_arm64.whl", hash = "sha256:63bbfd5ded17c4840ac07cd8f1c21ba9d9708141f840b324f422f41b207e3973", size = 180078, upload-time = "2026-08-03T21:20:15.917Z" },
    { url = "https://files.pythonhosted.org/packages/d3/7b/d6bbf82b8b96e7391438898c42f5bd96dd02030fd5b64937d248220003e2/cffi-2.1.1-cp314-cp314-ios_13_0_arm64_iphoneos.whl", hash = "sha256:7dbb61fe3a7699468030f71bbe5f8a0e326a151daa91beb11a6fc1f980c55e1c", size = 194064, upload-time = "2026-08-03T21:20:17.148Z" },
    { url = "https://files.pythonhosted.org/packages/94/e6/bcc91b283be94735e268487a054004f0aa19947b6348fa367db53230abc8/cffi-2.1.1-cp314-cp314-ios_13_0_arm64_iphonesimulator.whl", hash = "sha256:f24fb43132a4c6b4cb4eb029492919b2db645be6808d738f244fd146c03c32cb", size = 196720, upload-time = "2026-08-03T21:20:18.268Z" },
    { url = "https://files.pythonhosted.org/packages/d9/99/c4b0c17cacdc9c3b8f280026286a9826d6a208c0f047591a3c3ce99b91fd/cffi-2.1.1-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:d28630f5854ab07ab1fd4aba756de52326c82e6be15d414b12793f1975048b54", size = 184964, upload-time = "2026-08-03T21:20:19.708Z" },
    { url = "https://files.pythonhosted.org/packages/b3/a9/9db617d05d7367c1ad0ab00b3aa6e6f9281edd689b4ee9ea0e5a84e89c97/cffi-2.1.1-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:661c298b4821edebead0c91edd2b00374d67ad7c5a1f7a91d4442633b79d6a72", size = 184962, upload-time = "2026-08-03T21:20:20.833Z" },
    { url = "https://files.pythonhosted.org/packages/67/b8/b42132ca113dc567d37684437b46ca1dafc885902b02a110a02d5b511857/cffi-2.1.1-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:58acb8ab8e295e6c5ea12f888cbb13cf21511ef2a3303a23f4325c29d17fe5c1", size = 222328, upload-time = "2026-08-03T21:20:22.118Z" },
    { url = "https://files.pythonhosted.org/packages/80/10/c5c0cbf0a657aecf59ef511409734230bf556f05a0d6c9eed7aa5c0a0166/cffi-2.1.1-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:456a61fa52d579ebf9df2e9552ead5129855dbaff6c1e5a9b1bc408809bdc062", size = 209985, upload-ti
```

me = "2026-08-03T21:20:23.401Z" },
    { url = "https://files.pythonhosted.org/packages/d5/6c/bfa0b87b03b9238148beca990292843c9396ba069b54496596594173de7b/cffi-2.1.1-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:a4f00aa42f75d6e4595e8866e748cc1705adc0cddfeb2ca86d0d03993d63ba03", size = 208530, upload-time = "2026-08-03T21:20:24.628Z" },
    { url = "https://files.pythonhosted.org/packages/e9/02/4e7d553a7ac4b4238b38b3c1b80d486e9d4436f8d2acbf87a0997fe3f402/cffi-2.1.1-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:b0431303acaea1089ad4b3e9ce4e6518193def1118d4073ca848635ee4ea2e96", size = 221525, upload-time = "2026-08-03T21:20:25.758Z" },
    { url = "https://files.pythonhosted.org/packages/82/1d/a4aaf9babd75acb4d5f223bff71533bee748dd770a382619a798960ee9ba/cffi-2.1.1-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:64faea20f4e2613363a1a9b9c7dd73058f3ecd00133a511e72ad7c511658f527", size = 225053, upload-time = "2026-08-03T21:20:26.985Z" },
    { url = "https://files.pythonhosted.org/packages/81/10/5dc0e7bdd18e22107054288283380fc97a06ae3f1656a106908d666a3c88/cffi-2.1.1-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:5c58fe613dc5e5336357eff555824a314d8e43282600435c8d1cb6a7a2fedd13", size = 223213, upload-time = "2026-08-03T21:20:28.277Z" },
    { url = "https://files.pythonhosted.org/packages/0b/e9/d0061c364cde06ee43168a0d076ac1da512cbc380d44767b844ba34fe2b6/cffi-2.1.1-cp314-cp314-win32.whl", hash = "sha256:1a18a57b58cfb21fc28d72e876acf10eaed67a1ed96226f92af4df681d571c4c", size = 177682, upload-time = "2026-08-03T21:20:44.288Z" },
    { url = "https://files.pythonhosted.org/packages/a7/06/1c3e01e3ba14c39f6d10bfbac52753b7e22259e38088e5cfe1d704918690/cffi-2.1.1-cp314-cp314-win_amd64.whl", hash = "sha256:3222ba5d678f80a030e6afbcc33dc1ae5cb45facabb61cee2c7016b8432fde48", size = 187949, upload-time = "2026-08-03T21:20:45.623Z" },
    { url = "https://files.pythonhosted.org/packages/87/5b/da4e39efe18eeb89cf580ea9cfc66b6a7c3eadb808fc0cc1d3a295cb5a5d/cffi-2.1.1-cp314-cp314-win_arm64.whl", hash = "sha256:ab36d55f9ed2d067327667c2fea18dda018eb628dd6347aa01dda6cf1f5d3836", size = 182947, upload-time = "2026-08-03T21:20:46.955Z" },
    { url = "https://files.pythonhosted.org/packages/23/59/40338bf421c5accea1d45158170c87006ef1cd371b05c077e76476949728/cffi-2.1.1-cp314-cp314t-macosx_10_15_x86_64.whl", hash = "sha256:7750c6449dff7864bb9bb27ddfb0267756189201a3afc911d82b3caacd70dfc3", size = 188504, upload-time = "2026-08-03T21:20:29.495Z" },
    { url = "https://files.pythonhosted.org/packages/7d/47/5ecf1023850036e674c77ec4de86182d309ae344e39e7cba984b7df5d647/cffi-2.1.1-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:0beceaabe56af686895136a2de78db54ecd8e4046b236b8fd6d6cb61389e9bf2", size = 188259, upload-time = "2026-08-03T21:20:31.291Z" },
    { url = "https://files.pythonhosted.org/packages/2a/9c/92934c3bea9f785b23eba304538c0b4d37a2a96d2431eb3a1bc87a11aa19/cffi-2.1.1-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:49cbc70e6542d4ccccb936558d1064a8012541e78f821f955cff24e357776c94", size = 223864, upload-time = "2026-08-03T21:20:32.571Z" },
    { url = "https://files.pythonhosted.org/packages/4d/45/ba4c93527bc38616a8bd36488acb69a2212d60486794f0c1f318949bbb76/cffi-2.1.1-cp314-cp314t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:e2d65b31f36619cda3999b78b2aa9632e76b78448e7a56fc4240824200e7c4fc", size = 211538, upload-time = "2026-08-03T21:20:33.808Z" },
    { url = "https://files.pythonhosted.org/packages/80/e9/b6ef565e452acb932fb0cb5443f44a78efbd1233e566f02b5a83855e9115/cffi-2.1.1-cp314-cp314t-manylinux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:28907ab9bfb6aa13184cfc17c6b8e1023c5ab6fd7076d8c20a35e59fe04f8f29", size = 210688, upload-time = "2026-08-03T21:20:34.974Z" },
    { url = "https://files.pythonhosted.org/packages/9a/95/eff5f0cee78d2eabc7eebffec40d3fc1876b5f3c95582e018bb4b99601f2/cffi-2.1.1-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:51b31d1c98274844cfd7838ce00bfc27c7423a4dc00fc0772fc3331c2cc90676", size = 223803, upload-time = "2026-08-03T21:20:36.564Z" },
    { url = "https://files.pythonhosted.org/packages/fa/01/579d39fb8bef00a335a23d83757b44feb24cd6345a2c451b64cb67b9c362/cffi-2.1.1-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:5e7cecbaadb83884793e05828cee59b210b24583b9c7425d0ba6a754fe22eb4e", size = 226763, upload-time = "2026-08-03T21:20:37.816Z" },
    { url = "https://files.pythonhosted.org/packages/8d/b0/0b44f47c60b01b57b6e2bbd92343f13a85a1d93bc46ccf6e47e244acd99c/cffi-2.1.1-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:25792eac27877609e7bb06d42ff88278a6624fff2ba9bbb523c09616b117e80f", size = 225688, upload-time = "2026-08-03T21:20:38.959Z" },
    { url = "https://files.pythonhosted.org/packages/eb/d2/3b7176cb570a1d3e27faf67b72f591af508036e0d8b2be2ef9af9e8c84bb/cffi-2.1.1-cp314-cp314t-win32.whl", hash = "sha256:8ef53b2de9bcb9197d31854256575d59dbac0cba72ac627bb291ef5eceb74be4", size = 182868, upload-time = "2026-08-03T21:20:40.388Z" },
    { url = "https://files.pythonhosted.org/packages/56/78/31f00c1bcd97c9bbf55f1bfdf5bc809a5de8887473e90bb9960dca825e80/cffi-2.1.1-cp314-cp314t-win_amd64.whl", hash = "sha256:616f097f2fe415bc92a247f02e11f634e1f9e9a83d327e3c915c15089c87869e", size = 194104, upload-time = "2026-08-03T21:20:41.725Z" },
    { url = "https://files.pythonhosted.org/packages/7b/1b/58496f2ed0a35de575250c02a43ab3cc2c04d494a88fed31c1cabc0fd176/cffi-2.1.1-cp314-cp314t-win_arm64.whl", hash = "sha256:ad2c86c495b899d862ea0f4b42891b8713a3bd45dd4105c7fd51c2a72f39f3a5", size = 186402, upload-time = "2026-08-03T21:20:43.042Z" },
    { url = "https://files.pythonhosted.org/packages/c1/8f/9ebe220eab48a093d1a5a5e339ab0dc7316eef3bb04d63c42f0251b61f50/cffi-2.1.1-cp315-cp315-ios_13_0_arm64_iphoneos.whl", hash = "sha256:dddad92b554513a31f272570678ba307fb9f618f05e3d4a5eacafff9eae03e1d", size = 194043, upload-time = "2026-08-03T21:20:48.179Z" },
    { url = "https://files.pythonhosted.org/packages/ff/69/844bad3ece306c4782c2ecb93597035b6690d48704b803914c199da1e8b3/cffi-2.1.1-cp315-cp315-ios_13_0_arm64_iphonesimulator.whl", hash = "sha256:da0e573f9f97159390c89d9f1a9e41908b66d408cc5b58d08cf3847d844c531b", size = 196737, upload-time = "2026-08-03T21:20:49.457Z" },
    { url = "https://files.pythonhosted.org/packages/1b/8a/af668013284634733f02d683458a0728739c7d6ddb5e14cb0c20832266fe/cffi-2.1.1-cp315-cp315-macosx_10_15_x86_64.whl", hash = "sha256:fb92203a88b3d3053034db775110081c49d28be6551923805e039924093761e4", size = 184933, upload-time = "2026-08-03T21:20:50.639Z" },
    { url = "https://files.pythonhosted.org/packages/0c/75/2f5207ff6d1a613133b23a5203cc0c2a628313b5eb3974d7956ae3c57950/cffi-2.1.1-cp315-cp315-macosx_11_0_arm64.whl", hash = "sha256:2ae64be792b8966f2c69538199728b290e34726562896df1e5dc8ffd8d8188e8", size = 185002, upload-time = "2026-08-03T21:20:52.173Z" },
    { url = "https://files.pythonhosted.org/packages/e2/31/9e1313b0a6e30e91b3b3d3fff51ae99c857c07738e3afcce1f7334e1b7ab/cffi-2.1.1-cp315-cp315-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:507a24c282e0f42f8ed737cf048572cbf580468da5555764a8331735e9c736b6", size = 222271, upload-time = "2026-08-03T21:20:53.462Z" },
    { url = "https://files.pythonhosted.org/packages/50/e3/f6234a833e6e08c7007003074723c406559eecf9b48dfc97471e5a8eb7a0/cffi-2.1.1-cp315-cp315-manylinux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:246fa40ce8645a614ff682e0b70f37134e460eaf93a775e0cbe3cca585a67a80", size = 209919, upload-time = "2026-08-03T21:20:54.783Z" },
    { url = "https://files.pythonhosted.org/packages/0d/fc/5f74e293fced6edb51af3a46c4ccf6c23c9943774ecb375ddbd522c76add/cffi-2.1.1-cp315-cp315-manylinux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:471cee653ae88de62096552e6d24ccb4a5adb8c8c9f10b5054d0122c15bf2779", size = 208529, upload-time = "2026-08-03T21:20:56.066Z" },
    { url = "https://files.pythonhosted.org/packages/44/16/29e6d01b388bef055ecd6ca8244b3f4d336bd09e92d5d892187b9601084e/cffi-2.1.1-cp315-cp315-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:aeae0e330c9f6acd681f647d46cefd30c29f93e3392882e792e82080c9691399", size = 221630, upload-time = "2026-08-03T21:20:57.336Z" },
    { url = "https://files.pythonhosted.org/packages/a4/18/fa7f1f6857d5eb88a4ca99ffcbfb7c387a287ccc154c64a73e86314745d7/cffi-2.1.1-cp315-cp315-musllinux_1_2_aarch64.whl", hash = "sha256:42a494cee34437f05546455144f2b5d9ac09b1face62bcfce597d2e521066688", size = 225134, upload-time = "2026-08-03T21:20:58.675Z" },
    { url = "https://files.pythonhosted.org/packages/e0/9f/e8e3dfa04a1b4c241f8c91faacad872b4d4efd051d49764ad4e2fd4b9fea/cffi-2.1.1-cp315-cp315-musllinux_1_2_x86_64.whl", hash = "sha256:cc572dace3f60ef98d7b12ff411d20f5362feb31a0439eab0085bbfd349982d7", size = 223197, upload-time = "2026-08-03T21:20:59.968Z" },
    { url = "https://files.pythonhosted.org/packages/f8/7e/8debeb04f1ab9fe2a6963964cd6f1aaf7192627b83926586a6a4e089c9fa/cffi-2.1.1-cp315-cp315-win32.whl", hash = "sha256:4f42141fc14250de6dde5ee7ea4432be017252d91f19c5ad043c084cea629cac", size = 177683, upload-time = "2026-08-03T21:21:14.901Z" },
    { url = "https://files.pythonhosted.org/packages/e0/31/5158704cc474ab65c1647932e88be78dc0873f47130e253be38bcaf13d01/cffi-2.1.1-cp315-cp315-win_amd64.whl", hash = "sha256:e6e8cff14d6fb0be70a09c0bdc58096f501952d04624ebf867e0e56da2df8960", size = 187897, upload-time = "2026-08-03T21:21:16.108Z" },
    { url = "https://files.pythonhosted.org/packages/cc/4b/b3a2da8570c704ffc0f9762cdc3ec0f02c8573798e0b5cf7f11c82bbb70f/cffi-2.1.1-cp315-cp315-win_arm64.whl", hash = "sha256:27350daa11d4f10c540e6e89dada4c54feb7256ad03e9a4dc075ebad7ba360d1", size = 182935, upload-time = "2026-08-03T21:21:17.271Z" },
    { url = "https://files.pythonhosted.org/packages/d0/ef/5443574510a1207e6f6bc38ba6e1f1de36cb48fef07b2728bb896a21f430/cffi-2.1.1-cp315-cp315t-macosx_10_15_x86_64.whl", hash = "sha256:c26608d2222fb1e94487e4a387d85f13eb55d5ed725cb25a0c589ac4ee60e7bc", size = 188464, upload-time = "2026-08-03T21:21:01.163Z" },
    { url = "https://files.pythonhosted.org/packages/7e/ae/a56fa8c4686ad50e148fcbc8d3ae0d03915ff5c30d795058988c24118cef/cffi-2.1.1-cp315-cp315t-macosx_11_0_arm64.whl", hash = "sha256:4be96343e422f2dfcd12ab5c9f5aebe03f82f737c6bffeca6830b3875cb44aab", size = 188262, upload-time = "2026-08-03T21:21:02.382Z" },
    { url = "https://files.pythonhosted.org/packages/53/b2/6187f46f2912276a3ae284076109cc5c8680482f11f766ccf26db4a86427/cffi-2.1.1-cp315-cp315t-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:937c0052c05a31ca1daf18de3158eed4dbfcb9cc107adbea227728d647be701e", size = 223779, upload-time = "2026-08-03T21:21:03.553Z" },
    { url = "https://files.pythonhosted.org/packages/8a/f6/c3ad28bd19f77047a03084424fbd4cbe997303267c14423737324be0385d/cffi-2.1.1-cp315-cp315t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:df423d40ee8654634421812bc3b196da3f9bd7d32929da813f8394c4348a5358", size = 211520, upload-time = "2026-08-03T21:21:04.863Z" },
    { url = "https://files.pythonhosted.org/packages/a0/cd/ccac9013a5bd9fd764de118674ab9c805b5ca10c19270d90ee273f8b2240/cffi-2.1.1-cp315-cp315t-manylinux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:a730a083190634c65cca36ba5f489531576ebd79bcd5c8e172130f6453127231", size = 210673, upload-time = "2026-08-03T21:21:06.223Z" },
    { url = "https://files.pythonhosted.org/packages/52/86/2976131c639aead931c5bee5aba67e4b09fbeb8018b6f282f70803f923a7/cffi-2.1.1-cp315-cp315t-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:363e05fa78e15116c3c32c210ee36884fd6b9afa6d440e47112c3bd511d64cb6", size = 223835, upload-time = "2026-08-03T21:21:07.539Z" },
    { url = "https://files.pythonhosted.org/packages/ac/0c/33a7aeab2f9c76918c52e084beb39c570db3588133412929e8ec06fab90b/cffi-2.1.1-cp315-cp315t-musllinux_1_2_aarch64.whl", hash = "sha256:770de9db11e84213beec501cfcaa013b019820ca881e03344dea5844f7876d94", size = 226705, upload-time = "2026-08-03T21:21

```
:08.774Z" },
    { url = "https://files.pythonhosted.org/packages/e3/26/2cde30fdde421130bfc18f70395731a6e6b2053c6a1978a5258ff04e72fa/cffi-2.1.1-cp315-cp315t-musllinux_1_2_x86_64.whl", hash = "sha256:7da0c5eff80f0197f3b3d1232ec5a682a9325f4ae9016a78f5f5ca35f9ced1f5", size = 225539, upload-time = "2026-08-03T21:21:09.911Z" },
    { url = "https://files.pythonhosted.org/packages/6d/cd/a361394c94b2129d604bb846f624a8e88255a3ee33129c434a00d715e64f/cffi-2.1.1-cp315-cp315t-win32.whl", hash = "sha256:06c72bb76605a4b0cd0aad6930b69d4baf7dd5d806cfc409b824191099700e66", size = 182707, upload-time = "2026-08-03T21:21:11.226Z" },
    { url = "https://files.pythonhosted.org/packages/9b/b5/ba2b299993c26577d529b6ae29841f9e15b9fcf004d65f423f4fcf94ade9/cffi-2.1.1-cp315-cp315t-win_amd64.whl", hash = "sha256:d9c275eaacd24aa73f94ffd6de08fc3f932424d8b6c376f4bed7cde376fe7bc3", size = 193772, upload-time = "2026-08-03T21:21:12.39Z" },
    { url = "https://files.pythonhosted.org/packages/aa/29/35e016098c814cd93de9cd320c66b5bfba14dc6ecedd3cb518fa7c408c69/cffi-2.1.1-cp315-cp315t-win_arm64.whl", hash = "sha256:d18e5ac0f2f03f4f518d3e23db0f0cad7faa1da8620e9c09461d443bbf6e6692", size = 186360, upload-time = "2026-08-03T21:21:13.636Z" },
]

[[package]]
name = "cfgv"
version = "3.5.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/4e/b5/721b8799b04bf9afe054a3899c6cf4e880fcf8563cc71c15610242490a0c/cfgv-3.5.0.tar.gz", hash = "sha256:d5b1034354820651caa73ede66a6294d6e95c1b00acc5e9b098e917404669132", size = 7334, upload-time = "2025-11-19T20:55:51.612Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/db/3c/33bac158f8ab7f89b2e59426d5fe2e4f63f7ed25df84c036890172b412b5/cfgv-3.5.0-py2.py3-none-any.whl", hash = "sha256:a8dc6b26ad22ff227d2634a65cb388215ce6cc96bbcc5cfde7641ae87e8dacc0", size = 7445, upload-time = "2025-11-19T20:55:50.744Z" },
]

[[package]]
name = "charset-normalizer"
version = "3.4.9"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/bd/2a/23f34ec9d04624958e137efdc394888716353190e75f25dd22c7a2c7a8aa/charset_normalizer-3.4.9.tar.gz", hash = "sha256:673611bbd43f0810bec0b0f028ddeaaa501190339cac411f347ac76917c3ae7b", size = 152439, upload-time = "2026-07-07T14:34:58.454Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/70/4a/ecbd131485c07fcdfad54e28946d513e3da22ef3b4bd854dcafae54ec739/charset_normalizer-3.4.9-cp312-cp312-macosx_10_13_universal2.whl", hash = "sha256:45b0cc4e3556cd875e09102988d1ab8356c998b596c9fced84547c8138b487a0", size = 319300, upload-time = "2026-07-07T14:33:15.666Z" },
    { url = "https://files.pythonhosted.org/packages/ec/96/5d9364e3342d69f3a045e1777bc47c85c383e6e9466d561b33fdb419d1f9/charset_normalizer-3.4.9-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:9b2aff1c7b3884512b9512c3eaadd9bab39fb45042ffaaa1dd08ff2b9f8109d9", size = 215802, upload-time = "2026-07-07T14:33:17.031Z" },
    { url = "https://files.pythonhosted.org/packages/4b/4c/5361f9aa7f2cb58d94f2ab831b3d493f69efb1d239654b4744e3c09527cb/charset_normalizer-3.4.9-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:9104ed0bd76a429d46f9ec0dbc9b08ad1d2dcdf2b00a5a0daa1c145329b35b44", size = 237171, upload-time = "2026-07-07T14:33:18.576Z" },
    { url = "https://files.pythonhosted.org/packages/50/78/ce342ca4ff30b2eb49fe6d9578df85974f90c67d294113e94efdd9664cbd/charset_normalizer-3.4.9-cp312-cp312-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:7b86a2b16095d250c6f58b3d9b2eee6f4147754344f3dab0922f7c9bf7d226c9", size = 233075, upload-time = "2026-07-07T14:33:20.084Z" },
    { url = "https://files.pythonhosted.org/packages/01/c4/4fa4c8b3097a11f3c5f09a35b72ed6855fb1d332469504962ab7bafcc702/charset_normalizer-3.4.9-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:5e226f6218febc71f6c1fc2fafb91c226f75bdc1d8fb12d66823716e891608fd", size = 224256, upload-time = "2026-07-07T14:33:21.747Z" },
    { url = "https://files.pythonhosted.org/packages/87/3a/ad914516df7e358a81aae018caa5e0470ba827fa6d763b1d2e87d920a5f6/charset_normalizer-3.4.9-cp312-cp312-manylinux_2_31_armv7l.whl", hash = "sha256:90c44bc373b7687f6948b693cceaea1348ae0975d7474746559494468e3c1d84", size = 208784, upload-time = "2026-07-07T14:33:23.313Z" },
    { url = "https://files.pythonhosted.org/packages/d7/74/3c12f9755717dfe5c5c87da63f35d765fa0c00382ec26bf23f7fae34f2ba/charset_normalizer-3.4.9-cp312-cp312-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:9cdef90ae47919cae358d8ab15797a800ed41da7aba5d72419fb510729e2ed4b", size = 219928, upload-time = "2026-07-07T14:33:24.814Z" },
    { url = "https://files.pythonhosted.org/packages/33/9a/895095b83e7907abd6d3d99aad3a38ad0d9686cc186cb0c94c24320fe63e/charset_normalizer-3.4.9-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:60f44ade2cf573dad7a277e6f8ca9a51a21dda572b13bd7d8539bb3cd5dbedde", size = 218489, upload-time = "2026-07-07T14:33:26.42Z" },
    { url = "https://files.pythonhosted.org/packages/a1/34/ef5c05f412f42520d7709b7d3784d19640839eb7366ded1755511585429f/charset_normalizer-3.4.9-cp312-cp312-musllinux_1_2_armv7l.whl", hash = "sha256:a1786910334ed46ab1dd73222f2cd1e05c2c3bb39f6dddb4f8b36fc382058a39", size = 210267, upload-time = "2026-07-07T14:33:27.952Z" },
    { url = "https://files.pythonhosted.org/packages/83/dc/9b29fa4412b318bf3bfea985c35d67eb55e04b59a7c3f2237168b0e0be6f/charset_normalizer-3.4.9-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:03d07803992c6c7bbc976327f34b18b6160327fc81cb82c9d504720ac0be3b62", size = 226030, upload-time = "2026-07-07T14:33:29.397Z" },
    { url = "https://files.pythonhosted.org/packages/0e/42/6dbc00b8cd16011691203e33570fa42ed5746599a2e878112d16eab403a3/charset_normalizer-3.4.9-cp312-cp312-win32.whl", hash = "sha256:78841cccf1af7b40f6f716338d50c0902dbe88d9f800b3c973b7a9a0a693a642", size = 151185, upload-time = "2026-07-07T14:33:30.781Z" },
    { url = "https://files.pythonhosted.org/packages/80/cc/f920afd1a23c58ccd53c1d36085a71893a4737ff5e66e0371efab6809850/charset_normalizer-3.4.9-cp312-cp312-win_amd64.whl", hash = "sha256:4b3dac63058cc36820b0dd072f89898604e2d39686fe05321729d00d8ac185a0", size = 162557, upload-time = "2026-07-07T14:33:32.176Z" },
    { url = "https://files.pythonhosted.org/packages/f0/e6/0386d43a261ff4e4b30c5857af7df877254b46bec7b9d1b74b6bf969a90b/charset_normalizer-3.4.9-cp312-cp312-win_arm64.whl", hash = "sha256:78fa18e436a1a0e58dbd7e02fc4473f3f32cceb12df9dfca542d075961c307d2", size = 152665, upload-time = "2026-07-07T14:33:33.711Z" },
    { url = "https://files.pythonhosted.org/packages/b2/06/97ec2aeae780b31d742b6352218b43841a6871e2564578ca522dce4a45c3/charset_normalizer-3.4.9-cp313-cp313-macosx_10_13_universal2.whl", hash = "sha256:440eede837960000d74978f0eba527be106b5b9aee0daf779d395276ed0b0614", size = 317688, upload-time = "2026-07-07T14:33:35.408Z" },
    { url = "https://files.pythonhosted.org/packages/d0/39/8ff066c672434225f8d25f8b739f992af250944392173dcc88362681c9bf/charset_normalizer-3.4.9-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:21e764fd1e70b6a3e205a0e46f3051701f98a8cb3fad66eeb80e48bb502f8698", size = 214982, upload-time = "2026-07-07T14:33:36.996Z" },
    { url = "https://files.pythonhosted.org/packages/92/8f/3a47a3667c83c2df9483d91644c6c107de3bf8874aa1793da9d3012eb986/charset_normalizer-3.4.9-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:e4fd89cc178bced6ad29cb3e6dd4aa63fa5017c3524dbd0b25998fb64a87cc8b", size = 236460, upload-time = "2026-07-07T14:33:38.536Z" },
    { url = "https://files.pythonhosted.org/packages/f1/60/b22cdbee7e4013dab8b0d7647fc6181120fbbbc8f7025c226d15bd5a47fc/charset_normalizer-3.4.9-cp313-cp313-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:bd47ba7fc3ca94896759ea0109775132d3e7ab921fbf54038e1bab2e46c313c9", size = 232003, upload-time = "2026-07-07T14:33:40.059Z" },
    { url = "https://files.pythonhosted.org/packages/ea/f8/72eb13dcabe7257035cea8aefd922caad2f110d252bf9f67c4c2ca763aee/charset_normalizer-3.4.9-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:84fd18bcc17526fc2b3c1af7d2b9217d32c9c04448c16ec693b9b4f1985c3d33", size = 223149, upload-time = "2026-07-07T14:33:41.631Z" },
    { url = "https://files.pythonhosted.org/packages/b0/3e/faee8f9de92b14ee1198e9163252bb15efee7301b31256a3b6d9ebfdd0dd/charset_normalizer-3.4.9-cp313-cp313-manylinux_2_31_armv7l.whl", hash = "sha256:5b10cd92fc5c498b35a8635df6d5a100207f88b63a4dc1de7ef9a548e1e2cd63", size = 207901, upload-time = "2026-07-07T14:33:43.209Z" },
    { url = "https://files.pythonhosted.org/packages/3a/25/45f30093ae27dd7b92a793b61882a38685f993700113ca36e0c9c14965e1/charset_normalizer-3.4.9-cp313-cp313-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:a4fbdde9dd4a9ce5fd52c2b3a347bb50cc89483ef783f1cb00d408c13f7a96c0", size = 219176, upload-time = "2026-07-07T14:33:44.725Z" },
    { url = "https://files.pythonhosted.org/packages/48/18/c8f397329c35e32f6a837e488986f4ae03bd2abebc453b48714991630c2f/charset_normalizer-3.4.9-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:416c229f77e5ea25b3dfd4b582f8d73d7e43c22320302b9ab128a2d3a0b38efe", size = 217356, upload-time = "2026-07-07T14:33:46.192Z" },
    { url = "https://files.pythonhosted.org/packages/86/7e/5ce0bba863470fd1902d5e5843968951bddf38abe4742fc97116ef4598b3/charset_normalizer-3.4.9-cp313-cp313-musllinux_1_2_armv7l.whl", hash = "sha256:75286256590a6320cf106a0d28970d3560aad9ee09aa7b34fb40524792436d35", size = 209614, upload-time = "2026-07-07T14:33:47.705Z" },
    { url = "https://files.pythonhosted.org/packages/6c/ef/2473d3c4d869155be4af1191111d59c4d5c4e0173026f7e85b176e23bf65/charset_normalizer-3.4.9-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:69b157c5d3292bcd443faca052f3096f637f1e074b98212a933c074ae23dc3b8", size = 224991, upload-time = "2026-07-07T14:33:49.238Z" },
    { url = "https://files.pythonhosted.org/packages/d0/a3/53ddae3db108a088156aa8ddfafd411ebbc1340f48c5573f697b27f69a39/charset_normalizer-3.4.9-cp313-cp313-win32.whl", hash = "sha256:51307f5c71007673a2bf8232ad973483d281e74cb99c8c5a990af1eefa6277d9", size = 150622, upload-time = "2026-07-07T14:33:50.711Z" },
    { url = "https://files.pythonhosted.org/packages/e8/ef/6953a77c7cf2c2ff9998e6f575ab3e380119f100223381565a4f94c1f836/charset_normalizer-3.4.9-cp313-cp313-win_amd64.whl", hash = "sha256:fe2c7201c642b7c308f1675355ad7ff7b66acfe3541625efe5a3ad38f29d6115", size = 161947, upload-time = "2026-07-07T14:33:52.197Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/6e/fb/d560d1d1555debbfe7849d9cac6145c1b537709d79576bf22557ed803b82/charset_normalizer-3.4.9-cp313
-cp313-win_arm64.whl", hash = "sha256:611057cc5d5c0afc743ba8be6bd828c17e0aaa8643f9d0a9b9bb7dea80eb8012", size = 152594, upload-time = "2026-07-07T14:3
3:53.486Z" },
    { url = "https://files.pythonhosted.org/packages/7e/8d/496817fa0944239ecae662dd57ea765cfeaec6a735f9f025d4b7b72e7143/charset_normalizer-3.4.9-cp314
-cp314-macosx_10_15_universal2.whl", hash = "sha256:0327fcd59a935777d83410750c50600ee9571af2846f71ce40f25b13da1ef380", size = 317253, upload-time = "2
026-07-07T14:33:54.994Z" },
    { url = "https://files.pythonhosted.org/packages/2b/f9/ef4a69ea338ad3c0deceea0f5f7d2380ae8b52132b06d652cb0d2cd86706/charset_normalizer-3.4.9-cp314
-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:8a79d9f4d8001473a30c163556b3c3bfebec837495a412dde78b51
672f6134f9", size = 215898, upload-time = "2026-07-07T14:33:56.334Z" },
    { url = "https://files.pythonhosted.org/packages/8c/e7/5ddfd76fc061eb52de219658a4aa431cbacadf0a0219c8854f00da50d289/charset_normalizer-3.4.9-cp314
-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:33bdcc2a32c0a0e861f60841a512c8acc658c87c2ac59d89e3a46d
acf7d866e4", size = 236718, upload-time = "2026-07-07T14:33:57.9Z" },
    { url = "https://files.pythonhosted.org/packages/49/ba/768fa3f36048d81c477a0ce61f813bc1454d80917ccfe550abd9f44f5e24/charset_normalizer-3.4.9-cp314
-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:f840ed6d8ecba8255df8c42b87fadeda98ddfc6eeec05e2dc66e26d46dd6
f58a", size = 232519, upload-time = "2026-07-07T14:33:59.811Z" },
    { url = "https://files.pythonhosted.org/packages/f4/c4/b3e049d2aa3766180c78507110543d9d50894cc97f57de543f1be521dcdc/charset_normalizer-3.4.9-cp314
-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:c25fe15c70c59eb7c5ce8c06a1f3fa1da0ecc5ea1e7a5922c40fd2fa9
b0d5046", size = 223143, upload-time = "2026-07-07T14:34:01.517Z" },
    { url = "https://files.pythonhosted.org/packages/19/79/55c32d06d76ae4feafe053f061f3e3ab70bcf19f4007797ce8c3efda7830/charset_normalizer-3.4.9-cp314
-cp314-manylinux_2_31_armv7l.whl", hash = "sha256:f7fb7d750cfa0a070d2c24e831fd3481019a60dd317ea2b39acbcebc08b6ed81", size = 206742, upload-time = "202
6-07-07T14:34:03.04Z" },
    { url = "https://files.pythonhosted.org/packages/10/e0/47c079dd82d217c807479cd59ffd30af56307ea31c108b75758970459ad3/charset_normalizer-3.4.9-cp314
-cp314-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:4d1c96a7a18b9690a4d46df09e3e3382406ae3213727cd1019ebade1c4a81917", size = 21
9191, upload-time = "2026-07-07T14:34:04.657Z" },
    { url = "https://files.pythonhosted.org/packages/42/ab/b9bc2e77d6b44a7e46ef62ec5cac1c9a6ba7b9135a5d560f002696ec9995/charset_normalizer-3.4.9-cp314
-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:a4cfde78a9f2880208d16a93b795726a3017d5977e08d1e162a7a31322479c41", size = 218328, upload-time = "202
6-07-07T14:34:06.115Z" },
    { url = "https://files.pythonhosted.org/packages/f1/78/c9c71d599f5aa2d42bcdd35cbbd46d7f535351a57e40ff7d8e5a7e219401/charset_normalizer-3.4.9-cp314
-cp314-musllinux_1_2_armv7l.whl", hash = "sha256:d4d6fcde76f94f5cb9e43e9e9a61f16dacefd228cbbf6f1a09bd9b219a92f1a1", size = 207406, upload-time = "2026
-07-07T14:34:07.554Z" },
    { url = "https://files.pythonhosted.org/packages/f6/39/c914445c321a845097ce4f6ac7de9a18228a77b766272125a1ce00d851eb/charset_normalizer-3.4.9-cp314
-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:898f0e9068ca27d37f8e83a5b962821df851532e6c4a7d615c1c033f9da6eedf", size = 225157, upload-time = "2026
-07-07T14:34:09.061Z" },
    { url = "https://files.pythonhosted.org/packages/9b/f2/c0d4b8508565a36bc5c624e88ed297f5b0b1095011034d7f5b83a69908b5/charset_normalizer-3.4.9-cp314
-cp314-win32.whl", hash = "sha256:c1c948747b03be832dceed96ca815cef7360de9aa19d37c730f8e3f6101aca48", size = 151095, upload-time = "2026-07-07T14:34:10
.901Z" },
    { url = "https://files.pythonhosted.org/packages/49/fd/a1d26144398c67486422a72bf5812cda22cb4ccfcd95a290fb41ceb4b8e2/charset_normalizer-3.4.9-cp314
-cp314-win_amd64.whl", hash = "sha256:16b65ea0f2465b6fb52aa22de5eca612aa964ddfec00a912e26f4656cbef890b", size = 162796, upload-time = "2026-07-07T14:3
4:12.47Z" },
    { url = "https://files.pythonhosted.org/packages/20/95/d75e82f8ce9fd323ebf059c16c9aadefb22a1ecde13b7840b35835e4886c/charset_normalizer-3.4.9-cp314
-cp314-win_arm64.whl", hash = "sha256:40a126142a56b2dfc0aacbad1de8310cbf60da7656db0e6b16eebd48e3e93519", size = 153334, upload-time = "2026-07-07T14:3
4:14.044Z" },
    { url = "https://files.pythonhosted.org/packages/00/5e/17398df3a139985ba9d11ed072531986f408c8fca952835ef1ab1820c02b/charset_normalizer-3.4.9-cp314
-cp314t-macosx_10_15_universal2.whl", hash = "sha256:609b3ba8fcc0fb5ab7af00719d0fb6ad0cb518e48e7712d12fd68f1327951198", size = 338848, upload-time = "
2026-07-07T14:34:15.688Z" },
    { url = "https://files.pythonhosted.org/packages/cd/91/7253a32e86b7e1d1239b1b36ba6dd0f021a21107ab33054b53119cc083b9/charset_normalizer-3.4.9-cp314
-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:51447e9aa2684679af07ca5021c3db526e0284347ebf4ffcec115
4c3350cfe32", size = 223022, upload-time = "2026-07-07T14:34:17.248Z" },
    { url = "https://files.pythonhosted.org/packages/cb/32/2e64bd2be10e89c61e57ebe6a93fd98ae88eb7ebe414b5121f22c96c69eb/charset_normalizer-3.4.9-cp314
-cp314t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:cc1b0fff8ead343dae06305f954eb8468ba0ec1a97881f42489d1
98e4ce3c632", size = 241590, upload-time = "2026-07-07T14:34:18.813Z" },
    { url = "https://files.pythonhosted.org/packages/3d/ef/d96ec496cfea0c21db43b0ad03891308b02388d054cc902cf0e5a1ad6a88/charset_normalizer-3.4.9-cp314
-cp314t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:fa36ec09ef71d158186bc79e359ff5fdd6e7996fe8ab638f00d6b93139b
a4fcf", size = 239584, upload-time = "2026-07-07T14:34:20.52Z" },
    { url = "https://files.pythonhosted.org/packages/d4/ce/9af95f7876194bd7a14e3dfe4a4de2e0bff02666a3910d72beafd06cc297/charset_normalizer-3.4.9-cp314
-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:df115d4d83168fdf2cae48ef1ff6d1cb4c466364e30861b37121de0f
3bf1b990", size = 230224, upload-time = "2026-07-07T14:34:22.189Z" },
    { url = "https://files.pythonhosted.org/packages/52/94/af74dde74a3996bd959c350709bfe50e297823d70a8c1cbd54b838880863/charset_normalizer-3.4.9-cp314
-cp314t-manylinux_2_31_armv7l.whl", hash = "sha256:f86c6358749bd4fda175388691e3ba8c46e24c5347d0afd20f9b7edfc9faf07d", size = 212667, upload-time = "20
26-07-07T14:34:23.857Z" },
    { url = "https://files.pythonhosted.org/packages/ee/f0/f1c4fe746c395922961b5916ed1d7d6e7d4c84851d19ed43cc89980ec953/charset_normalizer-3.4.9-cp314
-cp314t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:32286a2c8d167e897177b673176c1e3e00d4057caf5d2b64eef9a3666b03018e", size = 2
27179, upload-time = "2026-07-07T14:34:25.586Z" },
    { url = "https://files.pythonhosted.org/packages/e4/56/6c745619ac397e8871e2bcd3cea1eec86b877488f33888b3aef5c3ed506e/charset_normalizer-3.4.9-cp314
-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:83aed2c10721ddd90f68140685391b50811a880af20654c59af6b6c66c40513c", size = 225372, upload-time = "20
26-07-07T14:34:27.212Z" },
    { url = "https://files.pythonhosted.org/packages/78/ad/98aae8630ac71f16711968e38a5acfecce41b778bf2f0312851020f565a8/charset_normalizer-3.4.9-cp314
-cp314t-musllinux_1_2_armv7l.whl", hash = "sha256:cd6c3d4b783c556fa00bf540854e42f135e2f256abd29669fcd0da0f2dec79c2", size = 215222, upload-time = "202
6-07-07T14:34:28.774Z" },
    { url = "https://files.pythonhosted.org/packages/f7/40/9593d54209765207a7f11073c06494c1721e4ca4a0a426c597679bf7f91e/charset_normalizer-3.4.9-cp314
-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:ee2f2a527e3c1a6e6411eb4209642e138b544a2d72fe5d0d76daf77b24063534", size = 231958, upload-time = "202
6-07-07T14:34:30.345Z" },
    { url = "https://files.pythonhosted.org/packages/b1/27/693ee5e8a18191eb38647360c51cd505013e2bd3b366aa43fd5344c21e3c/charset_normalizer-3.4.9-cp314
-cp314t-win32.whl", hash = "sha256:0d861473f743244d349b50f850d10eb87aeb22bbdcc8e64f79273c94af5a8226", size = 155580, upload-time = "2026-07-07T14:34:3
1.884Z" },
    { url = "https://files.pythonhosted.org/packages/80/3f/bd97d3d9c613013d07cb7733d299385b41df37f0471310f5a73dc359f0b8/charset_normalizer-3.4.9-cp314
-cp314t-win_amd64.whl", hash = "sha256:9b8e0f3107e2200b76f6054de99016eac3ee6762713587b36baaa7e4bd2ae177", size = 167620, upload-time = "2026-07-07T14:
34:33.438Z" },
    { url = "https://files.pythonhosted.org/packages/3d/c6/eee9dca4439b1061f76373f06ea855678cc4a64c1c3c90b50e479edbb8eb/charset_normalizer-3.4.9-cp314
-cp314t-win_arm64.whl", hash = "sha256:19ac87f93086ce37b86e098888555c4b4bc48102279bae3350098c0ed664b501", size = 158037, upload-time = "2026-07-07T14:
34:35.018Z" },
    { url = "https://files.pythonhosted.org/packages/98/2b/f97f1c193fb855c345d678f5077d6926034db0722df74c8f057020e05a25/charset_normalizer-3.4.9-py3-n
one-any.whl", hash = "sha256:68e5f26a1ad57ded6d1cfb85331d1c1a195314756471d97758c48498bb4dcdf5", size = 64538, upload-time = "2026-07-07T14:34:56.993Z"
 },
]

[[package]]
name = "click"
version = "8.4.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "colorama", marker = "sys_platform == 'win32'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/76/d4/81420972a676e8ffea40450d8c8c92943e7218a78fe9b64359836cc9876b/click-8.4.2.tar.gz", hash
= "sha256:9a6cea6e60b17ebe0a44c5cc636d94f09bd66142c1cd7d8b4cd731c4917a15f6", size = 338000, upload-time = "2026-06-24T17:45:15.148Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/fb/e2/79c688af8b210d232694e31e59da9f6ec747bae31c3f5946e4e9b98860d5/click-8.4.2-py3-none-any.whl",
 hash = "sha256:e6f9f66136c816745b9d65817da91d61d957fb16e02e4dcd0552553c5a197b76", size = 119243, upload-time = "2026-06-24T17:45:13.73Z" },
]

[[package]]
name = "colorama"
version = "0.4.6"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/d8/53/6f443c9a4a8358a93a6792e2acffb9d9d5cb0a5cfd8802644b7b1c9a02e4/colorama-0.4.6.tar.gz", ha
```

sh = "sha256:08695f5cb7ed6e0531a20572697297273c47b8cae5a63ffc6d6ed5c201be6e44", size = 27697, upload-time = "2022-10-25T02:36:22.414Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d1/d6/3965ed04c63042e047cb6a3e6ed1a63a35087b6a609aa3a15ed8ac56c221/colorama-0.4.6-py2.py3-none-any.whl", hash = "sha256:4f1d9991f5acc0ca119f9d443620b77f9d6b33703e51011c16baf57afb285fc6", size = 25335, upload-time = "2022-10-25T02:36:20.889Z" },
]

[[package]]
name = "coverage"
version = "7.15.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/f4/45/78dbf9604ee5b3db24efbf26bed1cb58862fb40480cba821963c69348751/coverage-7.15.3.tar.gz", hash = "sha256:ae7ea5a4614acf399ef0483c4cb34f8f8f01df848d8fcbe7d3ce0865733f1c4d", size = 935592, upload-time = "2026-08-02T18:50:17.006Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d1/6c/bac99d9d4c6abe856e93bf3f5212982ac0bfac126dd4a042753bd53bc5af/coverage-7.15.3-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:79a3e32e83227d83d9684459ed579769b56c369ac2d7313099b2d9e031d2e10f", size = 222499, upload-time = "2026-08-02T18:48:15.018Z" },
    { url = "https://files.pythonhosted.org/packages/aa/bc/cb9a39b083bc1aa70586482dab25c9be20bab0ec6c155340e50d9066bb1e/coverage-7.15.3-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:767feb87c5886d781d0a69fafd450a20826ddab7b79bce1665deb64d21441b60", size = 222866, upload-time = "2026-08-02T18:48:16.884Z" },
    { url = "https://files.pythonhosted.org/packages/58/fb/beaa453d62000a0a5b39838bee2a137afe609a50a71f55e83c73461e513b/coverage-7.15.3-cp312-cp312-manylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:50951e37033c40548d777b8a8454a2cd622dba1136780065678dccaec307c47f", size = 254367, upload-time = "2026-08-02T18:48:18.507Z" },
    { url = "https://files.pythonhosted.org/packages/66/64/43e72500ed6815cef189f9193f29d7af4b078830337c95ea976cd0c0d427/coverage-7.15.3-cp312-cp312-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:63a4ff67364afb2cac826b8bbd78a5c50ce656a7b7137436b44d7b96a9271088", size = 257103, upload-time = "2026-08-02T18:48:20.172Z" },
    { url = "https://files.pythonhosted.org/packages/66/3a/2893e2937adfe02f45fd38e4a8a0a0d8b7a02ff9e012ac3d009bee3c4f16/coverage-7.15.3-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:6e95e42856509675fe26560310313a6117640e96f9a1e19bb3d220116a27c94c", size = 258220, upload-time = "2026-08-02T18:48:21.963Z" },
    { url = "https://files.pythonhosted.org/packages/30/b4/d5e6e2eb1a62961083734291304b1f85df72e2abe95c76eb88a7f472afd0/coverage-7.15.3-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:abad631cba27094b4631993f4c72e89ac0ca1b3a0236c7abaf8ca79aea619851", size = 260481, upload-time = "2026-08-02T18:48:23.682Z" },
    { url = "https://files.pythonhosted.org/packages/dc/c9/9b72c5c6a9798a9a12cf65f66e077cc1fdd396e61915c862688f9afe1cae/coverage-7.15.3-cp312-cp312-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:2b0807f1f051dd82a234ad6acdb6f1425baede60be1e84e862496c8cc9262ab9", size = 254749, upload-time = "2026-08-02T18:48:25.32Z" },
    { url = "https://files.pythonhosted.org/packages/92/20/e1c2f759e2dbce559ba85c40c0e4acfecc6cff4b740c294c88e41ccc6111/coverage-7.15.3-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:d8d6df7aeb5bc464040bbc9ae173d875785d3677ebc4307817997d622d74225e", size = 256138, upload-time = "2026-08-02T18:48:27.064Z" },
    { url = "https://files.pythonhosted.org/packages/a5/ab/48cc7e760f769e86ae290a125ea6e7209dfbdbbbb7ff4f5d9d1ee7a45d57/coverage-7.15.3-cp312-cp312-musllinux_1_2_i686.whl", hash = "sha256:974471c506c9f5758808b47c1ebf7949ecd0848f5c1020e78675fefe5ff46866", size = 254283, upload-time = "2026-08-02T18:48:29.082Z" },
    { url = "https://files.pythonhosted.org/packages/15/26/39529a68154f99b3a1829debd8b25eac384effeec890a293b5bbdcb49186/coverage-7.15.3-cp312-cp312-musllinux_1_2_ppc64le.whl", hash = "sha256:5cba0c9c13e35c86df7998f1afaf6b1da224a3a39e4da59bdabf60c148046dcb", size = 258352, upload-time = "2026-08-02T18:48:30.892Z" },
    { url = "https://files.pythonhosted.org/packages/91/2f/55b82aa3d8d7dd8023a56e7c5c2a70e39a3c44b3353c6cf3faec9ad51566/coverage-7.15.3-cp312-cp312-musllinux_1_2_riscv64.whl", hash = "sha256:4d608dc36a364dce33acbf4fc3a50f9d2054c945f233bb0a2cdb4b90bfa17646", size = 253852, upload-time = "2026-08-02T18:48:32.934Z" },
    { url = "https://files.pythonhosted.org/packages/6a/6d/839f4045124cd3518ecf2c58967e58a911202834e7c5a03cfdf2ab0b29f6/coverage-7.15.3-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:2395869280554a1941da904423c12660c39f721315e1c02d076a7fe0971382f0", size = 255725, upload-time = "2026-08-02T18:48:34.848Z" },
    { url = "https://files.pythonhosted.org/packages/75/21/d25e3e2a9e327798078c877f469dfb6def860bf6e25036529046227d3e15/coverage-7.15.3-cp312-cp312-win32.whl", hash = "sha256:24f3b21840c3eb76cef3cc70b2bf6649010c64471a84a446538a39306e1ba04d", size = 224566, upload-time = "2026-08-02T18:48:36.661Z" },
    { url = "https://files.pythonhosted.org/packages/b1/0f/df90cc1e8d095ce263968a93e04829821b2afb31ac2752c06a2e0a8e3c13/coverage-7.15.3-cp312-cp312-win_amd64.whl", hash = "sha256:fa7b17902c3c1dd8a7adb52679b7f6340bba08443d710c8838e04db8cf62be2a", size = 225098, upload-time = "2026-08-02T18:48:38.941Z" },
    { url = "https://files.pythonhosted.org/packages/65/c7/ec49e43c58967a07163e2d1c6bbd58112b825b2772ab66784afd6a5400ba/coverage-7.15.3-cp312-cp312-win_arm64.whl", hash = "sha256:fcbe83fb7258eacd293bf5322d88807acb35ed12a5cfa99dd8215c083e3b0235", size = 224485, upload-time = "2026-08-02T18:48:40.682Z" },
    { url = "https://files.pythonhosted.org/packages/68/6e/62ae61e1fc434956bec38ed1d5b1c494f58cf579dbd998e77abffe7b3e6b/coverage-7.15.3-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:1182eed05674c63d40951fae27c43e822749f04d25f75df64c2e4fa3168678de", size = 222522, upload-time = "2026-08-02T18:48:42.476Z" },
    { url = "https://files.pythonhosted.org/packages/13/ff/c74c673d81e0e77b6608c3d21331e3db42e30daeb3c8a0a8860d4c9e2e14/coverage-7.15.3-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:c0c4b0d7c4cd56e470d0c9d8441f42e8a96cdfd95050fec027f1d4dd9f11006c", size = 222894, upload-time = "2026-08-02T18:48:44.274Z" },
    { url = "https://files.pythonhosted.org/packages/a1/91/ccb30f5ffafd7d69d0b18e5162f9b711a5654e807b7b0c13497f0826b33f/coverage-7.15.3-cp313-cp313-manylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:5c9fce9f4998b0d50a753da765b9215a14decc7863822c89d72da7a89ca625b3", size = 253890, upload-time = "2026-08-02T18:48:46.097Z" },
    { url = "https://files.pythonhosted.org/packages/29/c6/e92a66cda49a2751b09826d51258f199b92aa0cb005bc5f34e9729a52a9c/coverage-7.15.3-cp313-cp313-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:7a47e2a0a0ace9241e70ee00e44520f88b843094603dd54303f1bafecd929c30", size = 256484, upload-time = "2026-08-02T18:48:47.846Z" },
    { url = "https://files.pythonhosted.org/packages/96/7a/730929164b457cf25cf76c23898b90f9039a104a647890801b6586797b14/coverage-7.15.3-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:95bad94f83807ae60ed76f3ac012f69b2605ac9ea81bee959a5a483f7fa09c10", size = 257723, upload-time = "2026-08-02T18:48:49.664Z" },
    { url = "https://files.pythonhosted.org/packages/9e/be/04cb5672cb19f5c389eda81ba22d89807699a949653d3625b0e0fda169da/coverage-7.15.3-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:228e172a76c428bb17d1ab78a2ff188990b0597e5dbd291f52a4edf7412de049", size = 259854, upload-time = "2026-08-02T18:48:51.413Z" },
    { url = "https://files.pythonhosted.org/packages/96/25/5e7fd6af39f6507071455944b8906dd1fe5b7b6bffb6a163ceb20afa0d13/coverage-7.15.3-cp313-cp313-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:cea9fb33887c99349996266f1fd60abe5af3577a90633392001d27ef46b4b66e", size = 254085, upload-time = "2026-08-02T18:48:53.158Z" },
    { url = "https://files.pythonhosted.org/packages/23/c8/55e58a853f1e61163a6e755897bd14a059d78411e86560f39d9951c019b5/coverage-7.15.3-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:81760de3155d7f52c21860c4046628dc6bed182f72e3c028e2b4fd46f65aa040", size = 255850, upload-time = "2026-08-02T18:48:55.031Z" },
    { url = "https://files.pythonhosted.org/packages/be/74/8bcec66dbcf3d22bea2a0b2b77ee2fa6f766a647d0023d4eabbc4f2b2756/coverage-7.15.3-cp313-cp313-musllinux_1_2_i686.whl", hash = "sha256:b47ea0a1d3a3d089826c6cbfad8429d7d8872e28e86baa95ddef330f6875da21", size = 253818, upload-time = "2026-08-02T18:48:57.163Z" },
    { url = "https://files.pythonhosted.org/packages/ce/06/450b673fdfece0997b4e16a31d6bde6b18889c578f1013ddd34c962ac6f9/coverage-7.15.3-cp313-cp313-musllinux_1_2_ppc64le.whl", hash = "sha256:5459ba486b2a5d58a6c05254779ecdf525e7f20174d0210ceda75ba40fdb8f2c", size = 257973, upload-time = "2026-08-02T18:48:59.098Z" },
    { url = "https://files.pythonhosted.org/packages/56/fd/3ec7409aec0ddc943132452b65672f065f043b844f1830e1fe173c98b3ab/coverage-7.15.3-cp313-cp313-musllinux_1_2_riscv64.whl", hash = "sha256:c59209f80a08dbfcdd5109a80dc623cd3b9d22895c85757d34f57a6e6e95570f", size = 253638, upload-time = "2026-08-02T18:49:01.199Z" },
    { url = "https://files.pythonhosted.org/packages/75/20/30a8dabb194123631c93f860fdd86401ad405d56cfb1841873afbfe4e92b/coverage-7.15.3-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:f863856c1779d4a5bb6a94698a2f9073e09c6706501f76f3e7780e72df97d21c", size = 255407, upload-time = "2026-08-02T18:49:03.143Z" },
    { url = "https://files.pythonhosted.org/packages/13/4d/e14365b1953b43653341412f9088b0d752614c626a73a705ff9af400f3a3/coverage-7.15.3-cp313-cp313-win32.whl", hash = "sha256:00cbdc5e322927dc30c5e42b863819b1bb867cc66f26ab5372c585850876ab93", size = 224575, upload-time = "2026-08-02T18:49:05.011Z" },
    { url = "https://files.pythonhosted.org/packages/1c/64/88f762ea80de2070207246faef514513be874486b2773528f2cc2b4b515c/coverage-7.15.3-cp313-cp313-win_amd64.whl", hash = "sha256:835528518a1d823cf336740324b2f335f7c01e609e74abcb5d5163b3e66661e3", size = 225116, upload-time = "2026-08-02T18:49:06.894Z" },
    { url = "https://files.pythonhosted.org/packages/ab/66/03c34c53a319f522554cd29d4f2e16c5eab61aa4cdcf55753129fd7d926c/coverage-7.15.3-cp313-cp313-win_arm64.whl", hash = "sha256:0d2e1f2cbbf36b842f3e2aff8d118c60d677adb498bc6c7fa9c6838738f82767", size = 224509, upload-time = "2026-08-02T18:49:09.129Z" },
    { url = "https://files.pythonhosted.org/packages/35/6f/8c2dc014357618b3226c90f731b8282766c3685786f422558991dc49fbf2/coverage-7.15.3-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:1e3bb08ad574bd9fb6a991f645728f70d333c1c1958dd5fcde65e24cb862813d", size = 222571, upload-time = "2026-08-02T18:49:11.242Z" },
    { url = "https://files.pythonhosted.org/packages/07/50/d867c7ceae9d56b7e74ee61ea834f1aa4f9a1e1c7f0ce39393ba573b1c12/coverage-7.15.3-cp314-cp314-ma

```
cosx_11_0_arm64.whl", hash = "sha256:9e5860eaff02a0b7f1b73304bdf846596ee62ab3a78d25c68044ebf684cb1fef", size = 222902, upload-time = "2026-08-02T18:49
:13.448Z" },
    { url = "https://files.pythonhosted.org/packages/62/77/4f6dfc490c5f2bcacb2d296d9aa4d1e128c43b48e94ad313fec7f49f09ad/coverage-7.15.3-cp314-cp314-ma
nylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:60874e5bd67f0b1bdbe42ab42c7bafa66a6fb8de88721af6df3f7a02713960cd", size = 25
3947, upload-time = "2026-08-02T18:49:15.304Z" },
    { url = "https://files.pythonhosted.org/packages/16/8a/6777f192af264165103e2a3d3768dbadb9894a0a2359a16877141d9ae8f5/coverage-7.15.3-cp314-cp314-ma
nylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:f9147be876e9d83765e0b82176674dc248a6b9283e25e01e7462611b97e9b731", siz
e = 256452, upload-time = "2026-08-02T18:49:17.801Z" },
    { url = "https://files.pythonhosted.org/packages/7d/7b/3d7ac46a0234bc684f41ee42be95e29b2b6525695adb04083609d5ac2149/coverage-7.15.3-cp314-cp314-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:61a01f8c3804760fcc5a3d31c4f3cab792d660d44e17bf7adeaf0ea51e07821
e", size = 257798, upload-time = "2026-08-02T18:49:19.878Z" },
    { url = "https://files.pythonhosted.org/packages/ff/1e/c6ee59c29afcb5fdb35f936381340d1a06429a07c48f20e809646647acbe/coverage-7.15.3-cp314-cp314-ma
nylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:95bf3e7f26f792e25eb185f85a5a659d48479265176dcfe22b6f334fd0081b5
c", size = 260112, upload-time = "2026-08-02T18:49:21.858Z" },
    { url = "https://files.pythonhosted.org/packages/c1/e1/e8ea39a46e89e3a143312ee5f80336e992e3ae8fe44bf9c76b83fefeed42/coverage-7.15.3-cp314-cp314-ma
nylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:44c41eff9e413fed8740eca75d5438ebeb9d3e45e7cd37c67329213e7a72c764", size = 253944, upl
oad-time = "2026-08-02T18:49:23.926Z" },
    { url = "https://files.pythonhosted.org/packages/95/67/31ab5f6a37fd887d1386f81f0da9306851ad2264e9baaa9c7f606e0b3e17/coverage-7.15.3-cp314-cp314-mu
sllinux_1_2_aarch64.whl", hash = "sha256:54146bafb61f3ba9895b43af0dd17eba01561d586d44ce84ea221b0cbbee5a9e", size = 255805, upload-time = "2026-08-02T1
8:49:25.973Z" },
    { url = "https://files.pythonhosted.org/packages/fb/6a/ee505a80c8fd89620fb337c0596daecff87f33171fbb4ee3015fc3d7331f/coverage-7.15.3-cp314-cp314-mu
sllinux_1_2_i686.whl", hash = "sha256:af000dd1bb859ff8066fda4c79512ff938c79811654030722 6b373099c7b151f", size = 253769, upload-time = "2026-08-02T18:4
9:27.883Z" },
    { url = "https://files.pythonhosted.org/packages/b0/41/6ab0f81c9e89660230d8f3f581d4732e5ddb75a885b0a5dfc73d315dc94f/coverage-7.15.3-cp314-cp314-mu
sllinux_1_2_ppc64le.whl", hash = "sha256:a1b82490577f3889950b5a04f18712aef0207243e0749d60fe28c3c73ebfd5fd", size = 258045, upload-time = "2026-08-02T1
8:49:30.201Z" },
    { url = "https://files.pythonhosted.org/packages/bc/62/c995e91cae28cf31d6defab3bfb553dda5ac83ac7381b0f2b121264c307a/coverage-7.15.3-cp314-cp314-mu
sllinux_1_2_riscv64.whl", hash = "sha256:c4fc90a60154c3e4b8a2dc206d6dbe852f1c235c249e0dc0cef909d032c9591a", size = 253587, upload-time = "2026-08-02T1
8:49:32.349Z" },
    { url = "https://files.pythonhosted.org/packages/84/df/f2049980f82d6890321f2065f9e66216eabbf4b2001815db958bc543f40a/coverage-7.15.3-cp314-cp314-mu
sllinux_1_2_x86_64.whl", hash = "sha256:f25bb884814a892948b4c20394db3f2364dd452d9492736479e7a493e63b0eb6", size = 255243, upload-time = "2026-08-02T18
:49:34.324Z" },
    { url = "https://files.pythonhosted.org/packages/1d/82/2c841b67a978c0eb9c3707630b68f93f9e7585d78bb906bc8823ec6b07a5/coverage-7.15.3-cp314-cp314-wi
n32.whl", hash = "sha256:722dbf8e7828fbcfe0dc8586167dc0a5ce85ad6ea171dbb21ed3f8d6581d3cb8", size = 224759, upload-time = "2026-08-02T18:49:36.326Z" },
    { url = "https://files.pythonhosted.org/packages/b3/78/5c93ec43784fd3e404ca23cd0584ae24bc1732de4a3fc194b68c3be88db0/coverage-7.15.3-cp314-cp314-wi
n_amd64.whl", hash = "sha256:64d0845f9c3ed47302bed265c15ab4dbb64aa4ec1490839b8e328f4e7fa914d2", size = 225246, upload-time = "2026-08-02T18:49:38.366Z
" },
    { url = "https://files.pythonhosted.org/packages/9d/77/813a054371f3b018cc63c6bdb46a3c35d5e95d4e3ed4f1449d4196106db5/coverage-7.15.3-cp314-cp314-wi
n_arm64.whl", hash = "sha256:69bc14684f8fbbee9f9dbaa4fe79719b0da9725fc37956785c06ec365acf6926", size = 224673, upload-time = "2026-08-02T18:49:40.552Z
" },
    { url = "https://files.pythonhosted.org/packages/8f/63/8c9f36cc71178d26db930baa03a4494abcc516d8d41bf820d0d85ef1d80b/coverage-7.15.3-cp314-cp314t-m
acosx_10_15_x86_64.whl", hash = "sha256:f92df943c24b96cb215ca26b4f6a2283e63c5db80f1635aceea7fff11311917b", size = 223298, upload-time = "2026-08-02T18
:49:42.634Z" },
    { url = "https://files.pythonhosted.org/packages/54/66/211f24d058ce9f56ebf1420d55b7574fdae924f6da3836f83c8bd4793e38/coverage-7.15.3-cp314-cp314t-m
acosx_11_0_arm64.whl", hash = "sha256:66591c46bdd2971d3ae2bc503a5f0459c2edcaf6b7e045b292000cc95bc6cb95", size = 223568, upload-time = "2026-08-02T18:4
9:44.706Z" },
    { url = "https://files.pythonhosted.org/packages/dd/bb/9c2ad5574a0d6420a96c6cade4f8a683931b9e79fe609f8924d7b6964616/coverage-7.15.3-cp314-cp314t-m
anylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:caa64458b81b18bfc67cdf1f6dc02b23e3edc672f2f8e11771fad75865415a43", size = 2
64932, upload-time = "2026-08-02T18:49:47.153Z" },
    { url = "https://files.pythonhosted.org/packages/ba/91/938c39e77bdd5a0a440412f975609ce3702dabbda6ac715719d93ca45a7b/coverage-7.15.3-cp314-cp314t-m
anylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:447f5421ccf5475956cf516d4ca1d575f487947b6f4e11f9d80c6aefe24b3dc8", si
ze = 267052, upload-time = "2026-08-02T18:49:49.324Z" },
    { url = "https://files.pythonhosted.org/packages/b0/a3/7b431a98af35d9cc6394e54cde9435b33b8591672fbece6a4931267d7a8e/coverage-7.15.3-cp314-cp314t-m
anylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:7a0c77ef8cd483a4987a5d12d1d9d5f7ee598dfdc6c0844417d847e5768dc7
79", size = 269473, upload-time = "2026-08-02T18:49:51.599Z" },
    { url = "https://files.pythonhosted.org/packages/32/58/dbc9951dce46be47a732823a1c571f62bcabdd54a68d8c281489a1a55cfb/coverage-7.15.3-cp314-cp314t-m
anylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:0b273f4ff657446a06c2d85bf80e134fa869a92852ba5f87854a70e1fb44da
77", size = 270591, upload-time = "2026-08-02T18:49:53.865Z" },
    { url = "https://files.pythonhosted.org/packages/71/bd/1d610772c7c0889bfe477a59c46ee66ea53e271f3f06951e9d55b317f7c6/coverage-7.15.3-cp314-cp314t-m
anylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:daea8c4fafa22488600405be2c2be525a9406fba3fc0a83acc726db3e14e2005", size = 264007, up
load-time = "2026-08-02T18:49:55.875Z" },
    { url = "https://files.pythonhosted.org/packages/69/97/852eb3dcdba156b1a9078503f098499916bf889f964b61ad4a08223ac169/coverage-7.15.3-cp314-cp314t-m
usllinux_1_2_aarch64.whl", hash = "sha256:93ff57c530f3fa7aa69f92fb9b8892b8aa82712aa970842f4abf28657f42fb57", size = 266926, upload-time = "2026-08-02T
18:49:57.944Z" },
    { url = "https://files.pythonhosted.org/packages/52/f8/b72cd238757fba2b587fc7dee047efe6e10b0c18343509faaaf502dd4680/coverage-7.15.3-cp314-cp314t-m
usllinux_1_2_i686.whl", hash = "sha256:4df21bef8b800eebda9018f53d49c9ace3aeb0090c850139b27923aafcb83e91", size = 264529, upload-time = "2026-08-02T18:
50:00.035Z" },
    { url = "https://files.pythonhosted.org/packages/b8/0a/6c52ec4b7fb007cb6433d1fcfda4080cb15d75ad37ef9c31025f3427293e/coverage-7.15.3-cp314-cp314t-m
usllinux_1_2_ppc64le.whl", hash = "sha256:db567b02685f26034adcbd85055f80d12cdf02111b8ed00886093d98b2874ce2", size = 268263, upload-time = "2026-08-02T
18:50:02.161Z" },
    { url = "https://files.pythonhosted.org/packages/c0/4e/f1f9aa3efd109a04353563a43fb5155340c1fdcdeaa6296ebed3b6f510ea/coverage-7.15.3-cp314-cp314t-m
usllinux_1_2_riscv64.whl", hash = "sha256:5318dd51b8600b947e058cf5a4fe54d183d9d13c49b97b64ca7be05a34df9bef", size = 263377, upload-time = "2026-08-02T
18:50:04.243Z" },
    { url = "https://files.pythonhosted.org/packages/dd/fb/6b268a0b2728ef1c379ad656b899274477a5f6bed1bf6765b4b387fb0601/coverage-7.15.3-cp314-cp314t-m
usllinux_1_2_x86_64.whl", hash = "sha256:c995bfa383c54704839b6c4c2627a1c00895597ada0e5e8190c81d8bd620555c", size = 265688, upload-time = "2026-08-02T1
8:50:06.428Z" },
    { url = "https://files.pythonhosted.org/packages/29/54/1a3ea96e5d5e7cd41dc432597bfc60692910e635d05e1cc25a8ccc243581/coverage-7.15.3-cp314-cp314t-w
in32.whl", hash = "sha256:6433fafb8da0e1d02eb53411e0ecdadb6b88f0224fdc23317e703c0e88937d42", size = 225066, upload-time = "2026-08-02T18:50:08.533Z" }
,
    { url = "https://files.pythonhosted.org/packages/31/9d/a7b0d9afd18ed5274dd00651a78e7810a931c70d94b79996f150bec1a30f/coverage-7.15.3-cp314-cp314t-w
in_amd64.whl", hash = "sha256:fe578952b1b29fe8c777f43f241d49efac4b56724a3434f5d22ebe3c208df429", size = 225897, upload-time = "2026-08-02T18:50:10.572
Z" },
    { url = "https://files.pythonhosted.org/packages/ca/11/34c5ae40b945e69aa72b87dc268135b7049905f3824af573b7073acbb946/coverage-7.15.3-cp314-cp314t-w
in_arm64.whl", hash = "sha256:d2e1acb7aee29dfa8f3e48c23f36670898baca1209d9bdd3985a50c7f982165e", size = 225212, upload-time = "2026-08-02T18:50:12.63Z
" },
    { url = "https://files.pythonhosted.org/packages/37/e7/7069b3d6c018917f49ba2e1c5fb910e498c7fefa3a1b78cb1b79e61ff45d/coverage-7.15.3-py3-none-any.w
hl", hash = "sha256:da78fa6fc7dafe4212839173133ee85afcf42c5cd5f3e47fa7c1c210453b445e", size = 214297, upload-time = "2026-08-02T18:50:14.709Z" },
]

[[package]]
name = "cryptography"
version = "50.0.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "cffi", marker = "platform_python_implementation != 'PyPy'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/de/41/6cbdcf9142d00fe82836fbb51e503e58088575cf7a0fe1dbff6695bf0840/cryptography-50.0.0.tar.gz
", hash = "sha256:eeac2acb5a20ed25e0ad6d1df9891a520b78b404266b6d11778f25d5d691a6c9", size = 880201, upload-time = "2026-07-31T14:25:10.11Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/c5/5c/59086b4aac5e879d38ddbcf74e4be7ade89cebc3eb199a55da998c3bb46a/cryptography-50.0.0-cp311-abi3
-macosx_11_0_arm64.whl", hash = "sha256:031e2d5dd4bb9caa3ca9c82e5a197fd8ae680232cee62603d1a813f3f07e3d03", size = 4001252, upload-time = "2026-07-31T1
4:23:33.331Z" },
    { url = "https://files.pythonhosted.org/packages/57/ef/8f2df13c7216bcad3e1c74e07f6e193d93e998e114f524a53877c9af27ad/cryptography-50.0.0-cp311-abi3
-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:fd9192b7b70c573d7f214eb1ae35e00d359f6f5e4b27c7e21e30de1fc6204645", size = 4719554,
upload-time = "2026-07-31T14:23:35.611Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/d9/41/029086c34d91052fc3b88bcc8056f709a7c915c7a23b235a54eb800b1c97/cryptography-50.0.0-cp311-abi3-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:06a32a980526a6ab9a4b9bf8f7385800791e2bb960903cb6b530e4817509a3b7", size = 4702130, upload-time = "2026-07-31T14:23:37.635Z" },
    { url = "https://files.pythonhosted.org/packages/7d/ff/b6ce0954962e7f7b969f850a883744197bb3910bdfd7b6da162eab7d9f68/cryptography-50.0.0-cp311-abi3-manylinux_2_28_aarch64.whl", hash = "sha256:a1b30560f2acc95aa8b2e06e716a13dbfc97314747b80d9707e307f77b40d6b3", size = 4725244, upload-time = "2026-07-31T14:23:39.471Z" },
    { url = "https://files.pythonhosted.org/packages/06/1e/63a1027cb7fec360a182208e1b7767d5aa1fe57be3d6aa856e69a321edc0/cryptography-50.0.0-cp311-abi3-manylinux_2_28_ppc64le.whl", hash = "sha256:8d89f3976b10b4ce31118de72329025f70d2c6ead14a8217c5514dd2c6d5a78f", size = 5342265, upload-time = "2026-07-31T14:23:41.286Z" },
    { url = "https://files.pythonhosted.org/packages/6b/72/a1116d683a6d7ece94590013882515de087edf9ef0e6292aae615a44df73/cryptography-50.0.0-cp311-abi3-manylinux_2_28_x86_64.whl", hash = "sha256:b42a28c1844fd9de8f3f7d540e36b66f3a9c83fceac7170ebc7a6a19edd9dcae", size = 4734609, upload-time = "2026-07-31T14:23:43.139Z" },
    { url = "https://files.pythonhosted.org/packages/15/37/36a9c479bbe49acea2636c7fd3360d20f7b7e079c300352011c44850b181/cryptography-50.0.0-cp311-abi3-manylinux_2_31_armv7l.whl", hash = "sha256:900131fafd8aead39ac7dd3a7e833be754c17a95cfd91221636949fe4eb0aa8a", size = 4356517, upload-time = "2026-07-31T14:23:44.939Z" },
    { url = "https://files.pythonhosted.org/packages/32/98/8a151d64367204cbc63ec65d37502f1d9c53cf4bfc6ec3c532614dbec60d/cryptography-50.0.0-cp311-abi3-manylinux_2_34_aarch64.whl", hash = "sha256:07949c449a1abcf60d1ee6e88956d89404c7df3c8258f46589e912988e551987", size = 4724529, upload-time = "2026-07-31T14:23:46.93Z" },
    { url = "https://files.pythonhosted.org/packages/22/f6/ec13b470172126464a86bf54d2294a46d29837fc51ba3e45d4047946fb5e/cryptography-50.0.0-cp311-abi3-manylinux_2_34_ppc64le.whl", hash = "sha256:f89831ef99dd7dd169ab06d63a831adb9e20a87aac6d380266bbda5823349169", size = 5299852, upload-time = "2026-07-31T14:23:48.851Z" },
    { url = "https://files.pythonhosted.org/packages/da/3a/f05e32c99d440c9bb891ea0e36c9091891e36be5a9a87ab2ee6ea20729f6/cryptography-50.0.0-cp311-abi3-manylinux_2_34_x86_64.whl", hash = "sha256:82148ec5bddac30b51a5b3c1945075f896fa022cb93f8e4a01e9f6ee95292c5f", size = 4734462, upload-time = "2026-07-31T14:23:50.861Z" },
    { url = "https://files.pythonhosted.org/packages/ca/dc/bd72b26be8953f80625f63151efd38eee71c76ca6cf591c08ff34615a79e/cryptography-50.0.0-cp311-abi3-musllinux_1_2_aarch64.whl", hash = "sha256:1489e263a8048bb8b6a8bac662eb2d402ea5d2b7b4699b72f385f1e2772db105", size = 4852708, upload-time = "2026-07-31T14:23:52.715Z" },
    { url = "https://files.pythonhosted.org/packages/27/20/c930314a2ab476d15dec966ec87e2e9637bb02b06106b12c0396c57bb603/cryptography-50.0.0-cp311-abi3-musllinux_1_2_x86_64.whl", hash = "sha256:7cec5b856506da6defb290f30c9ee687d5f5e8cb0bd3f6459dde43b0b4fa40ef", size = 5004179, upload-time = "2026-07-31T14:23:54.887Z" },
    { url = "https://files.pythonhosted.org/packages/32/2e/c9db68a0c4bfa28e310707527c0ee3a2bd254104d2e02e68f368e197aa4c/cryptography-50.0.0-cp311-abi3-win_amd64.whl", hash = "sha256:bd1c592e4d5974f0d08d4888e432157adba757c66da0246918e43677fafa2d30", size = 3840395, upload-time = "2026-07-31T14:23:56.677Z" },
    { url = "https://files.pythonhosted.org/packages/c3/fb/951032a3bf22a5697c83183fb6294a4843772947a70e616c57b3ff5f522e/cryptography-50.0.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:49e7d93abdbd2990caced757e5fade25302f719c3c8fb6e6fff2dde98999fc41", size = 3989258, upload-time = "2026-07-31T14:23:58.881Z" },
    { url = "https://files.pythonhosted.org/packages/d4/67/91eb047e69c5e845f2f14b8a2e4a1aab0f283cb885531e9e22c8adb176bc/cryptography-50.0.0-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:19736989797678c6af1e55cd49055cdbcb55d8f6b5583ac5335f933aba9101dc", size = 4700648, upload-time = "2026-07-31T14:24:00.702Z" },
    { url = "https://files.pythonhosted.org/packages/30/82/85f0f7425c856b9f96459411eb12e74ef72df9caf6f8f15bf23a33ff131f/cryptography-50.0.0-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:80b63928fa35083b33966ce1efb70e5b9607181e49dcd1c22c8c005e319f667f", size = 4682442, upload-time = "2026-07-31T14:24:02.538Z" },
    { url = "https://files.pythonhosted.org/packages/1a/28/b555a365adff1cca2fbe7b9e487d68a40de6bc67ff2cb587473eb43de0e7/cryptography-50.0.0-cp314-cp314t-manylinux_2_28_aarch64.whl", hash = "sha256:d58c3db7cd6eed54e6c06744db55456b65ebd7492ddeae9c1e93cfca7aa857d3", size = 4707596, upload-time = "2026-07-31T14:24:04.394Z" },
    { url = "https://files.pythonhosted.org/packages/72/d8/f52538140cc719df62a01cf87d1c7142318d235817109d6f4054d7c352d6/cryptography-50.0.0-cp314-cp314t-manylinux_2_28_ppc64le.whl", hash = "sha256:df2a58a472f332225671c35b0a830208b86d004f82baa8530fa3782c85646533", size = 5314552, upload-time = "2026-07-31T14:24:06.31Z" },
    { url = "https://files.pythonhosted.org/packages/38/14/6120e5bd7c5aa022ad15424ba4d5c5269d0d9448ed4d55e492ea91e3c1c4/cryptography-50.0.0-cp314-cp314t-manylinux_2_28_x86_64.whl", hash = "sha256:11b74db56cdbe3cdee6e3f6982ecb70334fa10dce99ed58bf7894aaaa3b2a037", size = 4717113, upload-time = "2026-07-31T14:24:08.349Z" },
    { url = "https://files.pythonhosted.org/packages/fa/71/190bf38c3ee2e0f8efc9860ae100c9df4169742eef274b91e7aa1cb133b9/cryptography-50.0.0-cp314-cp314t-manylinux_2_31_armv7l.whl", hash = "sha256:f59e38625469987d7ef6d495323c55e7db6c212eaf6112267e0d3b565a2e9c9f", size = 4338580, upload-time = "2026-07-31T14:24:10.227Z" },
    { url = "https://files.pythonhosted.org/packages/3a/63/504ccfbbe61fd8aa983f7f146399cdf034c72c2fc55f5b2dfdcdcdb20c99/cryptography-50.0.0-cp314-cp314t-manylinux_2_34_aarch64.whl", hash = "sha256:ecfed7367f965a0328cfbdd70da860f15441f002f613185668c6e6ebf5a0ac11", size = 4707038, upload-time = "2026-07-31T14:24:12.169Z" },
    { url = "https://files.pythonhosted.org/packages/01/77/2cf79bbfc4d12ca106437a6e170d6aaa01a373e93093118aaaef0e801bd4/cryptography-50.0.0-cp314-cp314t-manylinux_2_34_ppc64le.whl", hash = "sha256:9aa87839c383bdbab6ef865787a1fb877af8dd03464c4400322726feaaadfc6d", size = 5273110, upload-time = "2026-07-31T14:24:14.38Z" },
    { url = "https://files.pythonhosted.org/packages/e5/45/8aae2972c520145377ea3559a605a899bebe227bf070b33cdb445929a9b9/cryptography-50.0.0-cp314-cp314t-manylinux_2_34_x86_64.whl", hash = "sha256:6ba6a53445bd3cfa809ef3ef5f1589aa6ba08784a1d962bf47d0940e871dab1c", size = 4716439, upload-time = "2026-07-31T14:24:16.415Z" },
    { url = "https://files.pythonhosted.org/packages/7b/20/4fe50b619a48c2525cc46e2dbc1ac490708d704be5d467bdaac6dc955682/cryptography-50.0.0-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:3f5735ffe4996d28b809371756219f5354864902a3b9e7c0b9ee87041209fc9c", size = 4837383, upload-time = "2026-07-31T14:24:18.553Z" },
    { url = "https://files.pythonhosted.org/packages/92/91/3a31366e183343d3703f8995c095f5734676bd6938118047e50fcf279eb4/cryptography-50.0.0-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:1b4a266766514614f8aa60416e71f2fc6e575d36e7bdc90f644fadb2f4b75b95", size = 4985772, upload-time = "2026-07-31T14:24:20.385Z" },
    { url = "https://files.pythonhosted.org/packages/74/9a/02ffe35b2853d121689871eb5dce862092562b3a1ed5cc98f1aaed441506/cryptography-50.0.0-cp314-cp314t-win_amd64.whl", hash = "sha256:12b9c6996425c76ea6c457ace4f3073e715b8c545add07cd1a8f3a4f90691269", size = 3816291, upload-time = "2026-07-31T14:24:22.125Z" },
    { url = "https://files.pythonhosted.org/packages/03/37/73d005be173aff344af30e9fd2a576575cb2391a7101d9cd3842e1fa8cce/cryptography-50.0.0-cp39-abi3-macosx_11_0_arm64.whl", hash = "sha256:ccdc4a71a4dabae05de219404f9f4abc38e3b58422177ff93d0da05967dafa07", size = 4036009, upload-time = "2026-07-31T14:24:24.122Z" },
    { url = "https://files.pythonhosted.org/packages/ff/c6/7a6202a534e32103a285b7834a120869557fe198d51d7cfe59754c8bda9c/cryptography-50.0.0-cp39-abi3-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:910e1d2668e7de9648f2bcee30e180db2a6b15c30f887d7c4c93ddf96e3992e3", size = 4745252, upload-time = "2026-07-31T14:24:26.118Z" },
    { url = "https://files.pythonhosted.org/packages/85/4f/0fa8c2f4428198f15d9ff8d63400e27afbf94ce833f6108da1eb3753f945/cryptography-50.0.0-cp39-abi3-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:a91296cb61e8df6f86d0c19cc4068228da256bf59bf86049fbd821084565327f", size = 4728939, upload-time = "2026-07-31T14:24:27.994Z" },
    { url = "https://files.pythonhosted.org/packages/d1/63/54dd723490ba2dc09b299682c10b38db38f159728bcaae8c591b8af2f22d/cryptography-50.0.0-cp39-abi3-manylinux_2_28_aarch64.whl", hash = "sha256:e722f16708d854fe924790e051061f6704a472c3bac347b6fd88033ea8dd0dc5", size = 4748483, upload-time = "2026-07-31T14:24:30.254Z" },
    { url = "https://files.pythonhosted.org/packages/1d/dd/7c77d26285cc7f6991efce64a0f5b4f9383bfa5dd8c5033003eaf7db4cdb/cryptography-50.0.0-cp39-abi3-manylinux_2_28_ppc64le.whl", hash = "sha256:d764dcf130c428ef66786f866dd750f53182bc608813489915e9fc106bb0c82f", size = 5367599, upload-time = "2026-07-31T14:24:32.457Z" },
    { url = "https://files.pythonhosted.org/packages/46/c9/f60aed34c013f317f92817b6c171c2d22a78270fa41109bd4b08af26b194/cryptography-50.0.0-cp39-abi3-manylinux_2_28_x86_64.whl", hash = "sha256:105110f43a471dbd0060b9c9516cb8a6a79233631a04cc2ba16f28323ac6e025", size = 4762647, upload-time = "2026-07-31T14:24:34.599Z" },
    { url = "https://files.pythonhosted.org/packages/be/f3/f9a0173b139372c3a48ed98154b45cc6b9de17c789d5ab552e621c293609/cryptography-50.0.0-cp39-abi3-manylinux_2_31_armv7l.whl", hash = "sha256:828743d939e9629bc267b8e2d08d8bb67cd4319c771a33d4b18b22dd8fb7440a", size = 4385197, upload-time = "2026-07-31T14:24:36.647Z" },
    { url = "https://files.pythonhosted.org/packages/d8/36/83bb81f6e569bc38e1e4a7bc80f29b46bb9601920bc455fc8e888f5d5742/cryptography-50.0.0-cp39-abi3-manylinux_2_34_aarch64.whl", hash = "sha256:2a8183b489dc1f7f80f135780fadc1108f14b31b8a40411c7a5b17425f65f28b", size = 4748095, upload-time = "2026-07-31T14:24:39.493Z" },
    { url = "https://files.pythonhosted.org/packages/6b/16/d3008eff98c764979865834c3d386d4fd041b5f52e7f34fc29ac1a5eb515/cryptography-50.0.0-cp39-abi3-manylinux_2_34_ppc64le.whl", hash = "sha256:6e7d61120573a7f2cd94cc095f9e81f6967c61ccdf194285aa143ecec8e0b708", size = 5325948, upload-time = "2026-07-31T14:24:41.556Z" },
    { url = "https://files.pythonhosted.org/packages/9c/f8/d97f9603efda3888187bfdb893f26c41be4735c10631d05d284ee6b047c4/cryptography-50.0.0-cp39-abi3-manylinux_2_34_x86_64.whl", hash = "sha256:37fdb0d0111f1e2ff07139dfb79f1b49531f8e213c46f1163dd7642979b58c47", size = 4762400, upload-time = "2026-07-31T14:24:43.636Z" },
    { url = "https://files.pythonhosted.org/packages/64/a2/4615c8f7d81a00b1d6e6afe19f694e1543582349fb5f4076f6cb5dc36485/cryptography-50.0.0-cp39-abi3-musllinux_1_2_aarch64.whl", hash = "sha256:c87f62a3d3b9888ed0fdde100ec06aa61ca9cd44bad9057d1dff9a516b5f5bb9", size = 4878208, upload-time = "2026-07-31T14:24:45.522Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/d2/1a/efcfb02f91407149a0dacffffab791f7e19bf6385f63b3666dc8b5e5c9c8/cryptography-50.0.0-cp39-abi3-musllinux_1_2_x86_64.whl", hash = "sha256:65c2c3add92b45fd0709db8594536aea39c2a67af0e27ffcf049c498501140b7", size = 5037050, upload-time = "2026-07-31T14:24:47.697Z" },
    { url = "https://files.pythonhosted.org/packages/57/30/4a22984d4f1bdfb8c054f07a92bc176b97a3134cc1d6c4b3bffb1f3688b4/cryptography-50.0.0-cp39-abi3-win_amd64.whl", hash = "sha256:d24fead1d4d076e1bfb006dcec392074a3cd8d7b4fc8a595aa64073b2b7a96ba", size = 3874135, upload-time = "2026-07-31T14:24:50.085Z" },
]

[[package]]
name = "dataclasses-json"
version = "0.6.7"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "marshmallow" },
    { name = "typing-inspect" },
]
sdist = { url = "https://files.pythonhosted.org/packages/64/a4/f71d9cf3a5ac257c993b5ca3f93df5f7fb395c725e7f1e6479d2514173c3/dataclasses_json-0.6.7.tar.gz", hash = "sha256:b6b3e528266ea45b9535223bc53ca645f5208833c29229e847b3f26a1cc55fc0", size = 32227, upload-time = "2024-06-09T16:20:19.103Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/c3/be/d0d44e092656fe7a06b55e6103cbce807cdbdee17884a5367c68c9860853/dataclasses_json-0.6.7-py3-none-any.whl", hash = "sha256:0dbf33f26c8d5305befd61b39d2b3414e8a407bedc2834dea9b8d642666fb40a", size = 28686, upload-time = "2024-06-09T16:20:16.715Z" },
]

[[package]]
name = "defusedxml"
version = "0.7.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/0f/d5/c66da9b79e5bdb124974bfe172b4daf3c984ebd9c2a06e2b8a4dc7331c72/defusedxml-0.7.1.tar.gz", hash = "sha256:1bb3032db185915b62d7c6209c5a8792be6a32ab2fedacc84e01b52c51aa3e69", size = 75520, upload-time = "2021-03-08T10:59:26.269Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/07/6c/aa3f2f849e01cb6a001cd8554a88d4c77c5c1a31c95bdf1cf9301e6d9ef4/defusedxml-0.7.1-py2.py3-none-any.whl", hash = "sha256:a352e7e428770286cc899e2542b6cdaedb2b4953ff269a210103ec58f6198a61", size = 25604, upload-time = "2021-03-08T10:59:24.45Z" },
]

[[package]]
name = "deprecated"
version = "1.3.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "wrapt" },
]
sdist = { url = "https://files.pythonhosted.org/packages/49/85/12f0a49a7c4ffb70572b6c2ef13c90c88fd190debda93b23f026b25f9634/deprecated-1.3.1.tar.gz", hash = "sha256:b1b50e0ff0c1fddaa5708a2c6b0a6588bb09b892825ab2b214ac9ea9d92a5223", size = 2932523, upload-time = "2025-10-30T08:19:02.757Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/84/d0/205d54408c08b13550c733c4b85429e7ead111c7f0014309637425520a9a/deprecated-1.3.1-py2.py3-none-any.whl", hash = "sha256:597bfef186b6f60181535a29fbe44865ce137a5079f295b479886c82729d5f3f", size = 11298, upload-time = "2025-10-30T08:19:00.758Z" },
]

[[package]]
name = "dirtyjson"
version = "1.0.8"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/db/04/d24f6e645ad82ba0ef092fa17d9ef7a21953781663648a01c9371d9e8e98/dirtyjson-1.0.8.tar.gz", hash = "sha256:90ca4a18f3ff30ce849d100dcf4a003953c79d3a2348ef056f1d9c22231a25fd", size = 30782, upload-time = "2022-11-28T23:32:33.319Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/68/69/1bcf70f81de1b4a9f21b3a62ec0c83bdff991c88d6cc2267d02408457e88/dirtyjson-1.0.8-py3-none-any.whl", hash = "sha256:125e27248435a58acace26d5c2c4c11a1c0de0a9c5124c5a94ba78e517d74f53", size = 25197, upload-time = "2022-11-28T23:32:31.219Z" },
]

[[package]]
name = "distlib"
version = "0.4.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/c9/02/bd72be9134d25ed783ecbbc38a539ffaefbf90c78418c7fb7229600dbac7/distlib-0.4.3.tar.gz", hash = "sha256:f152097224a0ae24be5a0f6bae1b9359af82133bce63f98a95f86cae1aede9ed", size = 615141, upload-time = "2026-06-12T08:04:52.847Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/02/08/9c41fb51ab5b43eb21674aff13df270e8ba6c4b29c8624e328dc7a9482af/distlib-0.4.3-py2.py3-none-any.whl", hash = "sha256:4b0ce306c966eb73bc3a7b6abad017c556dadd92c44701562cd528ac7fde4d5b", size = 470628, upload-time = "2026-06-12T08:04:50.506Z" },
]

[[package]]
name = "distro"
version = "1.9.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/fc/f8/98eea607f65de6527f8a2e8885fc8015d3e6f5775df186e443e0964a11c3/distro-1.9.0.tar.gz", hash = "sha256:2fa77c6fd8940f116ee1d6b94a2f90b13b5ea8d019b98bc8bafdcabcdd9bdbed", size = 60722, upload-time = "2023-12-24T09:54:32.31Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/12/b3/231ffd4ab1fc9d679809f356cebee130ac7daa00d6d6f3206dd4fd137e9e/distro-1.9.0-py3-none-any.whl", hash = "sha256:7bffd925d65168f85027d8da9af6bddab658135b840670a223589bc0c8ef02b2", size = 20277, upload-time = "2023-12-24T09:54:30.421Z" },
]

[[package]]
name = "docstring-parser"
version = "0.18.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/e0/4d/f332313098c1de1b2d2ff91cf2674415cc7cddab2ca1b01ae29774bd5fdf/docstring_parser-0.18.0.tar.gz", hash = "sha256:292510982205c12b1248696f44959db3cdd1740237a968ea1e2e7a900eeb2015", size = 29341, upload-time = "2026-04-14T04:09:19.867Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a7/5f/ed01f9a3cdffbd5a008556fc7b2a08ddb1cc6ace7effa7340604b1d16699/docstring_parser-0.18.0-py3-none-any.whl", hash = "sha256:b3fcbed555c47d8479be0796ef7e19c2670d428d72e96da63f3a40122860374b", size = 22484, upload-time = "2026-04-14T04:09:18.638Z" },
]

[[package]]
name = "ecdsa"
version = "0.19.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "six" },
]
sdist = { url = "https://files.pythonhosted.org/packages/25/ca/8de7744cb3bc966c85430ca2d0fcaeea872507c6a4cf6e007f7fe269ed9d/ecdsa-0.19.2.tar.gz", hash = "sha256:62635b0ac1ca2e027f82122b5b81cb706edc38cd91c63dda28e4f3455a2bf930", size = 202432, upload-time = "2026-03-26T09:58:17.675Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/51/79/119091c98e2bf49e24ed9f3ae69f816d715d2904aefa6a2baa039a2ba0b0/ecdsa-0.19.2-py2.py3-none-any.
```

```
whl", hash = "sha256:840f5dc5e375c68f36c1a7a5b9caad28f95daa65185c9253c0c08dd952bb7399", size = 150818, upload-time = "2026-03-26T09:58:15.808Z" },
]

[[package]]
name = "fastapi"
version = "0.141.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "annotated-doc" },
    { name = "pydantic" },
    { name = "starlette" },
    { name = "typing-extensions" },
    { name = "typing-inspection" },
]
sdist = { url = "https://files.pythonhosted.org/packages/8a/02/91e3416a8fdd715abb903a952a6bec7cdd8d14eed55d415fc8595524c319/fastapi-0.141.1.tar.gz", h
ash = "sha256:e8822fc40db1e1858054d7a949a888695bc9bdce70139178e33bd2871a453ca1", size = 425799, upload-time = "2026-07-29T17:18:05.568Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/cb/03/10388a42375ee7e4ac9b94eb2c5c569c8b5795e377e701c9ac3ad63de890/fastapi-0.141.1-py3-none-any.w
hl", hash = "sha256:bfb91aa2d334c61cb35ba9a116fc123b3d3df31640b801cf57a7a78ec3f603b3", size = 131954, upload-time = "2026-07-29T17:18:04.364Z" },
]

[[package]]
name = "filelock"
version = "3.32.2"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/f6/57/3ba6e6cb097f85b855b00163d169f35365f44277df044dcf96d55b8f62a3/filelock-3.32.2.tar.gz", h
ash = "sha256:c33351e1f49cae33414acbc6d56784e6ecee82514ec90795da1161fc4836b5b8", size = 217172, upload-time = "2026-07-29T22:46:04.895Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/c1/e8/72f8cef9fdfeffe06213fe8508039396ee48daa0e3259457ed766173bfd6/filelock-3.32.2-py3-none-any.w
hl", hash = "sha256:87dd94cf281e586d135fa51132b8e3d9a598b316e90377a288663c9321036c82", size = 98830, upload-time = "2026-07-29T22:46:03.52Z" },
]

[[package]]
name = "filetype"
version = "1.2.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/bb/29/745f7d30d47fe0f251d3ad3dc2978a2314191766199876 3bebb6da007eb1/filetype-1.2.0.tar.gz", ha
sh = "sha256:66b56cd6474bf41d8c54660347d37afcc3f7d1970648de365c102ef77548aadb", size = 998020, upload-time = "2022-11-02T17:34:04.141Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/18/79/1b8fa1bb3568781e84c9200f951c735f3f157429f44be0495da55894d620/filetype-1.2.0-py2.py3-none-an
y.whl", hash = "sha256:7ce71b6880181241cf7ac8697a2f1eb6a8bd9b429f7ad6d27b8db9ba5f1c2d25", size = 19970, upload-time = "2022-11-02T17:34:01.425Z" },
]

[[package]]
name = "frozenlist"
version = "1.8.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/2d/f5/c831fac6cc817d26fd54c7eaccd04ef7e0288806943f7cc5bbf69f3ac1f0/frozenlist-1.8.0.tar.gz",
hash = "sha256:3ede829ed8d842f6cd48fc7081d7a41001a56f1f38603f9d49bf3020d59a31ad", size = 45875, upload-time = "2025-10-06T05:38:17.865Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/69/29/948b9aa87e75820a38650af445d2ef2b6b8a6fab1a23b6bb9e4ef0be2d59/frozenlist-1.8.0-cp312-cp312-m
acosx_10_13_universal2.whl", hash = "sha256:78f7b9e5d6f2fdb88cdde9440dc147259b62b9d3b019924def9f6478be254ac1", size = 87782, upload-time = "2025-10-06
T05:36:06.649Z" },
    { url = "https://files.pythonhosted.org/packages/64/80/4f6e318ee2a7c0750ed724fa33a4bdf1eacdc5a39a7a24e818a773cd91af/frozenlist-1.8.0-cp312-cp312-m
acosx_10_13_x86_64.whl", hash = "sha256:229bf37d2e4acdaf808fd3f06e854a4a7a3661e871b10dc1f8f1896a3b05f18b", size = 50594, upload-time = "2025-10-06T05:
36:07.69Z" },
    { url = "https://files.pythonhosted.org/packages/2b/94/5c8a2b50a496b11dd519f4a24cb5496cf125681dd99e94c604ccdea9419a/frozenlist-1.8.0-cp312-cp312-m
acosx_11_0_arm64.whl", hash = "sha256:f833670942247a14eafbb675458b4e61c82e002a148f49e68257b79296e865c4", size = 50448, upload-time = "2025-10-06T05:36
:08.78Z" },
    { url = "https://files.pythonhosted.org/packages/6a/bd/d91c5e39f490a49df14320f4e8c80161cfcce09f1e2cde1edd16a551abb3/frozenlist-1.8.0-cp312-cp312-m
anylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:494a5952b1c597ba44e0e78113a7266e656b9794eec897b19ead706bd7074383", si
ze = 242411, upload-time = "2025-10-06T05:36:09.801Z" },
    { url = "https://files.pythonhosted.org/packages/8f/83/f61505a05109ef3293dfb1ff594d13d64a2324ac3482be2cedc2be818256/frozenlist-1.8.0-cp312-cp312-m
anylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:96f423a119f4777a4a056b66ce11527366a8bb92f54e541ade21f2374433f6
d4", size = 243014, upload-time = "2025-10-06T05:36:11.394Z" },
    { url = "https://files.pythonhosted.org/packages/d8/cb/cb6c7b0f7d4023ddda30cf56b8b17494eb3a79e3fda666bf735f63118b35/frozenlist-1.8.0-cp312-cp312-m
anylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:3462dd9475af2025c31cc61be6652dfa25cbfb56cbbf52f4ccfe029f38decaf8"
, size = 234909, upload-time = "2025-10-06T05:36:12.598Z" },
    { url = "https://files.pythonhosted.org/packages/31/c5/cd7a1f3b8b34af009fb17d4123c5a778b44ae2804e3ad6b86204255f9ec5/frozenlist-1.8.0-cp312-cp312-m
anylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:c4c800524c9cd9bac5166cd6f55285957fcfc907db323e193f2afcd4d9abd6
9b", size = 250049, upload-time = "2025-10-06T05:36:14.065Z" },
    { url = "https://files.pythonhosted.org/packages/c0/01/2f95d3b416c584a1e7f0e1d6d31998c4a795f7544069ee2e0962a4b60740/frozenlist-1.8.0-cp312-cp312-m
anylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:d6a5df73acd3399d893dafc71663ad22534b5aa4f94e8a2fabfe856c3c1b6a52", s
ize = 256485, upload-time = "2025-10-06T05:36:15.39Z" },
    { url = "https://files.pythonhosted.org/packages/ce/03/024bf7720b3abaebcff6d0793d73c154237b85bdf67b7ed55e5e9596dc9a/frozenlist-1.8.0-cp312-cp312-m
usllinux_1_2_aarch64.whl", hash = "sha256:405e8fe955c2280ce66428b3ca55e12b3c4e9c336fb2103a4937e891c69a4a29", size = 237619, upload-time = "2025-10-06T
05:36:16.558Z" },
    { url = "https://files.pythonhosted.org/packages/69/fa/f8abdfe7d76b731f5d8bd217827cf6764d4f1d9763407e42717b4bed50a0/frozenlist-1.8.0-cp312-cp312-m
usllinux_1_2_armv7l.whl", hash = "sha256:908bd3f6439f2fef9e85031b59fd4f1297af54415fb60e4254a95f75b3cab3f3", size = 250320, upload-time = "2025-10-06T0
5:36:17.821Z" },
    { url = "https://files.pythonhosted.org/packages/f5/3c/b051329f718b463b22613e269ad72138cc256c540f78a6de89452803a47d/frozenlist-1.8.0-cp312-cp312-m
usllinux_1_2_ppc64le.whl", hash = "sha256:294e487f9ec720bd8ffcebc99d575f7eff3568a08a253d1ee1a0378754b74143", size = 246820, upload-time = "2025-10-06T
05:36:19.046Z" },
    { url = "https://files.pythonhosted.org/packages/0f/ae/58282e8f98e444b3f4dd42448ff36fa38bef29e40d40f330b22e7108f565/frozenlist-1.8.0-cp312-cp312-m
usllinux_1_2_s390x.whl", hash = "sha256:74c51543498289c0c43656701be6b077f4b265868fa7f8a8859c197006efb608", size = 250518, upload-time = "2025-10-06T05
:36:20.763Z" },
    { url = "https://files.pythonhosted.org/packages/8f/96/007e5944694d66123183845a106547a15944fbbb7154788cbf7272789536/frozenlist-1.8.0-cp312-cp312-m
usllinux_1_2_x86_64.whl", hash = "sha256:776f352e8329135506a1d6bf16ac3f87bc25b28e765949282dcc627af36123aa", size = 239096, upload-time = "2025-10-06T0
5:36:22.129Z" },
    { url = "https://files.pythonhosted.org/packages/66/bb/852b9d6db2fa40be96f29c0d1205c306288f0684df8fd26ca1951d461a56/frozenlist-1.8.0-cp312-cp312-w
in32.whl", hash = "sha256:433403ae80709741ce34038da08511d4a77062aa924baf411ef73d1146e74faf", size = 39985, upload-time = "2025-10-06T05:36:23.661Z" },
    { url = "https://files.pythonhosted.org/packages/b8/af/38e51a553dd66eb064cdf193841f16f077585d4d28394c2fa6235cb41765/frozenlist-1.8.0-cp312-cp312-w
in_amd64.whl", hash = "sha256:34187385b08f866104f0c0617404c8eb08165ab1272e884abc89c112e9c00746", size = 44591, upload-time = "2025-10-06T05:36:24.958Z
" },
    { url = "https://files.pythonhosted.org/packages/a7/06/1dc65480ab147339fecc70797e9c2f69d9cea9cf38934ce08df070fdb9cb/frozenlist-1.8.0-cp312-cp312-w
in_arm64.whl", hash = "sha256:fe3c58d2f5db5fbd18c2987cba06d51b0529f52bc3a6cdc33d3f4eab725104bd", size = 40102, upload-time = "2025-10-06T05:36:26.333Z
" },
    { url = "https://files.pythonhosted.org/packages/2d/40/0832c31a37d60f60ed79e9dfb5a92e1e2af4f40a16a29abcc7992af9edff/frozenlist-1.8.0-cp313-cp313-m
acosx_10_13_universal2.whl", hash = "sha256:8d92f1a84bb12d9e56f818b3a746f3efba93c1b63c8387a73dde655e1e42282a", size = 85717, upload-time = "2025-10-06
T05:36:27.341Z" },
    { url = "https://files.pythonhosted.org/packages/30/ba/b0b3de23f40bc55a7057bd38434e25c34fa48e17f20ee273bbde5e0650f3/frozenlist-1.8.0-cp313-cp313-m
acosx_10_13_x86_64.whl", hash = "sha256:96153e77a591c8adc2ee805756c61f59fef4cf4073a9275ee86fe8cba41241f7", size = 49651, upload-time = "2025-10-06T05:
36:28.855Z" },
    { url = "https://files.pythonhosted.org/packages/0c/ab/6e5080ee374f875296c4243c381bbdef97a9ac39c6e3ce1d5f7d42cb78d6/frozenlist-1.8.0-cp313-cp313-m
acosx_11_0_arm64.whl", hash = "sha256:f21f00a91358803399890ab167098c131ec2ddd5f8f5fd5fe9c9f2c6fcd91e40", size = 49417, upload-time = "2025-10-06T05:36
```

:29.877Z" },
    { url = "https://files.pythonhosted.org/packages/d5/4e/e4691508f9477ce67da2015d8c00acd751e6287739123113a9fca6f1604e/frozenlist-1.8.0-cp313-cp313-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:fb30f9626572a76dfe4293c7194a09fb1fe93ba94c7d4f720dfae3b646b45027", size = 234391, upload-time = "2025-10-06T05:36:31.301Z" },
    { url = "https://files.pythonhosted.org/packages/40/76/c202df58e3acdf12969a7895fd6f3bc016c642e6726aa63bd3025e0fc71c/frozenlist-1.8.0-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:eaa352d7047a31d87dafcacbabe89df0aa506abb5b1b85a2fb91bc3faa02d822", size = 233048, upload-time = "2025-10-06T05:36:32.531Z" },
    { url = "https://files.pythonhosted.org/packages/f9/c0/8746afb90f17b73ca5979c7a3958116e105ff796e718575175319b5bb4ce/frozenlist-1.8.0-cp313-cp313-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:03ae967b4e297f58f8c774c7eabcce57fe3c2434817d4385c50661845a058121", size = 226549, upload-time = "2025-10-06T05:36:33.706Z" },
    { url = "https://files.pythonhosted.org/packages/7e/eb/4c7eefc718ff72f9b6c4893291abaae5fbc0c82226a32dcd8ef4f7a5dbef/frozenlist-1.8.0-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:f6292f1de555ffcc675941d65ffffffb0a5bcd992905015f85d0592201793e0e5", size = 239833, upload-time = "2025-10-06T05:36:34.947Z" },
    { url = "https://files.pythonhosted.org/packages/c2/4e/e5c02187cf704224f8b21bee886f3d713ca379535f16893233b9d672ea71/frozenlist-1.8.0-cp313-cp313-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:29548f9b5b5e3460ce7378144c3010363d8035cea44bc0bf02d57f5a685e084e", size = 245363, upload-time = "2025-10-06T05:36:36.534Z" },
    { url = "https://files.pythonhosted.org/packages/1f/96/cb85ec608464472e82ad37a17f844889c36100eed57bea094518bf270692/frozenlist-1.8.0-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:ec3cc8c5d4084591b4237c0a272cc4f50a5b03396a47d9caaf76f5d7b38a4f11", size = 229314, upload-time = "2025-10-06T05:36:38.582Z" },
    { url = "https://files.pythonhosted.org/packages/5d/6f/4ae69c550e4cee66b57887daeebe006fe985917c01d0fff9caab9883f6d0/frozenlist-1.8.0-cp313-cp313-musllinux_1_2_armv7l.whl", hash = "sha256:517279f58009d0b1f2e7c1b130b377a349405da3f7621ed6bfae50b10adf20c1", size = 243365, upload-time = "2025-10-06T05:36:40.152Z" },
    { url = "https://files.pythonhosted.org/packages/7a/58/afd56de246cf11780a40a2c28dc7cbabbf06337cc8ddb1c780a2d97e88d8/frozenlist-1.8.0-cp313-cp313-musllinux_1_2_ppc64le.whl", hash = "sha256:db1e72ede2d0d7ccb213f218df6a078a9c09a7de257c2fe8fcef16d5925230b1", size = 237763, upload-time = "2025-10-06T05:36:41.355Z" },
    { url = "https://files.pythonhosted.org/packages/cb/36/cdfaf6ed42e2644740d4a10452d8e97fa1c062e2a8006e4b09f1b5fd7d63/frozenlist-1.8.0-cp313-cp313-musllinux_1_2_s390x.whl", hash = "sha256:b4dec9482a65c54a5044486847b8a66bf10c9cb4926d42927ec4e8fd5db7fed8", size = 240110, upload-time = "2025-10-06T05:36:42.716Z" },
    { url = "https://files.pythonhosted.org/packages/03/a8/9ea226fbefad669f11b52e864c55f0bd57d3c8d7eb07e9f2e9a0b39502e1/frozenlist-1.8.0-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:21900c48ae04d13d416f0e1e0c4d81f7931f73a9dfa0b7a8746fb2fe7dd970ed", size = 233717, upload-time = "2025-10-06T05:36:44.251Z" },
    { url = "https://files.pythonhosted.org/packages/1e/0b/1b5531611e83ba7d13ccc9988967ea1b51186af64c42b7a7af465dcc9568/frozenlist-1.8.0-cp313-cp313-win32.whl", hash = "sha256:8b7b94a067d1c504ee0b16def57ad5738701e4ba10cec90529f13fa03c833496", size = 39628, upload-time = "2025-10-06T05:36:45.423Z" },
    { url = "https://files.pythonhosted.org/packages/d8/cf/174c91dbc9cc49bc7b7aab74d8b734e974d1faa8f191c74af9b7e80848e6/frozenlist-1.8.0-cp313-cp313-win_amd64.whl", hash = "sha256:878be833caa6a3821caf85eb39c5ba92d28e85df26d57afb06b35b2efd937231", size = 43882, upload-time = "2025-10-06T05:36:46.796Z" },
    { url = "https://files.pythonhosted.org/packages/c1/17/502cd212cbfa96eb1388614fe39a3fc9ab87dbbe042b66f97acb57474834/frozenlist-1.8.0-cp313-cp313-win_arm64.whl", hash = "sha256:44389d135b3ff43ba8cc89ff7f51f5a0bb6b63d829c8300f79a2fe4fe61bcc62", size = 39676, upload-time = "2025-10-06T05:36:47.8Z" },
    { url = "https://files.pythonhosted.org/packages/d2/5c/3bbfaa920dfab09e76946a5d2833a7cbdf7b9b4a91c714666ac4855b88b4/frozenlist-1.8.0-cp313-cp313t-macosx_10_13_universal2.whl", hash = "sha256:e25ac20a2ef37e91c1b39938b591457666a0fa835c7783c3a8f33ea42870db94", size = 89235, upload-time = "2025-10-06T05:36:48.78Z" },
    { url = "https://files.pythonhosted.org/packages/d2/d6/f03961ef72166cec1687e84e8925838442b615bd0b8854b54923ce5b7b8a/frozenlist-1.8.0-cp313-cp313t-macosx_10_13_x86_64.whl", hash = "sha256:07cdca25a91a4386d2e76ad992916a85038a9b97561bf7a3fd12d5d9ce31870c", size = 50742, upload-time = "2025-10-06T05:36:49.837Z" },
    { url = "https://files.pythonhosted.org/packages/1e/bb/a6d12b7ba4c3337667d0e421f7181c82dda448ce4e7ad7ecd249a16fa806/frozenlist-1.8.0-cp313-cp313t-macosx_11_0_arm64.whl", hash = "sha256:4e0c11f2cc6717e0a741f84a527c52616140741cd812a50422f83dc31749fb52", size = 51725, upload-time = "2025-10-06T05:36:50.851Z" },
    { url = "https://files.pythonhosted.org/packages/bc/71/d1fed0ffe2c2ccd70b43714c6cab0f4188f09f8a67a7914a6b46ee30f274/frozenlist-1.8.0-cp313-cp313t-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:b3210649ee28062ea6099cfda39e147fa1bc039583c8ee4481cb7811e2448c51", size = 284533, upload-time = "2025-10-06T05:36:51.898Z" },
    { url = "https://files.pythonhosted.org/packages/c9/1f/fb1685a7b009d89f9bf78a42d94461bc06581f6e718c39344754a5d9bada/frozenlist-1.8.0-cp313-cp313t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:581ef5194c48035a7de2aefc72ac6539823bb71508189e5de01d60c9dcd5fa65", size = 292506, upload-time = "2025-10-06T05:36:53.101Z" },
    { url = "https://files.pythonhosted.org/packages/e6/3b/b991fe1612703f7e0d05c0cf734c1b77aaf7c7d321df4572e8d36e7048c8/frozenlist-1.8.0-cp313-cp313t-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:3ef2d026f16a2b1866e1d86fc4e1291e1ed8a387b2c333809419a2f8b3a77b82", size = 274161, upload-time = "2025-10-06T05:36:54.309Z" },
    { url = "https://files.pythonhosted.org/packages/ca/ec/c5c618767bcdf66e88945ec0157d7f6c4a1322f1473392319b7a2501ded7/frozenlist-1.8.0-cp313-cp313t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:5500ef82073f599ac84d888e3a8c1f77ac831183244bfd7f11eaa0289fb30714", size = 294676, upload-time = "2025-10-06T05:36:55.566Z" },
    { url = "https://files.pythonhosted.org/packages/7c/ce/3934758637d8f8a88d11f0585d6495ef54b2044ed6ec84492a91fa3b27aa/frozenlist-1.8.0-cp313-cp313t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:50066c3997d0091c411a66e710f4e11752251e6d2d73d70d8d5d4c76442a199d", size = 300638, upload-time = "2025-10-06T05:36:56.758Z" },
    { url = "https://files.pythonhosted.org/packages/fc/4f/a7e4d0d467298f42de4b41cbc7ddaf19d3cfeabaf9ff97c20c6c7ee409f9/frozenlist-1.8.0-cp313-cp313t-musllinux_1_2_aarch64.whl", hash = "sha256:5c1c8e78426e59b3f8005e9b19f6ff46e5845895adbde20ece9218319eca6506", size = 283067, upload-time = "2025-10-06T05:36:57.965Z" },
    { url = "https://files.pythonhosted.org/packages/dc/48/c7b163063d55a83772b268e6d1affb960771b0e203b632cfe09522d67ea5/frozenlist-1.8.0-cp313-cp313t-musllinux_1_2_armv7l.whl", hash = "sha256:eefdba20de0d938cec6a89bd4d70f346a03108a19b9df4248d3cf0d88f1b0f51", size = 292101, upload-time = "2025-10-06T05:36:59.237Z" },
    { url = "https://files.pythonhosted.org/packages/9f/d0/2366d3c4ecdc2fd391e0afa6e11500bfba0ea772764d631bbf82f0136c9d/frozenlist-1.8.0-cp313-cp313t-musllinux_1_2_ppc64le.whl", hash = "sha256:cf253e0e1c3ceb4aaff6df637ce033ff6535fb8c70a764a8f46aafd3d6ab798e", size = 289901, upload-time = "2025-10-06T05:37:00.811Z" },
    { url = "https://files.pythonhosted.org/packages/b8/94/daff920e82c1b70e3618a2ac39fbc01ae3e2ff6124e80739ce5d71c9b920/frozenlist-1.8.0-cp313-cp313t-musllinux_1_2_s390x.whl", hash = "sha256:032efa2674356903cd0261c4317a561a6850f3ac864a63fc1583147fb05a79b0", size = 289395, upload-time = "2025-10-06T05:37:02.115Z" },
    { url = "https://files.pythonhosted.org/packages/e3/20/bba307ab4235a09fdcd3cc5508dbabd17c4634a1af4b96e0f69bfe551ebd/frozenlist-1.8.0-cp313-cp313t-musllinux_1_2_x86_64.whl", hash = "sha256:6da155091429aeba16851ecb10a9104a108bcd32f6c1642867eadaee401c1c41", size = 283659, upload-time = "2025-10-06T05:37:03.711Z" },
    { url = "https://files.pythonhosted.org/packages/fd/00/04ca1c3a7a124b6de4f8a9a17cc2fcad138b4608e7a3fc5877804b8715d7/frozenlist-1.8.0-cp313-cp313t-win32.whl", hash = "sha256:0f96534f8bfebc1a394209427d0f8a63d343c9779cda6fc25e8e121b5fd8555b", size = 43492, upload-time = "2025-10-06T05:37:04.915Z" },
    { url = "https://files.pythonhosted.org/packages/59/5e/c69f733a86a94ab10f68e496dc6b7e8bc078ebb415281d5698313e3af3a1/frozenlist-1.8.0-cp313-cp313t-win_amd64.whl", hash = "sha256:5d63a068f978fc69421fb0e6eb91a9603187527c86b7cd3f534a5b77a592b888", size = 48034, upload-time = "2025-10-06T05:37:06.343Z" },
    { url = "https://files.pythonhosted.org/packages/16/6c/be9d79775d8abe79b05fa6d23da99ad6e7763a1d080fbae7290b286093fd/frozenlist-1.8.0-cp313-cp313t-win_arm64.whl", hash = "sha256:bf0a7e10b077bf5fb9380ad3ae8ce20ef919a6ad93b4552896419ac7e1d8e042", size = 41749, upload-time = "2025-10-06T05:37:07.431Z" },
    { url = "https://files.pythonhosted.org/packages/f1/c8/85da824b7e7b9b6e7f7705b2ecaf9591ba6f79c1177f324c2735e41d36a2/frozenlist-1.8.0-cp314-cp314-macosx_10_13_universal2.whl", hash = "sha256:cee686f1f4cadeb2136007ddedd0aaf928ab95216e7691c63e50a8ec066336d0", size = 86127, upload-time = "2025-10-06T05:37:08.438Z" },
    { url = "https://files.pythonhosted.org/packages/8e/e8/a1185e236ec66c20afd72399522f142c3724c785789255202d27ae992818/frozenlist-1.8.0-cp314-cp314-macosx_10_13_x86_64.whl", hash = "sha256:119fb2a1bd47307e899c2fac7f28e85b9a543864df47aa7ec9d3c1b4545f096f", size = 49698, upload-time = "2025-10-06T05:37:09.48Z" },
    { url = "https://files.pythonhosted.org/packages/a1/93/72b1736d68f03fda5fdf0f2180fb6caaae3894f1b854d006ac61ecc727ee/frozenlist-1.8.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:4970ece02dbc8c3a92fcc5228e36a3e933a01a999f7094ff7c23fbd2beeaa67c", size = 49749, upload-time = "2025-10-06T05:37:10.569Z" },
    { url = "https://files.pythonhosted.org/packages/a7/b2/fabede9fafd976b991e9f1b9c8c873ed86f202889b864756f240ce6dd855/frozenlist-1.8.0-cp314-cp314-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:cba69cb73723c3f329622e34bdbf5ce1f80c21c290ff04256cff1cd3c2036ed2", size = 231298, upload-time = "2025-10-06T05:37:11.993Z" },
    { url = "https://files.pythonhosted.org/packages/3a/3b/d9b1e0b0eed36e70477ffb8360c49c85c8ca8ef9700a4e6711f39a6e8b45/frozenlist-1.8.0-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:778a11b15673f6f1df23d9586f83c4846c471a8af693a22e066508b77d201ec8", size = 232015, upload-time = "2025-10-06T05:37:13.194Z" },
    { url = "https://files.pythonhosted.org/packages/dc/94/be719d2766c1138148564a3960fc2c06eb688da592bdc25adcf856101be7/frozenlist-1.8.0-cp314-cp314-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:0325024fe97f94c41c08872db482cf8ac4800d80e79222c6b0b7b162d5b13686", size = 225038, upload-time = "2025-10-06T05:37:14.577Z" },

```
    { url = "https://files.pythonhosted.org/packages/e4/09/6712b6c5465f083f52f50cf74167b92d4ea2f50e46a9eea0523d658454ae/frozenlist-1.8.0-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:97260ff46b207a82a7567b581ab4190bd4dfa09f4db8a8b49d1a958f6aa4940e", size = 240130, upload-time = "2025-10-06T05:37:15.781Z" },
    { url = "https://files.pythonhosted.org/packages/f8/d4/cd065cdcf21550b54f3ce6a22e143ac9e4836ca42a0de1022da8498eac89/frozenlist-1.8.0-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:54b2077180eb7f83dd52c40b2750d0a9f175e06a42e3213ce047219de902717a", size = 242845, upload-time = "2025-10-06T05:37:17.037Z" },
    { url = "https://files.pythonhosted.org/packages/62/c3/f57a5c8c70cd1ead3d5d5f776f89d33110b1addae0ab010ad774d9a44fb9/frozenlist-1.8.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:2f05983daecab868a31e1da44462873306d3cbfd76d1f0b5b69c473d21dbb128", size = 229131, upload-time = "2025-10-06T05:37:18.221Z" },
    { url = "https://files.pythonhosted.org/packages/6c/52/232476fe9cb64f0742f3fde2b7d26c1dac18b6d62071c74d4ded55e0ef94/frozenlist-1.8.0-cp314-cp314-musllinux_1_2_armv7l.whl", hash = "sha256:33f48f51a446114bc5d251fb2954ab0164d5be02ad3382abcbfe07e2531d650f", size = 240542, upload-time = "2025-10-06T05:37:19.771Z" },
    { url = "https://files.pythonhosted.org/packages/5f/85/07bf3f5d0fb5414aee5f47d33c6f5c77bfe49aac680bfece33d4fdf6a246/frozenlist-1.8.0-cp314-cp314-musllinux_1_2_ppc64le.whl", hash = "sha256:154e55ec0655291b5dd1b8731c637ecdb50975a2ae70c606d100750a540082f7", size = 237308, upload-time = "2025-10-06T05:37:20.969Z" },
    { url = "https://files.pythonhosted.org/packages/11/99/ae3a33d5befd41ac0ca2cc7fd3aa707c9c324de2e89db0e0f45db9a64c26/frozenlist-1.8.0-cp314-cp314-musllinux_1_2_s390x.whl", hash = "sha256:4314debad13beb564b708b4a496020e5306c7333fa9a3ab90374169a20ffab30", size = 238210, upload-time = "2025-10-06T05:37:22.252Z" },
    { url = "https://files.pythonhosted.org/packages/b2/60/b1d2da22f4970e7a155f0adde9b1435712ece01b3cd45ba63702aea33938/frozenlist-1.8.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:073f8bf8becba60aa931eb3bc420b217bb7d5b8f4750e6f8b3be7f3da85d38b7", size = 231972, upload-time = "2025-10-06T05:37:23.5Z" },
    { url = "https://files.pythonhosted.org/packages/3f/ab/945b2f32de889993b9c9133216c068b7fcf257d8595a0ac420ac8677cab0/frozenlist-1.8.0-cp314-cp314-win32.whl", hash = "sha256:bac9c42ba2ac65ddc115d930c78d24ab8d4f465fd3fc473cdedfccadb9429806", size = 40536, upload-time = "2025-10-06T05:37:25.581Z" },
    { url = "https://files.pythonhosted.org/packages/59/ad/9caa9b9c836d9ad6f067157a531ac48b7d36499f5036d4141ce78c230b1b/frozenlist-1.8.0-cp314-cp314-win_amd64.whl", hash = "sha256:3e0761f4d1a44f1d1a47996511752cf3dcec5bbdd9cc2b4fe595caf97754b7a0", size = 44330, upload-time = "2025-10-06T05:37:26.928Z" },
    { url = "https://files.pythonhosted.org/packages/82/13/e6950121764f2676f43534c555249f57030150260aee9dcf7d64efda11dd/frozenlist-1.8.0-cp314-cp314-win_arm64.whl", hash = "sha256:d1eaff1d00c7751b7c6662e9c5ba6eb2c17a2306ba5e2a37f24ddf3cc953402b", size = 40627, upload-time = "2025-10-06T05:37:28.075Z" },
    { url = "https://files.pythonhosted.org/packages/c0/c7/43200656ecc4e02d3f8bc248df68256cd9572b3f0017f0a0c4e93440ae23/frozenlist-1.8.0-cp314-cp314t-macosx_10_13_universal2.whl", hash = "sha256:d3bb933317c52d7ea5004a1c442eef86f426886fba134ef8cf4226ea6ee1821d", size = 89238, upload-time = "2025-10-06T05:37:29.373Z" },
    { url = "https://files.pythonhosted.org/packages/d1/29/55c5f0689b9c0fb765055629f472c0de484dcaf0acee2f7707266ae3583c/frozenlist-1.8.0-cp314-cp314t-macosx_10_13_x86_64.whl", hash = "sha256:8009897cdef112072f93a0efdce29cd819e717fd2f649ee3016efd3cd885a7ed", size = 50738, upload-time = "2025-10-06T05:37:30.792Z" },
    { url = "https://files.pythonhosted.org/packages/ba/7d/b7282a445956506fa11da8c2db7d276adcbf2b17d8bb8407a47685263f90/frozenlist-1.8.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:2c5dcbbc55383e5883246d11fd179782a9d07a986c40f49abe89ddf865913930", size = 51739, upload-time = "2025-10-06T05:37:32.127Z" },
    { url = "https://files.pythonhosted.org/packages/62/1c/3d8622e60d0b767a5510d1d3cf21065b9db874696a51ea6d7a43180a259c/frozenlist-1.8.0-cp314-cp314t-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:39ecbc32f1390387d2aa4f5a995e465e9e2f79ba3adcac92d68e3e0afae6657c", size = 284186, upload-time = "2025-10-06T05:37:33.21Z" },
    { url = "https://files.pythonhosted.org/packages/2d/14/aa36d5f85a89679a85a1d44cd7a6657e0b1c75f61e7cad987b203d2daca8/frozenlist-1.8.0-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:92db2bf818d5cc8d9c1f1fc56b897662e24ea5adb36ad1f1d82875bd64e03c24", size = 292196, upload-time = "2025-10-06T05:37:36.107Z" },
    { url = "https://files.pythonhosted.org/packages/05/23/6bde59eb55abd407d34f77d39a5126fb7b4f109a3f611d3929f14b700c66/frozenlist-1.8.0-cp314-cp314t-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:2dc43a022e555de94c3b68a4ef0b11c4f747d12c024a520c7101709a2144fb37", size = 273830, upload-time = "2025-10-06T05:37:37.663Z" },
    { url = "https://files.pythonhosted.org/packages/d2/3f/22cff331bfad7a8afa616289000ba793347fcd7bc275f3b28ecea2a27909/frozenlist-1.8.0-cp314-cp314t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:cb89a7f2de3602cfed448095bab3f178399646ab7c61454315089787df07733a", size = 294289, upload-time = "2025-10-06T05:37:39.261Z" },
    { url = "https://files.pythonhosted.org/packages/a4/89/5b057c799de4838b6c69aa82b79705f2027615e01be996d2486a69ca99c4/frozenlist-1.8.0-cp314-cp314t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:33139dc858c580ea50e7e60a1b0ea003efa1fd42e6ec7fdbad78fff65fad2fd2", size = 300318, upload-time = "2025-10-06T05:37:43.213Z" },
    { url = "https://files.pythonhosted.org/packages/30/de/2c22ab3eb2a8af6d69dc799e48455813bab3690c760de58e1bf43b36da3e/frozenlist-1.8.0-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:168c0969a329b416119507ba30b9ea13688fafffac1b7822802537569a1cb0ef", size = 282814, upload-time = "2025-10-06T05:37:45.337Z" },
    { url = "https://files.pythonhosted.org/packages/59/f7/970141a6a8dbd7f556d94977858cfb36fa9b66e0892c6dd780d2219d8cd8/frozenlist-1.8.0-cp314-cp314t-musllinux_1_2_armv7l.whl", hash = "sha256:28bd570e8e189d7f7b001966435f9dac6718324b5be2990ac496cf1ea9ddb7fe", size = 291762, upload-time = "2025-10-06T05:37:46.657Z" },
    { url = "https://files.pythonhosted.org/packages/c1/15/ca1adae83a719f82df9116d66f5bb28bb95557b3951903d39135620ef157/frozenlist-1.8.0-cp314-cp314t-musllinux_1_2_ppc64le.whl", hash = "sha256:b2a095d45c5d46e5e79ba1e5b9cb787f541a8dee0433836cea4b96a2c439dcd8", size = 289470, upload-time = "2025-10-06T05:37:47.946Z" },
    { url = "https://files.pythonhosted.org/packages/ac/83/dca6dc53bf657d371fbc88ddeb21b79891e747189c5de990b9dfff2ccba1/frozenlist-1.8.0-cp314-cp314t-musllinux_1_2_s390x.whl", hash = "sha256:eab8145831a0d56ec9c4139b6c3e594c7a83c2c8be25d5bcf2d86136a532287a", size = 289042, upload-time = "2025-10-06T05:37:49.499Z" },
    { url = "https://files.pythonhosted.org/packages/96/52/abddd34ca99be142f354398700536c5bd315880ed0a213812bc491cff5e4/frozenlist-1.8.0-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:974b28cf63cc99dfb2188d8d222bc6843656188164848c4f679e63dae4b0708e", size = 283148, upload-time = "2025-10-06T05:37:50.745Z" },
    { url = "https://files.pythonhosted.org/packages/af/d3/76bd4ed4317e7119c2b7f57c3f6934aba26d277acc6309f873341640e21f/frozenlist-1.8.0-cp314-cp314t-win32.whl", hash = "sha256:342c97bf697ac5480c0a7ec73cd700ecfa5a8a40ac923bd035484616efecc2df", size = 44676, upload-time = "2025-10-06T05:37:52.222Z" },
    { url = "https://files.pythonhosted.org/packages/89/76/c615883b7b521ead2944bb3480398cbb07e12b7b4e4d073d3752eb721558/frozenlist-1.8.0-cp314-cp314t-win_amd64.whl", hash = "sha256:06be8f67f39c8b1dc671f5d83aaefd3358ae5cdcf8314552c57e7ed3e6475bdd", size = 49451, upload-time = "2025-10-06T05:37:53.425Z" },
    { url = "https://files.pythonhosted.org/packages/e0/a3/5982da14e113d07b325230f95060e2169f5311b1017ea8af2a29b374c289/frozenlist-1.8.0-cp314-cp314t-win_arm64.whl", hash = "sha256:102e6314ca4da683dca92e3b1355490fed5f313b768500084fbe6371fddfdb79", size = 42507, upload-time = "2025-10-06T05:37:54.513Z" },
    { url = "https://files.pythonhosted.org/packages/9a/9a/e35b4a917281c0b8419d4207f4334c8e8c5dbf4f3f5f9ada73958d937dcc/frozenlist-1.8.0-py3-none-any.whl", hash = "sha256:0c18a16eab41e82c295618a77502e17b195883241c563b00f0aa5106fc4eaa0d", size = 13409, upload-time = "2025-10-06T05:38:16.721Z" },
]

[[package]]
name = "fsspec"
version = "2026.7.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/00/78/f34251dadb8f3921264a1d9b8946f5e542014ee2614b285261b4e40e6775/fsspec-2026.7.0.tar.gz", hash = "sha256:c803c40f4cf860b49dea58ee3e1c33cb9c790520e233537e1340049f89b82a88", size = 317040, upload-time = "2026-07-28T16:34:51.052Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/fd/3c/6a2bf344106328fd04963664a60b9bb6496fc25df8e962fcdc1367285fb9/fsspec-2026.7.0-py3-none-any.whl", hash = "sha256:b57ddbafedfaef7018c1ecab32aa200a9d7ca26b77965f64e48b70061249d279", size = 206583, upload-time = "2026-07-28T16:34:49.538Z" },
]

[[package]]
name = "genbi"
version = "0.1.0"
source = { virtual = "." }

[package.dev-dependencies]
dev = [
    { name = "mypy" },
    { name = "pre-commit" },
    { name = "pytest" },
    { name = "pytest-asyncio" },
    { name = "pytest-cov" },
    { name = "ruff" },
```

```
]

[package.metadata]

[package.metadata.requires-dev]
dev = [
    { name = "mypy", specifier = ">=1.11" },
    { name = "pre-commit", specifier = ">=3.8" },
    { name = "pytest", specifier = ">=8.3" },
    { name = "pytest-asyncio", specifier = ">=0.24" },
    { name = "pytest-cov", specifier = ">=5.0" },
    { name = "ruff", specifier = ">=0.6" },
]

[[package]]
name = "genbi-backend"
version = "0.1.0"
source = { virtual = "backend" }
dependencies = [
    { name = "alembic" },
    { name = "asyncpg" },
    { name = "bcrypt" },
    { name = "fastapi" },
    { name = "gunicorn" },
    { name = "httpx" },
    { name = "langchain" },
    { name = "langchain-anthropic" },
    { name = "langfuse" },
    { name = "langgraph" },
    { name = "llama-index" },
    { name = "loguru" },
    { name = "mcp" },
    { name = "numpy" },
    { name = "openai" },
    { name = "opentelemetry-api" },
    { name = "opentelemetry-instrumentation-fastapi" },
    { name = "opentelemetry-sdk" },
    { name = "pandas" },
    { name = "pgvector" },
    { name = "prometheus-client" },
    { name = "pydantic" },
    { name = "pydantic-settings" },
    { name = "python-jose", extra = ["cryptography"] },
    { name = "python-multipart" },
    { name = "pyyaml" },
    { name = "redis" },
    { name = "sqlalchemy", extra = ["asyncio"] },
    { name = "uvicorn", extra = ["standard"] },
]

[package.optional-dependencies]
dev = [
    { name = "httpx" },
    { name = "mypy" },
    { name = "pytest" },
    { name = "pytest-asyncio" },
    { name = "pytest-cov" },
    { name = "ruff" },
]

[package.metadata]
requires-dist = [
    { name = "alembic", specifier = ">=1.14" },
    { name = "asyncpg", specifier = ">=0.30" },
    { name = "bcrypt", specifier = ">=5.0" },
    { name = "fastapi", specifier = ">=0.115" },
    { name = "gunicorn", specifier = ">=23.0" },
    { name = "httpx", specifier = ">=0.28" },
    { name = "httpx", marker = "extra == 'dev'", specifier = ">=0.28" },
    { name = "langchain", specifier = ">=0.3" },
    { name = "langchain-anthropic", specifier = ">=0.3" },
    { name = "langfuse", specifier = ">=2.55" },
    { name = "langgraph", specifier = ">=0.2" },
    { name = "llama-index", specifier = ">=0.12" },
    { name = "loguru", specifier = ">=0.7" },
    { name = "mcp", specifier = ">=1.5" },
    { name = "mypy", marker = "extra == 'dev'", specifier = ">=1.11" },
    { name = "numpy", specifier = ">=2.1" },
    { name = "openai", specifier = ">=1.58" },
    { name = "opentelemetry-api", specifier = ">=1.28" },
    { name = "opentelemetry-instrumentation-fastapi", specifier = ">=0.49" },
    { name = "opentelemetry-sdk", specifier = ">=1.28" },
    { name = "pandas", specifier = ">=2.2" },
    { name = "pgvector", specifier = ">=0.3" },
    { name = "prometheus-client", specifier = ">=0.21" },
    { name = "pydantic", specifier = ">=2.9" },
    { name = "pydantic-settings", specifier = ">=2.6" },
    { name = "pytest", marker = "extra == 'dev'", specifier = ">=8.3" },
    { name = "pytest-asyncio", marker = "extra == 'dev'", specifier = ">=0.24" },
    { name = "pytest-cov", marker = "extra == 'dev'", specifier = ">=5.0" },
    { name = "python-jose", extras = ["cryptography"], specifier = ">=3.3" },
    { name = "python-multipart", specifier = ">=0.0.17" },
    { name = "pyyaml", specifier = ">=6.0" },
    { name = "redis", specifier = ">=5.2" },
    { name = "ruff", marker = "extra == 'dev'", specifier = ">=0.6" },
    { name = "sqlalchemy", extras = ["asyncio"], specifier = ">=2.0.35" },
    { name = "uvicorn", extras = ["standard"], specifier = ">=0.32" },
]
provides-extras = ["dev"]

[[package]]
name = "googleapis-common-protos"
version = "1.75.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
```

```
    { name = "protobuf" },
]
sdist = { url = "https://files.pythonhosted.org/packages/b5/c8/f439cffde755cffa462bfbb156278fa6f9d09119719af9814b858fd4f81f/googleapis_common_protos-1.75.0.tar.gz", hash = "sha256:53a062ff3c32552fbd62c11fe23768b78e4ddf0494d5e5fd97d3f4689c75fbbd", size = 151035, upload-time = "2026-05-07T08:04:49.423Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e7/c8/e2645aa8ed02fd4c7a2f59d68783b65b1f3cbdfe39a6308e156509d1fee8/googleapis_common_protos-1.75.0-py3-none-any.whl", hash = "sha256:961ed60399c457ceb0ee8f285a84c870aabc9c6a832b9d37bb281b5bebde43ed", size = 300631, upload-time = "2026-05-07T08:03:30.345Z" },
]

[[package]]
name = "greenlet"
version = "3.5.4"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a3/74/b13368064b09053253555d3f2839cc2684d22d5aed0d2ccffbf7a6736558/greenlet-3.5.4.tar.gz", hash = "sha256:0232ae1de90a8e07867bb127d7a6ba2301e859145489f25cda8a6096dabe1d20", size = 206538, upload-time = "2026-07-22T12:47:14.468Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/f3/04/81bd731d6d1e3a469d9a4c36f5eb069bcf0cbb2d5d342c9fec22245b91fc/greenlet-3.5.4-cp312-cp312-macosx_11_0_universal2.whl", hash = "sha256:3d66250e8b09f182ede05490998c818b5961f7a3640332d44c4927caec7bbfe4", size = 295909, upload-time = "2026-07-22T11:38:09.261Z" },
    { url = "https://files.pythonhosted.org/packages/cc/dd/f5f22903a6ae70f5ea328ed0beaec92ad903f0e3b7d2845133b354abc4b8/greenlet-3.5.4-cp312-cp312-manylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:c90e930c9c192e5b3ee9fb8bcd920ea3926155e2e3ded39fc697323addecee17", size = 612011, upload-time = "2026-07-22T12:26:40.69Z" },
    { url = "https://files.pythonhosted.org/packages/8e/10/92a4a88d12b915d74ea5b6d288e4afefda4771647caa34442c156f7a454f/greenlet-3.5.4-cp312-cp312-manylinux_2_24_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:791fdfeeb9c6e0c7b10fa151bf110d2a6974866f13dcb5b1c7efae698245893a", size = 624299, upload-time = "2026-07-22T12:29:02.089Z" },
    { url = "https://files.pythonhosted.org/packages/6c/f9/03e26be3487c5238e81f2b84714959a86ea8515a869828cf41f4fc54b34e/greenlet-3.5.4-cp312-cp312-manylinux_2_24_s390x.manylinux_2_28_s390x.whl", hash = "sha256:b7c895310363f310361e0fe2072af85269d2a2a285cd04c0c59e79a5e3670dcf", size = 629603, upload-time = "2026-07-22T12:43:43.456Z" },
    { url = "https://files.pythonhosted.org/packages/50/6d/0b14bb9db2989f32cd9fe7f76afedea01ee8bee3f87c07e69f24adfe7e63/greenlet-3.5.4-cp312-cp312-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:f88193799d43dbf8c8a806d6405c9c52fe2af40bf75072a606357b33cc336c7f", size = 621541, upload-time = "2026-07-22T11:51:09.464Z" },
    { url = "https://files.pythonhosted.org/packages/57/6b/7c55ca72ef80d57c16c4a55210f82582622462dc4485799a30f4ec6f3372/greenlet-3.5.4-cp312-cp312-manylinux_2_39_riscv64.whl", hash = "sha256:13b980043cb1b3134e81ea469da1250ddcc6bfe6d245bbaa59168d9cdc8f228f", size = 432554, upload-time = "2026-07-22T12:39:51.379Z" },
    { url = "https://files.pythonhosted.org/packages/48/3d/25e9a2d9eb6b2e8b7ca4e80a3a26cb887cce6c8e0a87c921164f11bc5574/greenlet-3.5.4-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:b7a5f095767c4493afcd06067f2bb3b8716e3f3f9e92b99c88e7e99f885b3d4d", size = 1581444, upload-time = "2026-07-22T12:25:03.818Z" },
    { url = "https://files.pythonhosted.org/packages/b9/96/4c9bf2e2c408dcc0556edce69efa9f802e82223573c53240136a086821f1/greenlet-3.5.4-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:42afdc1ab5f66da8c586c32af9224a74a706b4f0ea0dc3a4188a0860a09c65c9", size = 1645842, upload-time = "2026-07-22T11:51:12.295Z" },
    { url = "https://files.pythonhosted.org/packages/b5/41/303ecb26a3a56122c0f4d4073ee078881847bd6b6f463ae0ec57ec20223b/greenlet-3.5.4-cp312-cp312-win_amd64.whl", hash = "sha256:60149df8f462d1b230038e6590c23c3b4768bb5d6c022b3b6e82532b34b0b8a3", size = 247169, upload-time = "2026-07-22T11:38:19.893Z" },
    { url = "https://files.pythonhosted.org/packages/a4/e3/ef56864b4c35fcb3eb3b41b869f6cc46f4cd3f5e2c68e74acde8ac433951/greenlet-3.5.4-cp312-cp312-win_arm64.whl", hash = "sha256:77d6ce04fed0d9aeed42e0f37923cc43eba9b027bdd9c34546bb4ccd143d0fe0", size = 245565, upload-time = "2026-07-22T11:38:27.061Z" },
    { url = "https://files.pythonhosted.org/packages/c0/9a/e51225dcd58713f16ccbdcc501a8da21098ea14515b7870f1f94459e5ff5/greenlet-3.5.4-cp313-cp313-macosx_11_0_universal2.whl", hash = "sha256:24e61b88cb7e1b1d794b32a10cc346ac779681d6d74ff137a3e0a444d2bf1f02", size = 294831, upload-time = "2026-07-22T11:38:53.389Z" },
    { url = "https://files.pythonhosted.org/packages/9f/ea/de50a50fadf979713ab18b46f22ad5ff5f2dcfc637a3ebdecf669801e1a5/greenlet-3.5.4-cp313-cp313-manylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:870d730fec833f5a06906a32596cc099b9161594642a92a520b7a88911c95356", size = 614619, upload-time = "2026-07-22T12:26:42.282Z" },
    { url = "https://files.pythonhosted.org/packages/db/c7/2aae27fea41205b8650294c301f042a2a4bb6155eea48c995b890a92f2c1/greenlet-3.5.4-cp313-cp313-manylinux_2_24_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:ec5ff0d1878df6af3bf9b638a5a92a7d5693291de77c91bff10fa48519c604ef", size = 627021, upload-time = "2026-07-22T12:29:03.445Z" },
    { url = "https://files.pythonhosted.org/packages/1b/80/fb4d4788bbc8e54761f1fc88533af9523a6e86299fa113d6e8a8503ed9fc/greenlet-3.5.4-cp313-cp313-manylinux_2_24_s390x.manylinux_2_28_s390x.whl", hash = "sha256:07bd44616608d873d06735b63ef1a88191d6ca57c8d291d6559c71bc14c0893c", size = 632845, upload-time = "2026-07-22T12:43:45.19Z" },
    { url = "https://files.pythonhosted.org/packages/eb/56/79fd826f9ccaae0b84e1b4ef68dabba5e105bb044ffcd448a0b782fcba9a/greenlet-3.5.4-cp313-cp313-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:d84d993f6e575c950d91a23c1345d18fe1a4310d447bf630849d7809196b52f0", size = 624002, upload-time = "2026-07-22T11:51:11.391Z" },
    { url = "https://files.pythonhosted.org/packages/42/e3/6086fa578ebb72772722cdc4bcd628459814b42e0c2db1e3cbd6552b3271/greenlet-3.5.4-cp313-cp313-manylinux_2_39_riscv64.whl", hash = "sha256:3529a8a933582ad19e224792cac7372489526576b75b4c124e8e4f29948f4861", size = 435053, upload-time = "2026-07-22T12:39:52.715Z" },
    { url = "https://files.pythonhosted.org/packages/0a/1a/27319f97e731298513dcba1a2e91b63e9d8811d9de22130f960b129b1bf1/greenlet-3.5.4-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:58023945f421093de5e6fa108c0985a8659d43f49e0216da25099369a121bcbd", size = 1581533, upload-time = "2026-07-22T12:25:05.322Z" },
    { url = "https://files.pythonhosted.org/packages/b1/6d/24240bf562e9786dd2799ee0a4a4dadb4ded22510f41b20245099159ac8c/greenlet-3.5.4-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:bae2728e1897aa8df8cb1af38cd48b3a743aefe29372de7b8b7a9f532501e69f", size = 1645781, upload-time = "2026-07-22T11:51:14.805Z" },
    { url = "https://files.pythonhosted.org/packages/c1/5a/442ab1a9ef7ca6bf7210e5397a95972206a91a31033a03c8900866a10039/greenlet-3.5.4-cp313-cp313-win_amd64.whl", hash = "sha256:ca5726c0b08ca35ae873557266a78b2c3f3b2b7d7401aa5ff886c2045dd0111c", size = 247133, upload-time = "2026-07-22T11:39:20.661Z" },
    { url = "https://files.pythonhosted.org/packages/3e/e6/9160210222386b1a378ff94db846b9508ca24a121cf684991561fdb69280/greenlet-3.5.4-cp313-cp313-win_arm64.whl", hash = "sha256:7c1303791d603080cac6fc3b34df51c3b75b723739c282c8029e48a0d241672f", size = 245500, upload-time = "2026-07-22T11:40:22.185Z" },
    { url = "https://files.pythonhosted.org/packages/a5/a7/6ab1d4f9cd548d15ab90da29947f2076100130bb179b0bde59f795a459e3/greenlet-3.5.4-cp314-cp314-macosx_11_0_universal2.whl", hash = "sha256:7e8afa5eac028f8140ceafe5ceec66e6aa127ddcb21452d2a564dcd2900b5f22", size = 295410, upload-time = "2026-07-22T11:40:35.747Z" },
    { url = "https://files.pythonhosted.org/packages/cd/7a/422f63b4715cbc0b24385305407adf38b48f6bb68b3e6b04090e994d0f5a/greenlet-3.5.4-cp314-cp314-manylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:73b37afe369021423ea53dd3123e04bffa7e93ac64429b9f50835b2e4fcae7cf", size = 661286, upload-time = "2026-07-22T12:26:43.8Z" },
    { url = "https://files.pythonhosted.org/packages/d0/31/5a1cac663bf5582190c5a714ef81364f03cde232227f39748f8ae4c11da5/greenlet-3.5.4-cp314-cp314-manylinux_2_24_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:3ef964f56dfcb6f9bbef2a190d9126795eac408716aeae47b5e7c73c32aafca9", size = 673517, upload-time = "2026-07-22T12:29:04.815Z" },
    { url = "https://files.pythonhosted.org/packages/9c/bf/250c2921c7b585dde12f5239e313ca2dcbc464d161ecca36e4e6ef21762d/greenlet-3.5.4-cp314-cp314-manylinux_2_24_s390x.manylinux_2_28_s390x.whl", hash = "sha256:cef589bc65fae02d10bca2ac341191c5b33acc2967892ebf4fcbd10eabb7a74c", size = 677968, upload-time = "2026-07-22T12:43:46.788Z" },
    { url = "https://files.pythonhosted.org/packages/15/4a/2a82a1e3f8aaca020853ac8d12211280ca2b231aa08ea39f636f1060c319/greenlet-3.5.4-cp314-cp314-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9c53ff01a5c53a40f2c16820ebc56d7c61a77f5fbe009dadd96292d5682f80f8", size = 670917, upload-time = "2026-07-22T11:51:13.589Z" },
    { url = "https://files.pythonhosted.org/packages/18/40/10bfcf6513558d82f7b95dd728001c63bd388259fe27d3e30ae01f103430/greenlet-3.5.4-cp314-cp314-manylinux_2_39_riscv64.whl", hash = "sha256:dfc41ae893d9ceaf22c824f2153a88b30651b20e8758c2cd9ac143f23640563c", size = 480643, upload-time = "2026-07-22T12:39:54.149Z" },
    { url = "https://files.pythonhosted.org/packages/68/b0/e379a152b17bfdfa95795af4049e37c0fd1b4d81f020d426db104ed07c77/greenlet-3.5.4-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:ecca4d80d55a01ad6b23b33262662956149fbb7b2c6be2910f1705921958cbf3", size = 1628478, upload-time = "2026-07-22T12:25:06.678Z" },
    { url = "https://files.pythonhosted.org/packages/5e/43/bffdfa64f7317f954c5c1230b5dd5922676ce198689a68c1ac1ed4b1b1a5/greenlet-3.5.4-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:2ffbc533e0eaf8e80d8471411646ab88fe58f641d508c0b02b24494479f4d9ec", size = 1692021, upload-time = "2026-07-22T11:51:17.008Z" },
    { url = "https://files.pythonhosted.org/packages/d0/11/f799f9637e2c6e9b0b716015e339040598b058cf7654dfc0d67468b177ed/greenlet-3.5.4-cp314-cp314-win_amd64.whl", hash = "sha256:305f69e6c4523d7f6979ed001cff4e5853c063e5da04880296603aa0227e544c", size = 248031, upload-time = "2026-07-22T11:40:11.007Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/05/75/625bcdd74d5e6b2dca1ecba3c3ac77bcf8a026c21a649a46cef23e421f97/greenlet-3.5.4-cp314-cp314-win_arm64.whl", hash = "sha256:f260930bbbbcf9caee661211235a5111c86dfe5832fdf6ae4570da1e0995320f", size = 246892, upload-time = "2026-07-22T11:40:27.357Z" },
    { url = "https://files.pythonhosted.org/packages/ec/69/35c62ed49c320cb4d98e14698ccca5467d3bfe683984172be9cb564d9ce3/greenlet-3.5.4-cp314-cp314t-macosx_11_0_universal2.whl", hash = "sha256:41ddab54e4b238f4a6c323f39b4e59e176affd5a94d461a9fb7583dac74240a3", size = 305571, upload-time = "2026-07-22T11:40:31.659Z" },
    { url = "https://files.pythonhosted.org/packages/5b/6c/64d60216b3640dcb0b62d913dd9e0d80030c09115bb2e4ba70c95d10ca45/greenlet-3.5.4-cp314-cp314t-manylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:b3dabe3e2809013052c68bdf0b7fa5f5f2859c43a80803131ad61af9cabd7867", size = 672568, upload-time = "2026-07-22T12:26:45.298Z" },
    { url = "https://files.pythonhosted.org/packages/5c/de/ba3ab0a96292e53039530333b0d2ae18d9e508f3a325cd7bf15f8172944c/greenlet-3.5.4-cp314-cp314t-manylinux_2_24_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:27d3f00718634d4520a3a150154ac5da36f257869d41321953375b90bfbbc72c", size = 680076, upload-time = "2026-07-22T12:29:06.125Z" },
    { url = "https://files.pythonhosted.org/packages/ae/db/24a10af12bf8e639cec46c38b9ce1a282543ba42ff4fb0b31a970f1ab603/greenlet-3.5.4-cp314-cp314t-manylinux_2_24_s390x.manylinux_2_28_s390x.whl", hash = "sha256:39169a11d87a6a263afda3e9a27d1df16d0f919d40a4837cc73986c9884c0dd8", size = 681690, upload-time = "2026-07-22T12:43:48.109Z" },
    { url = "https://files.pythonhosted.org/packages/a5/be/aeada79083c6f1c15f45d77a332f9c441af263ee298e3eb17522cd337d22/greenlet-3.5.4-cp314-cp314t-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:cbd60b5763c6543c1827e48faaf14ea9bfbad245f52b1a4d76a2a2d8884c6c66", size = 676733, upload-time = "2026-07-22T11:51:16.027Z" },
    { url = "https://files.pythonhosted.org/packages/f4/60/44a2eca7b9fd71ae0fae7ff184da1cd3169d176652b97aa1cffcbb0ef961/greenlet-3.5.4-cp314-cp314t-manylinux_2_39_riscv64.whl", hash = "sha256:bd3d1145f603b2db19feb9078c2e6855eb7c67e15580c010ed815cee519b86fd", size = 510263, upload-time = "2026-07-22T12:39:55.678Z" },
    { url = "https://files.pythonhosted.org/packages/e9/10/2392fc3a98948652ef5fd1e7275c04f861dd13f74b78a2b4309f4ee4d090/greenlet-3.5.4-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:f00f910f0e7b35416c63b23ad78b769aeccfc1775f712b43c4ee525624a2eef7", size = 1637327, upload-time = "2026-07-22T12:25:07.879Z" },
    { url = "https://files.pythonhosted.org/packages/55/c6/e7237a3dfa1f205ed0d9ea1e46d70bd2811b32d516266399fb59d28ab90a/greenlet-3.5.4-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:91c26423753b92caf41ab3f98fd547d7374d4d9fc2d85be041886c1579d9255e", size = 1697493, upload-time = "2026-07-22T11:51:19.214Z" },
    { url = "https://files.pythonhosted.org/packages/55/e3/4ba8154ba2a3d43729e499f72471b4b5c993f3826d3e24da81d5f06d6572/greenlet-3.5.4-cp314-cp314t-win_amd64.whl", hash = "sha256:ee032b91fd8ec29ec6c4cea2b8c561b178435134bd0752c7334b94e9c736c132", size = 251637, upload-time = "2026-07-22T11:40:37.44Z" },
    { url = "https://files.pythonhosted.org/packages/90/03/e3f96dfc100261a29545ddc8270cafe58f9195b6651466910e820910de77/greenlet-3.5.4-cp315-cp315-macosx_11_0_universal2.whl", hash = "sha256:178111881dd7a6c946471fda85485ec796e1043c2b939f694b096e2ecf986809", size = 296076, upload-time = "2026-07-22T11:39:38.364Z" },
    { url = "https://files.pythonhosted.org/packages/a4/3d/da52d208e5c977bce8667e784729e584e38b5785f4c1ec0f4c836e9a1c42/greenlet-3.5.4-cp315-cp315-manylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:d92df08dd65fede97fc37aad36c2e9dcda3b31c467f8e0c2c096456cb818e927", size = 666870, upload-time = "2026-07-22T12:26:46.691Z" },
    { url = "https://files.pythonhosted.org/packages/2d/8a/7e6dee25cb8a8cf9b362c8e597cc269593378bd916f16c736c059a52e85a/greenlet-3.5.4-cp315-cp315-manylinux_2_24_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:99e8f8c4ebc4fd80aa26c1280ae9ad43a0976e786349703a181cf0bae60413e5", size = 677678, upload-time = "2026-07-22T12:29:07.508Z" },
    { url = "https://files.pythonhosted.org/packages/51/a7/dafc7415d430b0a43a16396eb49ecb3b62fd720877fb259cc4dcfaf5f31e/greenlet-3.5.4-cp315-cp315-manylinux_2_24_s390x.manylinux_2_28_s390x.whl", hash = "sha256:1f17e362d78e37559e0506c5a7d066bdd45073c36a0127a543e8a0df27242ff3", size = 681428, upload-time = "2026-07-22T12:43:49.623Z" },
    { url = "https://files.pythonhosted.org/packages/6c/21/5a38699fa45de749e3857d93b8f07e4c20489e77c2d35d915a2e1c456606/greenlet-3.5.4-cp315-cp315-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:394de08dad5ffcb1f50c2159d93e398d9d2da3ed437645eaa54771fa720db9f0", size = 676067, upload-time = "2026-07-22T11:51:18.163Z" },
    { url = "https://files.pythonhosted.org/packages/2e/d9/6298f3432de301d4718766cf934bd73c418c73f81fbb77247319364b0d96/greenlet-3.5.4-cp315-cp315-manylinux_2_39_riscv64.whl", hash = "sha256:cd320d998cbaa032932830448e39abf3c6a12901295e386e8114db926e10cffb", size = 487446, upload-time = "2026-07-22T12:39:57.044Z" },
    { url = "https://files.pythonhosted.org/packages/5e/ba/863116ab8ff1ca7a729e327800268939d182db47aa433db70e216e7d9194/greenlet-3.5.4-cp315-cp315-musllinux_1_2_aarch64.whl", hash = "sha256:c883d61f2282d72c767a14936641b3efcbde9d82f1080712aaea0b1d3126cb88", size = 1633489, upload-time = "2026-07-22T12:25:09.605Z" },
    { url = "https://files.pythonhosted.org/packages/fa/06/7466ced82818d6132462d7f26b3f83c66ea15d2b193a6c0088d558ed7d95/greenlet-3.5.4-cp315-cp315-musllinux_1_2_x86_64.whl", hash = "sha256:2a924f15d17957e252a810acefcb5942f5ca712298e8b6fcaed9a307d357522c", size = 1696584, upload-time = "2026-07-22T11:51:21.304Z" },
    { url = "https://files.pythonhosted.org/packages/f9/4d/55b638489260065de9ffce606c8b5d04507bef705de4b212a0c3d6a1a0df/greenlet-3.5.4-cp315-cp315-win_amd64.whl", hash = "sha256:ed17e5f3420360d5b459de8462efb52060399a5326a613d4cde31cef63ef95da", size = 248297, upload-time = "2026-07-22T11:42:04.055Z" },
    { url = "https://files.pythonhosted.org/packages/bb/08/9dd4ae635da93d41dc268bc34bd62a9d711ed8b8825c5d22ac910c7d6e6d/greenlet-3.5.4-cp315-cp315-win_arm64.whl", hash = "sha256:f908898d6fa484ce4b6f447ce70ea99b52c503fee419e53cf74d60a16bc9e667", size = 247423, upload-time = "2026-07-22T11:44:00.764Z" },
    { url = "https://files.pythonhosted.org/packages/19/66/7c87ed9cdbf1d49c2c6cd1c7b9dd4d16c33b24235ca03972293a1876b30c/greenlet-3.5.4-cp315-cp315t-macosx_11_0_universal2.whl", hash = "sha256:1833637f17d5e7472548a48575c394fe39f1b1890d676d162d86593610f44d8c", size = 306487, upload-time = "2026-07-22T11:41:25.118Z" },
    { url = "https://files.pythonhosted.org/packages/5b/05/0a4201e7c0054866eefc05da234f236dd4c950d0fbf9ca0517141f01b269/greenlet-3.5.4-cp315-cp315t-manylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:12cda9122e03341f1cb6b8207a19d7a9d375e52f1b4e9243918375f40fd7b4b9", size = 676479, upload-time = "2026-07-22T12:26:48.129Z" },
    { url = "https://files.pythonhosted.org/packages/3d/c0/4b6b8c5a3aec70f0649cd89662d120fdd6421e2bdc8e3b15c3ab5ec568d8/greenlet-3.5.4-cp315-cp315t-manylinux_2_24_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:d83ae0e32d14957ab7170785a20f582635c8474deab1bfbb552b17e769a6ce25", size = 684321, upload-time = "2026-07-22T12:29:08.925Z" },
    { url = "https://files.pythonhosted.org/packages/88/15/0b167aeea95285b0e654ddce651922f666c089363c2ec528ca8b9a9ba74f/greenlet-3.5.4-cp315-cp315t-manylinux_2_24_s390x.manylinux_2_28_s390x.whl", hash = "sha256:123aa379c962ed5fe90a880327e0c3066124ac64ec99e12a238be9fd8eb3db3d", size = 685995, upload-time = "2026-07-22T12:43:50.993Z" },
    { url = "https://files.pythonhosted.org/packages/24/c9/b49c31c9a972eee91e260445770e922244a7efc542697f51a012ec046d0f/greenlet-3.5.4-cp315-cp315t-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9f1467de1bb767f75db0aa34c195e3a496d8d1278c796e70c24ce205d3e99cde", size = 681293, upload-time = "2026-07-22T11:51:20.43Z" },
    { url = "https://files.pythonhosted.org/packages/de/90/c023ec337f32ff505be7db759c80d98f0532bb94d0c6fa13645efe9bee2e/greenlet-3.5.4-cp315-cp315t-manylinux_2_39_riscv64.whl", hash = "sha256:adf2244d7f69409925a8f22ed22cc5f93cdfe5c9dc87ff3476be2c2aaae61a05", size = 516928, upload-time = "2026-07-22T12:39:58.359Z" },
    { url = "https://files.pythonhosted.org/packages/95/6f/7f2d4653770500eee667866016d42d7a68e3d3462f80df6b8e3fcd48a0eb/greenlet-3.5.4-cp315-cp315t-musllinux_1_2_aarch64.whl", hash = "sha256:0fa53040b78b578120eecdc0265e3f1051487cc425d11a2b7c761daadf4feaa8", size = 1642474, upload-time = "2026-07-22T12:25:10.819Z" },
    { url = "https://files.pythonhosted.org/packages/5a/d8/8cba31036a4caae448087ba5d150660ab03b4a0f54d9150f6495a3be7262/greenlet-3.5.4-cp315-cp315t-musllinux_1_2_x86_64.whl", hash = "sha256:60e0bc961d367df506660e9ac0177a76bc6d81305300704b0977d1634f76efe2", size = 1701012, upload-time = "2026-07-22T11:51:23.17Z" },
    { url = "https://files.pythonhosted.org/packages/e3/cd/3f77a4cce3bae631b08eb52f53a82a976669600337e21dfdba811cb50267/greenlet-3.5.4-cp315-cp315t-win_amd64.whl", hash = "sha256:f680e549edb3eaf21eea4e7fe101e15ec180c74b7879ab46adc080f22d4015d2", size = 251977, upload-time = "2026-07-22T11:41:38.125Z" },
    { url = "https://files.pythonhosted.org/packages/93/e8/65e8707d00fe2a49bf12f609a9b2b39ba6dd23c2810eacad877c4fc94bfe/greenlet-3.5.4-cp315-cp315t-win_arm64.whl", hash = "sha256:08fc36de8442d5c3e95b044550dbea9bf144d31ec0cc58e36fb241cb6ef6a994", size = 250538, upload-time = "2026-07-22T11:40:17.985Z" },
]

[[package]]
name = "griffe"
version = "2.1.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "griffecli" },
    { name = "griffelib" },
]
sdist = { url = "https://files.pythonhosted.org/packages/00/44/63913c007814cab5ba9d36f25ad40dfc640c2e2931d195bd2d05f774a5d6/griffe-2.1.0.tar.gz", hash = "sha256:c58845df5a364feaabd05ee8c767b97b03e478da8aa18b9923553c812fb0d955", size = 244879, upload-time = "2026-06-19T12:05:41.262Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/6b/fb/3c65d392feae6c36dc2b55a14dd8270b7b35a3171c93b71a4d2ee4abf241/griffe-2.1.0-py3-none-any.whl", hash = "sha256:2ccdab17fb9cd76f278d7b5611cfc8f68cbe846d8d48df63dff80b62ecfa6f65", size = 5140, upload-time = "2026-06-19T12:05:39.913Z" },
```

```
]

[[package]]
name = "griffecli"
version = "2.1.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "colorama" },
    { name = "griffelib" },
]
sdist = { url = "https://files.pythonhosted.org/packages/8e/c6/90f85d47af96300d629b38c25b71aad9467a620cac964a39280e822efc8a/griffecli-2.1.0.tar.gz", h
ash = "sha256:2ff68dbee9395fdb668b10374c51683392d697b226ac60159798f4add1ee716c", size = 56913, upload-time = "2026-06-19T12:05:43.119Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/80/2f/513232ec1d5f5da182e4ce45a11427e37dd210844a9e2bca451fc9661fb3/griffecli-2.1.0-py3-none-any.w
hl", hash = "sha256:6e22b1423d562ddc510997b4be1fe89de59e19dcff78831c0f4bfc3b8134a718", size = 9500, upload-time = "2026-06-19T12:05:37.517Z" },
]

[[package]]
name = "griffelib"
version = "2.1.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/33/e4/8d187ea29c2e30b3a09505c567513077d6117861bde1fbd997a167f262ec/griffelib-2.1.0.tar.gz", h
ash = "sha256:762a186d2c6fd6794d4ea20d428d597ffb857cb56b66421651cbba15bdd5e813", size = 216234, upload-time = "2026-06-19T12:05:42.278Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e4/d3/5268aeabf2ad82658c4e2ff3a060648d0f02f3926cb53247c0e4d0dab49e/griffelib-2.1.0-py3-none-any.w
hl", hash = "sha256:cc7b3d2d2865ad0b909fcc38086e3f554b5ea7acbaa7bbb7ecaa3f5dfb7d9f00", size = 142560, upload-time = "2026-06-19T12:05:38.742Z" },
]

[[package]]
name = "gunicorn"
version = "26.0.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "packaging" },
]
sdist = { url = "https://files.pythonhosted.org/packages/6d/b7/a4a3f632f823e432ce6bc65f62961b7980c898c77f075a2f7118cb3846fe/gunicorn-26.0.0.tar.gz", h
ash = "sha256:ca9346f85e3a4aeeb64d491045c16b9a35647abd37ea15efe53080eb8b090baf", size = 727286, upload-time = "2026-05-05T06:38:25.529Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e6/40/9c2384fc2be4ad25dd4a49decd5ad9ea5a3639814c11bd40ab77cb9f0a14/gunicorn-26.0.0-py3-none-any.w
hl", hash = "sha256:40233d26a5f0d1872916188c276e21641155111c2853f0c2cd55260aec0d24fc", size = 212009, upload-time = "2026-05-05T06:38:23.007Z" },
]

[[package]]
name = "h11"
version = "0.16.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/01/ee/02a2c011bdab74c6fb3c75474d40b3052059d95df7e73351460c8588d963/h11-0.16.0.tar.gz", hash =
 "sha256:4e35b956cf45792e4caa5885e69fba00bdbc6ffafbfa020300e549b208ee5ff1", size = 101250, upload-time = "2025-04-24T03:35:25.427Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/04/4b/29cac41a4d98d144bf5f6d33995617b185d14b22401f75ca86f384e87ff1/h11-0.16.0-py3-none-any.whl",
hash = "sha256:63cf8bbe7522de3bf65932fda1d9c2772064ffb3dae62d55932da54b31cb6c86", size = 37515, upload-time = "2025-04-24T03:35:24.344Z" },
]

[[package]]
name = "httpcore"
version = "1.0.9"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "certifi" },
    { name = "h11" },
]
sdist = { url = "https://files.pythonhosted.org/packages/06/94/82699a10bca87a5556c9c59b5963f2d039dbd239f25bc2a63907a05a14cb/httpcore-1.0.9.tar.gz", ha
sh = "sha256:6e34463af53fd2ab5d807f399a9b45ea31c3dfa2276f15a2c3f00afff6e176e8", size = 85484, upload-time = "2025-04-24T22:06:22.219Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/7e/f5/f66802a942d491edb555dd61e3a9961140fd64c90bce1eafd741609d334d/httpcore-1.0.9-py3-none-any.wh
l", hash = "sha256:2d400746a40668fc9dec9810239072b40b4484b640a8c38fd654a024c7a1bf55", size = 78784, upload-time = "2025-04-24T22:06:20.566Z" },
]

[[package]]
name = "httpcore2"
version = "2.9.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "h11" },
    { name = "truststore" },
]
sdist = { url = "https://files.pythonhosted.org/packages/39/a8/20ed1ed79cbc2ecdf5301c0968ab7c85547212e2a7bd126ddd2d986e206e/httpcore2-2.9.1.tar.gz", h
ash = "sha256:4d8acbf8b306f48c9d6046591fd5ba4037d1b1b1000d140fc2c3eab1e9a0c0e2", size = 67089, upload-time = "2026-07-24T09:21:03.867Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/9f/fb/46c52b781975c335a2bcf1072c7bbc007cbdc8d674217f5ee1daba2c848b/httpcore2-2.9.1-py3-none-any.w
hl", hash = "sha256:6182472379e855fe4221246a2bb7ecede403bc61c6798062ae1787d051ccde26", size = 82809, upload-time = "2026-07-24T09:21:01.178Z" },
]

[[package]]
name = "httptools"
version = "0.8.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/43/e5/d471fcb0e14523fe1c3f4ba58ca52480e7bd70ad7109a3846bc75892f7fb/httptools-0.8.0.tar.gz", h
ash = "sha256:6b2a32f18d97e16e90827d7a819ffa8dbd8cc245fc4e1fa9d1095b54ef4bd999", size = 271342, upload-time = "2026-05-25T22:17:48.841Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/14/88/1d21a36da8f5cb0fa49eafd4b169eba5608d57e75bbcf61845cbc6243216/httptools-0.8.0-cp312-cp312-ma
cosx_10_13_universal2.whl", hash = "sha256:880490234c10f70a9830743097e8958d6e4b9f5a0ffc24515023afeef984054d", size = 208247, upload-time = "2026-05-25
T22:17:07.843Z" },
    { url = "https://files.pythonhosted.org/packages/a5/42/cc4feea2945cb3051038f090c9b36bd5b8a9d7f5a894a506a8983e33fd1c/httptools-0.8.0-cp312-cp312-ma
cosx_11_0_arm64.whl", hash = "sha256:5931891fb7b441b8a3853cf1b85c82c903defce084dd5f6771ca46e31bf862c5", size = 113064, upload-time = "2026-05-25T22:17
:09.136Z" },
    { url = "https://files.pythonhosted.org/packages/e3/a6/febbb8b8db0f58b38e44ad6cb946e6a255ae49b55f2e8543408fb7501ccd/httptools-0.8.0-cp312-cp312-ma
nylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:b15fc622b0f869d19207c4089a501d9bcc63ca5e071ffdd2f03f922df882dcb2", siz
e = 523851, upload-time = "2026-05-25T22:17:10.106Z" },
    { url = "https://files.pythonhosted.org/packages/b7/e4/f90a0df0b83beff265b7e3b65f2a4cefd95792d4be0ac3e16049f2acd3c2/httptools-0.8.0-cp312-cp312-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:425f83884fd6343828d8c565f046cb72b6d19063f6924093e11bcd8e1548cd0
9", size = 518842, upload-time = "2026-05-25T22:17:11.218Z" },
    { url = "https://files.pythonhosted.org/packages/9e/2d/0c9ac76dd2c893841fbf6498d6acec4f2442e1b7067f6e3e316a80e494e8/httptools-0.8.0-cp312-cp312-mu
sllinux_1_2_aarch64.whl", hash = "sha256:ef7c3c97f4311c7be57e2986629df89d49cb434dbff78eafcd48c2bff986b15a", size = 501238, upload-time = "2026-05-25T2
2:17:12.728Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/ca/42/906adc91ae3a5fa9c59c0a2f21c139725bd7e5b41ae6acd485cd14123ebf/httptools-0.8.0-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:a1afd7c9fbff0d9f5d489c4ce2768bd09c84a46ddefc7161e6aa82ae35c85745", size = 509567, upload-time = "2026-05-25T22:17:13.842Z" },
    { url = "https://files.pythonhosted.org/packages/05/0b/4240efeb672751ee5b9b380cb0e3fdc050bc05f68adc7a8aefc4fcd9a69a/httptools-0.8.0-cp312-cp312-win_amd64.whl", hash = "sha256:cd96f29b4bab1d42fa6e3d008711c75e0f79e94e06827330160e3a304227f150", size = 90918, upload-time = "2026-05-25T22:17:15.155Z" },
    { url = "https://files.pythonhosted.org/packages/5e/e5/8cfcabc5546e8022f168be28bcdaa128a240a0befdd03b59d558b4f18bd6/httptools-0.8.0-cp313-cp313-macosx_10_13_universal2.whl", hash = "sha256:614ceea8ea606848bece2338ac03b3ce5324bcb4be8dc7d377ed708012fa4db8", size = 205148, upload-time = "2026-05-25T22:17:16.333Z" },
    { url = "https://files.pythonhosted.org/packages/2a/0e/0fb14848c19a686c8062ff9067c1a48793e3224b47bc5b201535b6036fce/httptools-0.8.0-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:2d689918c15a013c65ef52d9fd495d766893ab831a2c8d89f2ac5940a5df847c", size = 111368, upload-time = "2026-05-25T22:17:17.586Z" },
    { url = "https://files.pythonhosted.org/packages/2e/1b/46f1cecf06b9bbde8e4b8c88034ac7908989e5ff7a3a388ef38392949c1f/httptools-0.8.0-cp313-cp313-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:eb3028cca2fc0a6d720e52ef61d8ebb62fcbfeb1de56874546d858d3f25a26b7", size = 486447, upload-time = "2026-05-25T22:17:18.564Z" },
    { url = "https://files.pythonhosted.org/packages/77/00/258bfc0837221f81d9725c45f9b948a6a6b2994a147a4fb66e85100c668f/httptools-0.8.0-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:88bdd940f2b5d487b4d032c6afa5489a7dc4694410d43de3c38c4fb3af0dc45d", size = 482448, upload-time = "2026-05-25T22:17:19.912Z" },
    { url = "https://files.pythonhosted.org/packages/04/ab/d1cef3b5523f4d272a70f42a776c3169a2dddfe3a54de4b2ce4a36341528/httptools-0.8.0-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:6a43c9dd399758ccc0531acb0a3c4a6c299ee893ee9400e9c893b7bdcfae0681", size = 464460, upload-time = "2026-05-25T22:17:20.882Z" },
    { url = "https://files.pythonhosted.org/packages/ce/48/5d1d072442277bb2b3434e0e60690b8e8c23840ef7de8b6ea54040a536d3/httptools-0.8.0-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:0770728beb05094c809b98e814edff5fef69d26ad7d21185f2f6d5884a0ba683", size = 471312, upload-time = "2026-05-25T22:17:22.085Z" },
    { url = "https://files.pythonhosted.org/packages/0d/66/b96623b27e51a68199ef4efdda0613cced9233fe3062ac74e50749c5ad37/httptools-0.8.0-cp313-cp313-win_amd64.whl", hash = "sha256:7685df791fad561384bfb139e77fde27a1ffd93134e016f95a0db424ffbf77b1", size = 90117, upload-time = "2026-05-25T22:17:23.074Z" },
    { url = "https://files.pythonhosted.org/packages/1a/12/fa3fbf5f9517b273edea2dc982aa82a8c634091e67c590792b729017bc6f/httptools-0.8.0-cp314-cp314-macosx_10_13_universal2.whl", hash = "sha256:de242a49b5d18e0a8776e654e9f6bf6d89f3875a5c35b425a0e7ce940feb3fd6", size = 206183, upload-time = "2026-05-25T22:17:24.004Z" },
    { url = "https://files.pythonhosted.org/packages/30/fc/5e7c4cb443370f2090a3aba0453a07384d29ff66b7435bb90e77e1037599/httptools-0.8.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:159e9ab5f701ccd42e555a12f1ad8ff69702910fc1c996cf2bb66e5fcb7a231b", size = 112079, upload-time = "2026-05-25T22:17:25.216Z" },
    { url = "https://files.pythonhosted.org/packages/ba/53/771bd891eb0f236f32145d6a1775777ec85745f3cc983a1f23d1a3b8ddfe/httptools-0.8.0-cp314-cp314-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:c4a9f1707e4823d54dfec6c33fa3697d302aed536ed352a7ebb5a061ddb869d0", size = 481596, upload-time = "2026-05-25T22:17:26.186Z" },
    { url = "https://files.pythonhosted.org/packages/62/42/94e15bc68ce3d423243c45d7f1b0c7561f13844f97dc52ae23182fb65628/httptools-0.8.0-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:d76ad7b951387e3632c8716a9bb03ac5b45c5f16119aa409db0459520887944e", size = 480865, upload-time = "2026-05-25T22:17:27.542Z" },
    { url = "https://files.pythonhosted.org/packages/1c/7c/fe2980fc03723272e30f135b62360b075f513dfe7cc73aef36c7f04012bd/httptools-0.8.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:a3b7387147361c3fd47a0bde763c5c91b5b4cd4dc9989b8ece84ff436c99843b", size = 463189, upload-time = "2026-05-25T22:17:28.546Z" },
    { url = "https://files.pythonhosted.org/packages/15/1b/47fc5fff68acd1bfa20b4734059c9a06cadb88119dcd5258b5b0d21d91c8/httptools-0.8.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:f256d6ce930c52ca1cb2a960b7da03548c454e7d28b06059ad41bfe789036ce0", size = 466610, upload-time = "2026-05-25T22:17:29.816Z" },
    { url = "https://files.pythonhosted.org/packages/60/bd/07b13c93ffd9bec9546e0d43f8e19378dd696dbd278511406bc07371ef1f/httptools-0.8.0-cp314-cp314-win_amd64.whl", hash = "sha256:19d1ee275bb59ba2643ba9a3a1e51cc0c788caf2b8df506368e03f56fdd08527", size = 92705, upload-time = "2026-05-25T22:17:31.133Z" },
    { url = "https://files.pythonhosted.org/packages/fd/c4/121648f68ce066d7bd762d6b6d97e620847642d38d54f3d90ff11d947629/httptools-0.8.0-cp314-cp314t-macosx_10_13_universal2.whl", hash = "sha256:de1ed58a974e75d56560acc7e7fed01a454994429456f65209789992e41f2568", size = 215023, upload-time = "2026-05-25T22:17:32.401Z" },
    { url = "https://files.pythonhosted.org/packages/b9/b0/312a062ae741ae3e8baa8c8bf20be81b2e67337b259ab4349bebc7b6142e/httptools-0.8.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:e93c227b595c6926c1acee96891dd9da4be338cfbe82e5cd3bb9d8dd7dc4ac0b", size = 117405, upload-time = "2026-05-25T22:17:33.742Z" },
    { url = "https://files.pythonhosted.org/packages/fc/37/fccd705f795386bb05bf413012fecff2a33e5aa8c2f069096de3e9fd8702/httptools-0.8.0-cp314-cp314t-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:2a021c3a8e65cc125390d72f59b968afca3bdcaff25bd67965e0a055a14946ca", size = 558497, upload-time = "2026-05-25T22:17:34.732Z" },
    { url = "https://files.pythonhosted.org/packages/bd/39/f172e8003576de35f5ba77ff417cf0e34429d35dc014deef15afa337a72c/httptools-0.8.0-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:48774d39cbb70e2b1f71f88852a3087ae1d3a1eb80482bb48c13067ab080c14f", size = 571585, upload-time = "2026-05-25T22:17:35.813Z" },
    { url = "https://files.pythonhosted.org/packages/3e/b9/f5564760af99f3dbbf3f9104dc00e5da27e96cf433c6bdcf77617f70bf3f/httptools-0.8.0-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:88eead8ec8680a9f146c655bc88445a325bd7921cfd8194c7337e9467282427d", size = 543297, upload-time = "2026-05-25T22:17:37.08Z" },
    { url = "https://files.pythonhosted.org/packages/99/67/8d9f2c313618e161b82f3873188e7196126da1d6e29688df40eb3997c77a/httptools-0.8.0-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:2c032fa028f46871ec7e1fc59fc15e8023eab3e6bbe6ece786a1611719a5d081", size = 539535, upload-time = "2026-05-25T22:17:38.032Z" },
    { url = "https://files.pythonhosted.org/packages/48/63/b906c01e53f50d432c0defe43ce52764a111dc1bdd028bafbeb54dcfd008/httptools-0.8.0-cp314-cp314t-win_amd64.whl", hash = "sha256:384c17174464c8e873398b7af24f0b1f44d992c820328413951a625323155d77", size = 108209, upload-time = "2026-05-25T22:17:39.473Z" },
]

[[package]]
name = "httpx"
version = "0.28.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "certifi" },
    { name = "httpcore" },
    { name = "idna" },
]
sdist = { url = "https://files.pythonhosted.org/packages/b1/df/48c586a5fe32a0f01324ee087459e112ebb7224f646c0b5023f5e79e9956/httpx-0.28.1.tar.gz", hash = "sha256:75e98c5f16b0f35b567856f597f06ff2270a374470a5c2392242528e3e3e42fc", size = 141406, upload-time = "2024-12-06T15:37:23.222Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/2a/39/e50c7c3a983047577ee07d2a9e53faf5a69493943ec3f6a384bdc792deb2/httpx-0.28.1-py3-none-any.whl", hash = "sha256:d909fcccc110f8c7faf814ca82a9a4d816bc5a6dbfea25d6591d6985b8ba59ad", size = 73517, upload-time = "2024-12-06T15:37:21.509Z" },
]

[[package]]
name = "httpx2"
version = "2.9.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "httpcore2" },
    { name = "idna" },
    { name = "truststore" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/21/14/38128fbafd7e0ed41d874df6c9a653d47c2d111cfe59e2b4ac95161b4abd/httpx2-2.9.1.tar.gz", hash = "sha256:1932a768737e3666291582833da748cc4e563c337cf96706fccc04fa6e58764a", size = 95458, upload-time = "2026-07-24T09:21:04.972Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/13/b8/cfd91c4ab9134d386d48f0b6ac662ff3d4be6efdee59ee1c67ebc3c0487c/httpx2-2.9.1-py3-none-any.whl", hash = "sha256:1820fe14a9ab1107bfeff39259987429450b070ec0ff38cc87eb0d8c97fdc71a", size = 91191, upload-time = "2026-07-24T09:21:02.6Z" },
]
```

```
[[package]]
name = "identify"
version = "2.6.19"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/52/63/51723b5f116cc04b061cb6f5a561790abf249d25931d515cd375e063e0f4/identify-2.6.19.tar.gz", hash = "sha256:6be5020c38fcb07da56c53733538a3081ea5aa70d36a156f83044bfbf9173842", size = 99567, upload-time = "2026-04-17T18:39:50.265Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/94/84/d9273cd09688070a6523c4aee4663a8538721b2b755c4962aafae0011e72/identify-2.6.19-py2.py3-none-any.whl", hash = "sha256:20e6a87f786f768c092a721ad107fc9df0eb89347be9396cadf3f4abbd1fb78a", size = 99397, upload-time = "2026-04-17T18:39:49.221Z" },
]

[[package]]
name = "idna"
version = "3.18"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/cd/63/9496c57188a2ee585e0f1db071d75089a11e98aa86eb99d9d7618fc1edce/idna-3.18.tar.gz", hash = "sha256:ffb385a7e039654cef1ab9ef32c6fafe283c0c0467bba1d9029738ce4a14a848", size = 196711, upload-time = "2026-06-02T14:34:07.794Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/1e/5e/d4e9f1a599fb8e573b7b87160658329fbf28d19eac2718f51fc3def3aa5a/idna-3.18-py3-none-any.whl", hash = "sha256:7f952cbe720b688055e3f87de14f5c3e5fdaa8bc3928985c4077ca689de849a2", size = 65455, upload-time = "2026-06-02T14:34:06.319Z" },
]

[[package]]
name = "iniconfig"
version = "2.3.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/72/34/14ca021ce8e5dfedc35312d08ba8bf51fdd999c576889fc2c24cb97f4f10/iniconfig-2.3.0.tar.gz", hash = "sha256:c76315c77db068650d49c5b56314774a7804df16fee4402c1f19d6d15d8c4730", size = 20503, upload-time = "2025-10-18T21:55:43.219Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/cb/b1/3846dd7f199d53cb17f49cba7e651e9ce294d8497c8c150530ed11865bb8/iniconfig-2.3.0-py3-none-any.whl", hash = "sha256:f631c04d2c48c52b84d0d0549c99ff3859c98df65b3101406327ecc7d53fbf12", size = 7484, upload-time = "2025-10-18T21:55:41.639Z" },
]

[[package]]
name = "jinja2"
version = "3.1.6"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "markupsafe" },
]
sdist = { url = "https://files.pythonhosted.org/packages/df/bf/f7da0350254c0ed7c72f3e33cef02e048281fec7ecec5f032d4aac52226b/jinja2-3.1.6.tar.gz", hash = "sha256:0137fb05990d35f1275a587e9aee6d56da821fc83491a0fb838183be43f66d6d", size = 245115, upload-time = "2025-03-05T20:05:02.478Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/62/a1/3d680cbfd5f4b8f15abc1d571870c5fc3e594bb582bc3b64ea099db13e56/jinja2-3.1.6-py3-none-any.whl", hash = "sha256:85ece4451f492d0c13c5dd7c13a64681a86afae63a5f347908daf103ce6d2f67", size = 134899, upload-time = "2025-03-05T20:05:00.369Z" },
]

[[package]]
name = "jiter"
version = "0.16.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/1d/1f/10936e16d8860c70698a1aa939a46aa0224813b782bce4e000e637da0b2d/jiter-0.16.0.tar.gz", hash = "sha256:7b24c3492c5f4f84a37946ad9cf504910cf6a782d6a4e0689b6673c5894b4a1c", size = 176431, upload-time = "2026-06-29T13:05:13.657Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/83/2b/52ace16ed031354f0539749a49e4bf33797d82bea5137910835fa4b09793/jiter-0.16.0-cp312-cp312-macosx_10_12_x86_64.whl", hash = "sha256:67c3bc1760f8c99d805dcab4e644027142a53b1d5d861f18780ebdbd5d40b72a", size = 306943, upload-time = "2026-06-29T13:03:14.035Z" },
    { url = "https://files.pythonhosted.org/packages/94/2e/34957c2c1b661c252ba9bcc60ae0bddc27e0f7202c6073326a13c5390eec/jiter-0.16.0-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:5af7780e4a26bd7d0d989592bf9ef12ebf806b74ab709223ecca37c749872ea9", size = 307779, upload-time = "2026-06-29T13:03:15.418Z" },
    { url = "https://files.pythonhosted.org/packages/88/6c/59bd309cab4460c54cf1079f3eb7fe7af6a4c895c5c957a53378693bad2b/jiter-0.16.0-cp312-cp312-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:d5bf78d0e05e45cfdd66558893938d59afe3d1b1a824a202039b20e607d25a72", size = 335826, upload-time = "2026-06-29T13:03:17.11Z" },
    { url = "https://files.pythonhosted.org/packages/3b/8c/f5ef7b65f0df47afa16596969defb281ebb86e96df346d62be6fd853d620/jiter-0.16.0-cp312-cp312-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:f4444a83f946605990c98f625cdd3d2725bfb818158760c5748c653170a20e0e", size = 362573, upload-time = "2026-06-29T13:03:18.781Z" },
    { url = "https://files.pythonhosted.org/packages/2b/0b/ace4354da061ee38844a0c27dc2c21eecd27aea119e8da324bea987522d0/jiter-0.16.0-cp312-cp312-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:3a23f0e4f957e1be65752d2dfac9a5a06b1917af8dc85deb639c3b9d02e31290", size = 457979, upload-time = "2026-06-29T13:03:20.293Z" },
    { url = "https://files.pythonhosted.org/packages/55/40/c0253d3772eb9dcd8e6606ee9b2d53ec8e5b814589c47f140aa585f21eaa/jiter-0.16.0-cp312-cp312-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:c22a488f7b9218e245a0025a9ba6b100e2e54700831cf4cf16833a27fba3ad01", size = 372302, upload-time = "2026-06-29T13:03:21.739Z" },
    { url = "https://files.pythonhosted.org/packages/a8/d2/4839422241aa12860ce597b20068727094ba0bc480723c74924ca5bad483/jiter-0.16.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:46add52f4ad47a08bfb1219f3e673da97219148 9a33016edefdb5ea55bfa8c48", size = 343805, upload-time = "2026-06-29T13:03:23.384Z" },
    { url = "https://files.pythonhosted.org/packages/e2/59/e196888a05befdda7dbe299b722d56f2f6eec65402bc34c0a3306d595feb/jiter-0.16.0-cp312-cp312-manylinux_2_31_riscv64.whl", hash = "sha256:9c8a956fd72c2cf1e730d01ea080341f13aa0a97a4a33b51abebe725b7ae9ca9", size = 351107, upload-time = "2026-06-29T13:03:24.815Z" },
    { url = "https://files.pythonhosted.org/packages/ec/74/4cd9e0fca65232136400354b630fbfcd2de634e22ccbb96567725981b548/jiter-0.16.0-cp312-cp312-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:561926e0573ffe4a32498420a76d64b16c513e1ab413b9d28158a8764ac701e5", size = 388441, upload-time = "2026-06-29T13:03:26.266Z" },
    { url = "https://files.pythonhosted.org/packages/d9/8c/554691e48bc711299c0a293dd8a6179e24b2d66a54dc295421fcf64569c0/jiter-0.16.0-cp312-cp312-musllinux_1_1_aarch64.whl", hash = "sha256:44d019fa8cdaf89bf29c71b39e3712143fdd0ac76725c6ef954f9957a5ea8730", size = 516354, upload-time = "2026-06-29T13:03:28.02Z" },
    { url = "https://files.pythonhosted.org/packages/a4/cb/01e9d69dc2cc6759d4f91e230b34489c4fdb2518992650633f9e20bece89/jiter-0.16.0-cp312-cp312-musllinux_1_1_x86_64.whl", hash = "sha256:0df91907609837f33341b8e6fe73b95991fdaa57caf1a0fbd343dffe826f386f", size = 547880, upload-time = "2026-06-29T13:03:29.534Z" },
    { url = "https://files.pythonhosted.org/packages/79/70/2953195f1c6ad00f49fa67e13df7e60acb3dd4f387101bc15abccddd905e/jiter-0.16.0-cp312-cp312-win32.whl", hash = "sha256:51d7b836acb0108d7c77df1742332cac2a1fa04a74d6dacec46e7091f0e91274", size = 203473, upload-time = "2026-06-29T13:03:31.025Z" },
    { url = "https://files.pythonhosted.org/packages/2d/05/2909a8b10699a4d560f8c502b6b2c5f3991b682b1922c1eedda242b225bd/jiter-0.16.0-cp312-cp312-win_amd64.whl", hash = "sha256:1878349266f8ee36ecb1375cc5ba2f115f35fd9f0a1a4119e725e379126647f7", size = 196905, upload-time = "2026-06-29T13:03:32.472Z" },
    { url = "https://files.pythonhosted.org/packages/e9/a9/6b82bb1c8d7790d602489b967b982a909e5d092875a6c2ade96444c8dfc5/jiter-0.16.0-cp312-cp312-win_arm64.whl", hash = "sha256:2ed5738ae4af18271a51a528b8811b0cbfa4a1858de9d83359e4169855d6a331", size = 190618, upload-time = "2026-06-29T13:03:34.672Z" },
    { url = "https://files.pythonhosted.org/packages/91/c0/555fc60473d30d66894ba825e63615e3be7524fac23858356afa7a38906c/jiter-0.16.0-cp313-cp313-macosx_10_12_x86_64.whl", hash = "sha256:41977aa5654023948c2dae2a81cbf9c43343954bef1cd59a154dd15a4d84c195", size = 306203, upload-time = "2026-06-29T13:03:36.243Z" },
    { url = "https://files.pythonhosted.org/packages/d0/2b/c3eaf16f5d7c9bad66ea32f40a95bd169b29a91217fcc7f081375157e99c/jiter-0.16.0-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:d28bb3c26762358dadf3e5bf0bccd29ae987d65e6988d2e6f49829c76b003c09", size = 306489, upload-time = "2026-06-29T13:03:37.846Z" },
    { url = "https://files.pythonhosted.org/packages/96/3f/02fdfc6705cad96127d883af5c34e4867f554f29ec7705ec1a46156400a9/jiter-0.16.0-cp313-cp313-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:0542a7189c26920778658fc8fcf2af8bae05bae9924577f71804acef37996536", size = 335453, upload-time = "2026-06-29T13:03:39.221Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/b2/a6/e4bda5920d4b0d7c5dfb7174ce4a6b2e4d3e11c9162c452ef0eab4cdbdbd/jiter-0.16.0-cp313-cp313-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:8fb8de1e23a0cb2a7f53c335049c7b72b6db41aa6227cdcc0972a1de5cb39450", size = 361625, upload-time = "2026-06-29T13:03:40.597Z" },
    { url = "https://files.pythonhosted.org/packages/b7/97/4e6b59b2c6e55cbb3e183595f81ad65dcfb21c915fee5e19e335df21bc55/jiter-0.16.0-cp313-cp313-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:b72d0b2990ca754a9102779ac98d8597b7cb3167895856221 4a007f909eab78e", size = 456958, upload-time = "2026-06-29T13:03:42.074Z" },
    { url = "https://files.pythonhosted.org/packages/15/e0/97e9557686d2f94f4b93786eccb7eed28e9228ad132ea8237f44727314a7/jiter-0.16.0-cp313-cp313-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:d5f91b1c27fc22a57993d5a5cb8a627cb8ed4b10502716fac1ffbfe1d19d84e8", size = 372017, upload-time = "2026-06-29T13:03:43.658Z" },
    { url = "https://files.pythonhosted.org/packages/0f/94/db768b6938e0df35c86beeba3dfbbb025c9ee5c19e1aa271f2396e50864d/jiter-0.16.0-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:c682bea068a90b764577bdb78a60a4c1d1606daf9cd4c893832a37c7cc9d9026", size = 343320, upload-time = "2026-06-29T13:03:45.226Z" },
    { url = "https://files.pythonhosted.org/packages/c1/d6/5a59d938244a30735fe62d9433fd325f9021ea29d89780ea4596ea93bc89/jiter-0.16.0-cp313-cp313-manylinux_2_31_riscv64.whl", hash = "sha256:8d031aabecc4f1b6276adfb42e3aabb77c89d468bf616600e8d3a11328929053", size = 350520, upload-time = "2026-06-29T13:03:46.671Z" },
    { url = "https://files.pythonhosted.org/packages/67/f8/c4a857f49c9af125f6bbcac7e3eee7f7978ed89682833062e2dbf62576b1/jiter-0.16.0-cp313-cp313-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:eab2cd170150e70153de16896a1774e3a1dca80154c56b54d7a812c479a7165e", size = 387550, upload-time = "2026-06-29T13:03:48.361Z" },
    { url = "https://files.pythonhosted.org/packages/8b/d6/5fbc2f7d6b67b754caa61a993a2e626e815dec47ffc2f9e35f01adfebec7/jiter-0.16.0-cp313-cp313-musllinux_1_1_aarch64.whl", hash = "sha256:6edb63a46e65a82c26800a868e49b2cac30dd5a4218b88d74bc2c848c8ad60bb", size = 515424, upload-time = "2026-06-29T13:03:49.881Z" },
    { url = "https://files.pythonhosted.org/packages/ed/54/284f0164b64a5fed915fea6ba7e9ba9b3d8d37c67d59cf2e3bb99d45cdfe/jiter-0.16.0-cp313-cp313-musllinux_1_1_x86_64.whl", hash = "sha256:659039cc50b5addcc35fcc87ae2c1833b7c0a8e5326ef631a75e4478447bcf84", size = 546981, upload-time = "2026-06-29T13:03:51.363Z" },
    { url = "https://files.pythonhosted.org/packages/13/c5/2a467585a576594384e1d2c43e1224deaafc085f24e243529cf98beef8e1/jiter-0.16.0-cp313-cp313-win32.whl", hash = "sha256:c9c53be232c2e206ef9cdbad81a48bfa74c3d3f08bcf8124630a8a748aad993e", size = 202853, upload-time = "2026-06-29T13:03:53.015Z" },
    { url = "https://files.pythonhosted.org/packages/88/6a/de61d04b9eec69c71719968d2f716532a3bc121170c44a39e14979c6be81/jiter-0.16.0-cp313-cp313-win_amd64.whl", hash = "sha256:baad945ed47f163ad833314f8e3288c396118934f94e7bbb9e243ce4b341a4fd", size = 196160, upload-time = "2026-06-29T13:03:54.447Z" },
    { url = "https://files.pythonhosted.org/packages/19/4b/b390ed59bafb3f31d008d1218578f10327714484b334439947f7e5b11e7f/jiter-0.16.0-cp313-cp313-win_arm64.whl", hash = "sha256:3c1fd2dbe1b0af19e987f03fe66c5f5bd105a2229c1aff4ab14890b24f41d21a", size = 189862, upload-time = "2026-06-29T13:03:55.754Z" },
    { url = "https://files.pythonhosted.org/packages/a7/89/bc4f1b57d5da938fd344a466396541e586d161320d70bffd929aaafcd8f4/jiter-0.16.0-cp314-cp314-macosx_10_12_x86_64.whl", hash = "sha256:b2c61484666ad42726029af0c00ef4541f0f3b5cdc550221f56c2343208018ee", size = 308239, upload-time = "2026-06-29T13:03:57.205Z" },
    { url = "https://files.pythonhosted.org/packages/65/7a/c415453e5213001bf3b411ff65dec3d303b0e76a4a2cfea9768cd4960994/jiter-0.16.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:63efadc657488f45db1c676d81e704cac2abf3fdb892def1faea61db053127e2", size = 308928, upload-time = "2026-06-29T13:03:58.643Z" },
    { url = "https://files.pythonhosted.org/packages/11/fc/1f4fb7ebf9a724c7741994f4aae18fba1e2f3133df14521a79194952c34a/jiter-0.16.0-cp314-cp314-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:cf0d73f50e7b6935677854f6e8e31d499ca7064dd24734f703e060f5b237d883", size = 336998, upload-time = "2026-06-29T13:04:00.071Z" },
    { url = "https://files.pythonhosted.org/packages/a0/8d/72cadaac05ccfa7cc3a0a2232862e6c72443ca40cf300ba8b57f9f18b69b/jiter-0.16.0-cp314-cp314-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:bf3ea07d9bc8e7d03a9fbc051295462e6dbc295b894fd72457c3136e3e43d898", size = 362112, upload-time = "2026-06-29T13:04:01.52Z" },
    { url = "https://files.pythonhosted.org/packages/58/4a/c4b0d5f651fda90a24ffce9f8d56cde462a2e09d31ae3de3c68cef34c04e/jiter-0.16.0-cp314-cp314-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:26798522707abb47d767db536e4148ceac1b14446bf028ee85e579a2e043cfe5", size = 459807, upload-time = "2026-06-29T13:04:03.214Z" },
    { url = "https://files.pythonhosted.org/packages/80/58/ef77879ea9aa56b50824edc5a445e226422c7a8d211f3fd2a56bcb9493cf/jiter-0.16.0-cp314-cp314-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:bc837c1b9631be10abfe0191537fe8009838204cec7e44827401ace390ddb567", size = 373181, upload-time = "2026-06-29T13:04:04.629Z" },
    { url = "https://files.pythonhosted.org/packages/49/2e/ffbc3f254e4d8a66da3062c624a7df4b7c2b2cf9e1fe43cf394b3e104041/jiter-0.16.0-cp314-cp314-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:49060fd70737fad59d33ba9dcc0d83247dc9e77187de26053a19c16c9f32bd69", size = 344927, upload-time = "2026-06-29T13:04:06.067Z" },
    { url = "https://files.pythonhosted.org/packages/9a/f6/0be5dc6d64a89f80aa8fec984f94dedb2973e251edcae55841d60786d578/jiter-0.16.0-cp314-cp314-manylinux_2_31_riscv64.whl", hash = "sha256:adbb8edeadd431bc4477879d5d371ece7cb1334486584e0f252656dd7ffada29", size = 352754, upload-time = "2026-06-29T13:04:07.477Z" },
    { url = "https://files.pythonhosted.org/packages/da/6e/7d31243b3b91cd261dd19e9d3557fc3251a80883d3d8049c86174e7ab7af/jiter-0.16.0-cp314-cp314-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:31aaee5b80f672c1dc21272bcfb9cbdcfc1ea04ff50f00ed5af500b80c44fa93", size = 390553, upload-time = "2026-06-29T13:04:08.92Z" },
    { url = "https://files.pythonhosted.org/packages/25/33/51ae371fde3c88897520f62b4d5f8b27ad7103e2bb10812ff52195609853/jiter-0.16.0-cp314-cp314-musllinux_1_1_aarch64.whl", hash = "sha256:6722bcef4ffc86c835574b1b2fac6b33b9fb4a889c781e67950e891591f3c55a", size = 516900, upload-time = "2026-06-29T13:04:10.407Z" },
    { url = "https://files.pythonhosted.org/packages/a0/45/6449b3d123ea439ba79507c657288f461d55049e7bcbdc2cf8eb8210f491/jiter-0.16.0-cp314-cp314-musllinux_1_1_x86_64.whl", hash = "sha256:5ab4f50ff971b611d656554ea10b75f80097392c827bc32923c6eeb6386c8b00", size = 548754, upload-time = "2026-06-29T13:04:12.046Z" },
    { url = "https://files.pythonhosted.org/packages/9b/e7/fd2fb11ae3e2649333da3aa170d04d7b3000bbdc3b270f6513382fdf4e04/jiter-0.16.0-cp314-cp314-pyemscripten_2026_0_wasm32.whl", hash = "sha256:710cc51d4ebdcd3c1f70b232c1db1ea1344a075770422bbd4bede5708335acbe", size = 122381, upload-time = "2026-06-29T13:04:13.413Z" },
    { url = "https://files.pythonhosted.org/packages/26/80/f0b147a62c315a164ed2168908286ca302310824c218d3aae52b06c0c9a9/jiter-0.16.0-cp314-cp314-win32.whl", hash = "sha256:57b37fc887a32d44798e4d8ebfa7c9683ff3da1d5bf38f08d1bb3573ccb39106", size = 204578, upload-time = "2026-06-29T13:04:14.813Z" },
    { url = "https://files.pythonhosted.org/packages/5e/e6/4758a14304b4523a6f5adb2419340086aa3593bd4327c2b25b5948a90548/jiter-0.16.0-cp314-cp314-win_amd64.whl", hash = "sha256:cbd18dd5e2df96b580487b5745adf57ef64ad89ba2d9662fc3c19386acce7db8", size = 198154, upload-time = "2026-06-29T13:04:16.272Z" },
    { url = "https://files.pythonhosted.org/packages/26/be/41fa54a2e7ea41d6c99f1dc5b1f0fd4cb474680304b5d268dd518e81da3a/jiter-0.16.0-cp314-cp314-win_arm64.whl", hash = "sha256:a32d2027a9fa67f109ff245a3252ece3ccc32cc56703e1deab6cc846a59e0585", size = 191458, upload-time = "2026-06-29T13:04:17.707Z" },
    { url = "https://files.pythonhosted.org/packages/81/6b/59127338b86d9fe4d99418f5a15118bea778103ee0fe9d9dd7e0af174e95/jiter-0.16.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:2577196f4474ef3fc4779a088a23b0897bbf86f9ea3679c372d45b8383b43207", size = 316739, upload-time = "2026-06-29T13:04:19.663Z" },
    { url = "https://files.pythonhosted.org/packages/2d/95/49461034d5388196d3dabf98748935f017b7785d8f3f5349f834bcc4ed0d/jiter-0.16.0-cp314-cp314t-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:616e89e008a93c01104161c75b4988e58716b01d62307ebfe161e52a56d2a818", size = 340911, upload-time = "2026-06-29T13:04:21.257Z" },
    { url = "https://files.pythonhosted.org/packages/cd/97/a4369f2fb82cb3dda13b98622f31249b2e014b223fe64ee534413ad72294/jiter-0.16.0-cp314-cp314t-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:0e2e9efbe042210df657bade597f66d6d75723e3d8f45a12ea6d8167ff8bbce3", size = 361747, upload-time = "2026-06-29T13:04:22.677Z" },
    { url = "https://files.pythonhosted.org/packages/28/51/49b6ed456261646e1906016a6760367a28aacd3c24805e4e5fe64116c1db/jiter-0.16.0-cp314-cp314t-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:3f4d9e473a5ce7d27fef8b848df4dc16e283893d3f53b4a585e72c9595f3c284", size = 460225, upload-time = "2026-06-29T13:04:24.441Z" },
    { url = "https://files.pythonhosted.org/packages/33/b5/5689aff4f66c5b60be63106e591dbfcba2190df97d2c9c7cf052361ddb98/jiter-0.16.0-cp314-cp314t-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:8d30a4a1c87713060c8d1cc59a7b6c8fb6b8ef0a6900368014c76c87922a2929", size = 373169, upload-time = "2026-06-29T13:04:25.884Z" },
    { url = "https://files.pythonhosted.org/packages/a2/96/3ae1b85ee0d6d6cab254fb7f8da018272b932bbf2d69b07e98aa2a96c746/jiter-0.16.0-cp314-cp314t-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:bae96332410f866e5900d809298b1ed82735932986c672495f9701daacd80620", size = 350332, upload-time = "2026-06-29T13:04:27.302Z" },
    { url = "https://files.pythonhosted.org/packages/15/32/c99d7bafd78986556c95bf60ce84c6cc98786eac56066c12d7f828bb6747/jiter-0.16.0-cp314-cp314t-manylinux_2_31_riscv64.whl", hash = "sha256:da3d7ec75dc83bb18bca888b5edfae0656a26849056c59e05a7728badd17e7af", size = 353377, upload-time = "2026-06-29T13:04:28.731Z" },
    { url = "https://files.pythonhosted.org/packages/0e/4b/f99a8e571287c3dec766bcc18528bbe8e8fb5365522ab5e6d64c93e87066/jiter-0.16.0-cp314-cp314t-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:ee6162b77d49a9939229df666dfa8af3e656b6701b54c4c84966d740e189264e", size = 387746, upload-time = "2026-06-29T13:04:30.319Z" },
    { url = "https://files.pythonhosted.org/packages/75/69/c78a5b3f71040e34eb5917df26fb7ae9a2174cad1ccbf277512507c53a6e/jiter-0.16.0-cp314-cp314t-musllinux_1_1_aarch64.whl", hash = "sha256:63ffdbdae7d4499f4cda14eadc12ddcabef0fc0c081191bdc2247489cb698077", size = 517292, upload-time = "2026-06-29T13:04:31.709Z" },
    { url = "https://files.pythonhosted.org/packages/c2/f7/095b38eda4c70d03651c403f29a5590f16d12ddc5d544aac9f9cddf72277/jiter-0.16.0-cp314-cp314t-musllinux_1_1_x86_64.whl", hash = "sha256:a111256a7193bea0759267b10385e5870949c239ed7b6ddbaaf57573edb38734", size = 549259, upload-time = "2026-06-29T13:0
```

```
4:33.721Z" },
    { url = "https://files.pythonhosted.org/packages/2e/c5/6a0207d90e5f656d95af98ebd0934f382d37674416f215aeda2ff8063e51/jiter-0.16.0-cp314-cp314t-win32.whl", hash = "sha256:de5ba8763e56b793561f43bed197c9ea55776daa5e9a6b91eed68a909bc9cdbf", size = 206523, upload-time = "2026-06-29T13:04:35.068Z" },
    { url = "https://files.pythonhosted.org/packages/a5/31/c757d5f30a8980fd945ce7b98be10be9e4ff59c7c42f5fd86804c2e87db8/jiter-0.16.0-cp314-cp314t-win_amd64.whl", hash = "sha256:b8a3f9a6008048fe9def7bf465180564a6e458047d2ce499149cfbe73c3ae9db", size = 200366, upload-time = "2026-06-29T13:04:36.61Z" },
    { url = "https://files.pythonhosted.org/packages/7c/a2/d88de6d313d734a544a7901353ad5db67cb38dcfcd91713b7979dafc345d/jiter-0.16.0-cp314-cp314t-win_arm64.whl", hash = "sha256:0fa25b09b13075c46f5bc174f2690525a925a4fc2f7c82969a2bbabff22386ce", size = 190516, upload-time = "2026-06-29T13:04:38.004Z" },
    { url = "https://files.pythonhosted.org/packages/98/ab/664fd8c4be028b2bedd3d2ff08769c4ede23d0dbc87a77c62384a0515b5d/jiter-0.16.0-graalpy312-graalpy250_312_native-macosx_10_12_x86_64.whl", hash = "sha256:f17d61a28b4b3e0e3e2ba98490c70501403b4d196f78732439160e7fd3678127", size = 303106, upload-time = "2026-06-29T13:05:07.118Z" },
    { url = "https://files.pythonhosted.org/packages/1a/07/421f1d5b65493a76e16027b848aba6a7d28073ae75944fa4289cc914d39f/jiter-0.16.0-graalpy312-graalpy250_312_native-macosx_11_0_arm64.whl", hash = "sha256:96e38eea538c8ddf853a35727c7be0741c76c13f04148ac5c116222f50ece3b3", size = 304658, upload-time = "2026-06-29T13:05:08.708Z" },
    { url = "https://files.pythonhosted.org/packages/0a/db/bba1155f01a01c3c37a89425d571da751bbedf5c54247b831a04cb971798/jiter-0.16.0-graalpy312-graalpy250_312_native-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:d284fb8d94d5855d60c44fefcab4bf966f1da6fada73992b01f6f0c9bc0c6702", size = 339719, upload-time = "2026-06-29T13:05:10.41Z" },
    { url = "https://files.pythonhosted.org/packages/78/f7/18a1afcd64f35314b68c1f23afcd9994d0bc13e65cc77517afff4e83986d/jiter-0.16.0-graalpy312-graalpy250_312_native-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:64d613743df53199b1aa256a7d328340da6d7078aac7705a7db9d7a791e9cfd2", size = 343885, upload-time = "2026-06-29T13:05:12.087Z" },
]

[[package]]
name = "joblib"
version = "1.5.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/41/f2/d34e8b3a08a9cc79a50b2208a93dce981fe615b64d5a4d4abee421d898df/joblib-1.5.3.tar.gz", hash = "sha256:8561a3269e6801106863fd0d6d84bb737be9e7631e33aaed3fb9ce5953688da3", size = 331603, upload-time = "2025-12-15T08:41:46.427Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/7b/91/984aca2ec129e2757d1e4e3c81c3fcda9d0f85b74670a094cc443d9ee949/joblib-1.5.3-py3-none-any.whl", hash = "sha256:5fc3c5039fc5ca8c0276333a188bbd59d6b7ab37fe6632daa76bc7f9ec18e713", size = 309071, upload-time = "2025-12-15T08:41:44.973Z" },
]

[[package]]
name = "jsonpatch"
version = "1.33"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "jsonpointer" },
]
sdist = { url = "https://files.pythonhosted.org/packages/42/78/18813351fe5d63acad16aec57f94ec2b70a09e53ca98145589e185423873/jsonpatch-1.33.tar.gz", hash = "sha256:9fcd4009c41e6d12348b4a0ff2563ba56a2923a7dfee731d004e212e1ee5030c", size = 21699, upload-time = "2023-06-26T12:07:29.144Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/73/07/02e16ed01e04a374e644b575638ec7987ae846d25ad97bcc9945a3ee4b0e/jsonpatch-1.33-py2.py3-none-any.whl", hash = "sha256:0ae28c0cd062bbd8b8ecc26d7d164fbbea9652a1a3693f3b956c1eae5145dade", size = 12898, upload-time = "2023-06-16T21:01:28.466Z" },
]

[[package]]
name = "jsonpointer"
version = "3.1.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/18/c7/af399a2e7a67fd18d63c40c5e62d3af4e67b836a2107468b6a5ea24c4304/jsonpointer-3.1.1.tar.gz", hash = "sha256:0b801c7db33a904024f6004d526dcc53bbb8a4a0f4e32bfd10beadf60adf1900", size = 9068, upload-time = "2026-03-23T22:32:32.458Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/9e/6a/a83720e953b1682d2d109d3c2dbb0bc9bf28cc1cbc205be4ef4be5da709d/jsonpointer-3.1.1-py3-none-any.whl", hash = "sha256:8ff8b95779d071ba472cf5bc913028df06031797532f08a7d5b602d8b2a488ca", size = 7659, upload-time = "2026-03-23T22:32:31.568Z" },
]

[[package]]
name = "jsonschema"
version = "4.26.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "attrs" },
    { name = "jsonschema-specifications" },
    { name = "referencing" },
    { name = "rpds-py" },
]
sdist = { url = "https://files.pythonhosted.org/packages/b3/fc/e067678238fa451312d4c62bf6e6cf5ec56375422aee02f9cb5f909b3047/jsonschema-4.26.0.tar.gz", hash = "sha256:0c26707e2efad8aa1bfc5b7ce170f3fccc2e4918ff85989ba9ffa9facb2be326", size = 366583, upload-time = "2026-01-07T13:41:07.246Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/69/90/f63fb5873511e014207a475e2bb4e8b2e570d655b00ac19a9a0ca0a385ee/jsonschema-4.26.0-py3-none-any.whl", hash = "sha256:d489f15263b8d200f8387e64b4c3a75f06629559fb73deb8fdfb525f2dab50ce", size = 90630, upload-time = "2026-01-07T13:41:05.306Z" },
]

[[package]]
name = "jsonschema-specifications"
version = "2025.9.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "referencing" },
]
sdist = { url = "https://files.pythonhosted.org/packages/19/74/a633ee74eb36c44aa6d1095e7cc5569bebf04342ee146178e2d36600708b/jsonschema_specifications-2025.9.1.tar.gz", hash = "sha256:b540987f239e745613c7a9176f3edb72b832a4ac465cf02712288397832b5e8d", size = 32855, upload-time = "2025-09-08T01:34:59.186Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/41/45/1a4ed80516f02155c51f51e8cedb3c1902296743db0bbc66608a0db2814f/jsonschema_specifications-2025.9.1-py3-none-any.whl", hash = "sha256:98802fee3a11ee76ecaca44429fda8a41bff98b00a0f2838151b113f210cc6fe", size = 18437, upload-time = "2025-09-08T01:34:57.871Z" },
]

[[package]]
name = "langchain"
version = "1.3.14"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "langchain-core" },
    { name = "langgraph" },
    { name = "pydantic" },
]
sdist = { url = "https://files.pythonhosted.org/packages/29/68/a6dbad9c22df4087a0f9e79ddd46226c442b30128bfeee538d5889492a73/langchain-1.3.14.tar.gz", hash = "sha256:1b6696c72ba3bbbce54d745e0180742c9f6ece8bbc59ed5a46c3e20b9a435929", size = 645181, upload-time = "2026-07-16T13:28:18.29Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a9/ec/0f942e78a621f8e3162ff1ed24284f469aaf51fb4607ee5831c626f2b2bc/langchain-1.3.14-py3-none-any.
```

```
whl", hash = "sha256:4d10dbe91005952cddd56d0dc77aa108964da6bae90ab20063653957e901f782", size = 139560, upload-time = "2026-07-16T13:28:16.498Z" },
]

[[package]]
name = "langchain-anthropic"
version = "1.5.4"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anthropic" },
    { name = "langchain-core" },
    { name = "pydantic" },
]
sdist = { url = "https://files.pythonhosted.org/packages/d8/9d/d56f68ec9a2c8aba3b7639e5d7a502d82cbf71c31e178a06c0955d789792/langchain_anthropic-1.5.4.
tar.gz", hash = "sha256:113cb9bdac3169f2da65ea395dedb290e30dc6f899d01c205ba88aa88a2a02c2", size = 718860, upload-time = "2026-08-05T19:03:16.848Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/23/d8/47fb549e91f55a54ce829b5034a21360d9383342a3bf4986188e254e89e4/langchain_anthropic-1.5.4-py3-
none-any.whl", hash = "sha256:730a9cb1ad384c9f1642840469c1ebbf20237066b1635f5d4fa9876e365fceaf", size = 56025, upload-time = "2026-08-05T19:03:15.579Z
" },
]

[[package]]
name = "langchain-core"
version = "1.5.3"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "jsonpatch" },
    { name = "langchain-protocol" },
    { name = "langsmith" },
    { name = "packaging" },
    { name = "pydantic" },
    { name = "pyyaml" },
    { name = "tenacity" },
    { name = "typing-extensions" },
    { name = "uuid-utils" },
]
sdist = { url = "https://files.pythonhosted.org/packages/65/3e/63af6b9d76d9be907c7c524d6ec18a2efed7e0e2d123fea0230d78dbd73f/langchain_core-1.5.3.tar.g
z", hash = "sha256:a56457ac444fef41e9404443c187f0ecea708d36e816ea4ba9573c027f7d1a2d", size = 972461, upload-time = "2026-07-30T14:55:55.833Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/36/e6/c7c39efe0bc7e1b7c3d8f54f85846e04c901913c3d3e99068b218558c6f1/langchain_core-1.5.3-py3-none-
any.whl", hash = "sha256:48b56fa580277209594dd7baf837f5b9a2a3651613f34ff9fb1728b429df015f", size = 561687, upload-time = "2026-07-30T14:55:54.419Z" },
]

[[package]]
name = "langchain-protocol"
version = "0.0.18"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/d2/59/b5959aea96faa9146e2e49a7a22882b3528c62efafe9a6a95beab30c2305/langchain_protocol-0.0.18.
tar.gz", hash = "sha256:ec3e11782f1ed0c9db38e5a9ed01b0e7a0d3fba406faa8aef6594b73c56a63e6", size = 6150, upload-time = "2026-06-18T17:08:26.959Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/99/2e/d82db9eec13ad0f72e7aaad5c4bc730ab111934fdc83c85523206eb9b0a0/langchain_protocol-0.0.18-py3-
none-any.whl", hash = "sha256:70b53a86fbf9cedc863555effe44da192ab02d556ddbf2cf95b8873adcf41b5a", size = 7221, upload-time = "2026-06-18T17:08:25.996Z"
 },
]

[[package]]
name = "langfuse"
version = "4.14.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "backoff" },
    { name = "httpx" },
    { name = "opentelemetry-api" },
    { name = "opentelemetry-exporter-otlp-proto-http" },
    { name = "opentelemetry-sdk" },
    { name = "packaging" },
    { name = "pydantic" },
    { name = "wrapt" },
]
sdist = { url = "https://files.pythonhosted.org/packages/36/7c/4cdf76a100f169a0855d4a7e69598a3bbe19bd2f048c5de3c298af1aa2c0/langfuse-4.14.2.tar.gz", h
ash = "sha256:76eb3f2c2106ddd63271154d1a33f8b3d4036f449a9f9b48403c36bd5e7c1bba", size = 389507, upload-time = "2026-07-30T11:54:48.46Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/8b/b5/1499bcedb6872e8acc4e7043c522c9f3a4fe2eafd78a6f37e73308d18753/langfuse-4.14.2-py3-none-any.w
hl", hash = "sha256:27879cbaea570e94323c6ced60a0997feedf5294f18b51436009947c42f9f92e", size = 688132, upload-time = "2026-07-30T11:54:49.553Z" },
]

[[package]]
name = "langgraph"
version = "1.2.10"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "langchain-core" },
    { name = "langgraph-checkpoint" },
    { name = "langgraph-prebuilt" },
    { name = "langgraph-sdk" },
    { name = "pydantic" },
    { name = "xxhash" },
]
sdist = { url = "https://files.pythonhosted.org/packages/70/1d/a32f3caf4b3d60651656c0d64976b48d168653e81c71bb7512e9a31541aa/langgraph-1.2.10.tar.gz",
hash = "sha256:05a183a746ed570a06c7c1b879920163509a75df9e44e92dd2238218d677fd37", size = 723404, upload-time = "2026-07-28T18:33:51.441Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e2/4d/3fc3e2535ee2c731130d71371848ebc6d4a9d2e8ae6060b11987ba134951/langgraph-1.2.10-py3-none-any.
whl", hash = "sha256:52c48bd42fa31a1de0e1c0f0ebfe342e11ca2957b8b3563f83dbd60d8e30f921", size = 247753, upload-time = "2026-07-28T18:33:50.028Z" },
]

[[package]]
name = "langgraph-checkpoint"
version = "4.1.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "langchain-core" },
    { name = "ormsgpack" },
]
```

```
sdist = { url = "https://files.pythonhosted.org/packages/83/47/886af6f886f0bff2273164a45f008694e48a96ff3cd25ff0228f2aa9480e/langgraph_checkpoint-4.1.1
.tar.gz", hash = "sha256:6c2bdb530c91f91d7d9c1bd100925d0fc4f498d418c17f3587d1526279482a25", size = 184020, upload-time = "2026-05-22T16:57:38.503Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/bd/b4/71425e3e38be92611300b9cc5e46a5bf98ab23f5ea8a75b73d02a2f1413c/langgraph_checkpoint-4.1.1-py3
-none-any.whl", hash = "sha256:25d29144b082827218e7bc3f1e9b0566a4bb007895cd6cc26f66a8428739f56e", size = 56212, upload-time = "2026-05-22T16:57:37.203
Z" },
]

[[package]]
name = "langgraph-prebuilt"
version = "1.1.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "langchain-core" },
    { name = "langgraph-checkpoint" },
]
sdist = { url = "https://files.pythonhosted.org/packages/29/66/ed9b93f56bc17ef22d551892f0ac2b225a97fe0fcf23a511b857f70d590b/langgraph_prebuilt-1.1.0.t
ar.gz", hash = "sha256:3c579cf6eed2d17f9c157c2d0fcaddcd8688524e7022d3b22b37a3bf4589d528", size = 178833, upload-time = "2026-05-12T03:37:49.332Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e9/43/3fe1a700b8490ed02679cdbbc8c915eb23a092faf496c9c1118abcd10be3/langgraph_prebuilt-1.1.0-py3-n
one-any.whl", hash = "sha256:51e311747d755b751d5c6b39b0c1446124d3a7643d2515017e6714b323508fc9", size = 41043, upload-time = "2026-05-12T03:37:48.007Z"
 },
]

[[package]]
name = "langgraph-sdk"
version = "0.4.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "httpx" },
    { name = "langchain-core" },
    { name = "langchain-protocol" },
    { name = "orjson" },
    { name = "websockets" },
]
sdist = { url = "https://files.pythonhosted.org/packages/b4/2b/bd8ac26d4e97f6df88ef05ce5b6a38945a3903e1025d926f4752aa88aa97/langgraph_sdk-0.4.2.tar.gz
", hash = "sha256:b88f0f5f6328ac0680d6790614a905b2bcfa257f2276dba4e38f0e86db0aa738", size = 348327, upload-time = "2026-06-01T17:51:19.856Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a0/05/aac507337cceae773c2cc9ab91eb6301963af7aeeb55b4217a00e15aff17/langgraph_sdk-0.4.2-py3-none-a
ny.whl", hash = "sha256:75fa5096c1177ce39c847096a8fe3745ffd480ddb412995f836e9f5f884c43dd", size = 160521, upload-time = "2026-06-01T17:51:18.849Z" },
]

[[package]]
name = "langsmith"
version = "0.10.16"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "distro" },
    { name = "httpx" },
    { name = "orjson", marker = "platform_python_implementation != 'PyPy'" },
    { name = "packaging" },
    { name = "pydantic" },
    { name = "requests" },
    { name = "requests-toolbelt" },
    { name = "sniffio" },
    { name = "typing-extensions" },
    { name = "uuid-utils" },
    { name = "websockets" },
    { name = "xxhash" },
    { name = "zstandard" },
]
sdist = { url = "https://files.pythonhosted.org/packages/b0/f1/ee288a685ae666121a5196575e56886e853b8fde9bbc5b796279cf85b2ba/langsmith-0.10.16.tar.gz",
 hash = "sha256:7dc5ff6477d50101b4944a985773bf9a3e2a9374e5d71ed2549ada929ca500b2", size = 4797058, upload-time = "2026-08-05T16:59:31.13Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/95/0b/cf0a51c19f659b40351dab76b4d3f0a3554807966a912111fda3156c96fe/langsmith-0.10.16-py3-none-any
.whl", hash = "sha256:f0b3882e0d2d69bda596c7b452a7857943aefb2c531c047677470b0a394cc490", size = 734191, upload-time = "2026-08-05T16:59:28.021Z" },
]

[[package]]
name = "librt"
version = "0.13.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/dc/2f/3908645ddddab7120b46295e541ead308109fa48dbec7d67d7a778870d60/librt-0.13.0.tar.gz", hash
 = "sha256:1d2a610c14ac0d0750ee0a3ab8548e83155258387891caaca04def4bf7289781", size = 211402, upload-time = "2026-07-08T12:26:29.834Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/f0/f4/b2933ddae222dac338476abb872641169a5cfed2c2bb5444a5b07b32b0c3/librt-0.13.0-cp312-cp312-macos
x_10_13_x86_64.whl", hash = "sha256:30536798f4504c0fad0885b1d371b0539abb081e4570c9d7c641cb51141b49f0", size = 150990, upload-time = "2026-07-08T12:25:
02.422Z" },
    { url = "https://files.pythonhosted.org/packages/90/ef/db98f744ca50e6efc9c95c70ee49b77aefac31f6a3fc7c83754a42d6a74f/librt-0.13.0-cp312-cp312-macos
x_11_0_arm64.whl", hash = "sha256:93d24ebb82aa4420b1409c389e7857bc35bd0b668007ac8172427d5c73cc8cc5", size = 155238, upload-time = "2026-07-08T12:25:03
.681Z" },
    { url = "https://files.pythonhosted.org/packages/03/e7/a197e7bc72baf2c61ce7fdc6906a5054dc05bd8da0819aa894e4857bf87e/librt-0.13.0-cp312-cp312-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:cb8a1adce42d8b75485a5d56a9623a50bcab995b6079f1dac59fc44034dd93d9",
 size = 503073, upload-time = "2026-07-08T12:25:05.049Z" },
    { url = "https://files.pythonhosted.org/packages/f8/e7/7887712e27da7c1ab80fcabb1de6eb24243964f6557cae530d4b70706dbd/librt-0.13.0-cp312-cp312-manyl
inux2014_i686.manylinux_2_17_i686.manylinux_2_28_i686.whl", hash = "sha256:0763ca2ab66058174f9dee426dc64f5e0a89c24a7df8d3fe3f1836c04e25de4b", size = 4
96528, upload-time = "2026-07-08T12:25:06.26Z" },
    { url = "https://files.pythonhosted.org/packages/94/f0/f2283385bb6b950b26a1410f4ce51ec27231e0b3a4b925c46366d218b198/librt-0.13.0-cp312-cp312-manyl
inux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:b222493da6e7b6199db9bd79502436cf5a27da3c1f7fa83c7e285444fc93fd03", si
ze = 531786, upload-time = "2026-07-08T12:25:07.658Z" },
    { url = "https://files.pythonhosted.org/packages/36/11/69ac3b54766ffba5fd7e5acebfb048d66dbe1f9f2d14516c2b3edc59cf87/librt-0.13.0-cp312-cp312-manyl
inux_2_34_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:fadc63331f4388c3dc90090448f682a7e9feafc11481391c1e94f2f907a3976e", size = 524393, upload
-time = "2026-07-08T12:25:09.121Z" },
    { url = "https://files.pythonhosted.org/packages/61/5f/d72f95fd444a926a3c14b4e24979474116988dd57a45be242077c45d3c22/librt-0.13.0-cp312-cp312-musll
inux_1_2_aarch64.whl", hash = "sha256:70d9c62a4cffd9f23396cd5ef93fc5d11b31596b9b7d6306074abe3d5fcf09bd", size = 543026, upload-time = "2026-07-08T12:2
5:10.459Z" },
    { url = "https://files.pythonhosted.org/packages/c4/08/dcd9993ad192737a004ba263d549f8ea605b326b952e7d6205c7d4170b76/librt-0.13.0-cp312-cp312-musll
inux_1_2_i686.whl", hash = "sha256:66c0e7e6b02a155576df2c77ec933a70b72da726e248c494abf690923e624348", size = 546829, upload-time = "2026-07-08T12:25:1
1.716Z" },
    { url = "https://files.pythonhosted.org/packages/96/d5/6d9bb2f54e4109a956b7128836529653eb9d740f784bc47ed10a02c1000e/librt-0.13.0-cp312-cp312-musll
inux_1_2_riscv64.whl", hash = "sha256:ac04bcd3328eb91d99dfedf6a60d9c1f15d3434e6f6daf922f0420f7d90b85c7", size = 535700, upload-time = "2026-07-08T12:2
5:13.144Z" },
    { url = "https://files.pythonhosted.org/packages/8c/f2/10946922503858a359492fa27f13e86228bde702116a740ac7b3cd185f24/librt-0.13.0-cp312-cp312-musll
inux_1_2_x86_64.whl", hash = "sha256:db327e7271e653c32040b85ae6188059c924b57d7e1e29f935523fa017cd4e82", size = 573566, upload-time = "2026-07-08T12:25
```

:14.336Z" },
    { url = "https://files.pythonhosted.org/packages/48/a8/94f00e3c99479a18088af3685ea016c42f3c7d5d1964d8dbb40c08d7f1aa/librt-0.13.0-cp312-cp312-win32.whl", hash = "sha256:860bd1d8ba48456ce08feaf8d343a8aaeb2fa086f2bcaa2a923fa3f7a3ff9aa3", size = 106099, upload-time = "2026-07-08T12:25:16.159Z" },
    { url = "https://files.pythonhosted.org/packages/c9/7b/2da9c74c1ed25a89cc4e1c8e007ea2eb4a0f1fafa3e70d757fe3242c5c5c/librt-0.13.0-cp312-cp312-win_amd64.whl", hash = "sha256:e54a315caf843c8d77e388cadc56ea9ded569935ee2d2347d7ea94992e5aa6fa", size = 126934, upload-time = "2026-07-08T12:25:17.275Z" },
    { url = "https://files.pythonhosted.org/packages/d0/65/aead61bbf3b5358593f9d4779d2a0e88eaf6ec191a6342dde36dd1df6371/librt-0.13.0-cp312-cp312-win_arm64.whl", hash = "sha256:c718e99a0992127af84385378460db624103b559ab260435abcfe77a4e4ed1c1", size = 112236, upload-time = "2026-07-08T12:25:18.425Z" },
    { url = "https://files.pythonhosted.org/packages/67/3b/18e7b63255297a2bdc9c25c8d6d4ca8eca9f63aceb1252c0f7427ac7099e/librt-0.13.0-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:a468951af16155824e88bdd8326ebe5bdb371f3ec0ac04642994b98201d914f3", size = 151027, upload-time = "2026-07-08T12:25:19.638Z" },
    { url = "https://files.pythonhosted.org/packages/4d/68/e2248452c00d1a03b45fee1752cdc8f790a476efd2402b75181da88a9e61/librt-0.13.0-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:ae01d8512cc17079e53425635327dbf3f7ff57a42c00dec348bf79791c56444c", size = 155152, upload-time = "2026-07-08T12:25:20.851Z" },
    { url = "https://files.pythonhosted.org/packages/0e/16/52b1c99bf19057a062aac39c900cbb81499f6f75d6c537c14463d247ba78/librt-0.13.0-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:32c26893cd085c1efe83219e78d866da23fb20a066101b8f68210004361d224c", size = 502499, upload-time = "2026-07-08T12:25:22.055Z" },
    { url = "https://files.pythonhosted.org/packages/9f/54/b811151805c795f55e0dedee6ec687b75f9982a8105d240ea3910737a77b/librt-0.13.0-cp313-cp313-manylinux2014_i686.manylinux_2_17_i686.manylinux_2_28_i686.whl", hash = "sha256:5929da1981a46bcf4b28b1b9499905f0ff58e2419da402a048234e9783acbc4b", size = 496108, upload-time = "2026-07-08T12:25:23.296Z" },
    { url = "https://files.pythonhosted.org/packages/8f/f8/094d6b2bd93f3fdaa54db54cc788c4a365333bddad65ab02e04da0b1d004/librt-0.13.0-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:94b85d664d777bab6c0d709416cb42938251fda9e221b79e3a2215d85df5f4f9", size = 531576, upload-time = "2026-07-08T12:25:24.648Z" },
    { url = "https://files.pythonhosted.org/packages/2e/40/541733d5755824f968f7ec39d78ffbd75d145964157ae5e69a09ec6d7326/librt-0.13.0-cp313-cp313-manylinux_2_34_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:531b2df3e9fe96b1fcf73a6d165921e4656be5f58d631d384ebce344298368db", size = 524390, upload-time = "2026-07-08T12:25:25.898Z" },
    { url = "https://files.pythonhosted.org/packages/c6/b5/255673cfdbf5ba663339d36cd863c897289ab4337577e19f9405ce059f36/librt-0.13.0-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:109b84a9edf69ad89dc1f66358659e14a031baca95e3e5b0060bd903ede8efd6", size = 543053, upload-time = "2026-07-08T12:25:27.436Z" },
    { url = "https://files.pythonhosted.org/packages/9e/11/ab5005e9c9850710f21e354201bf090646349d3fabf5f951eaf70235729e/librt-0.13.0-cp313-cp313-musllinux_1_2_i686.whl", hash = "sha256:1304368a3e7ffc3e9db986796cc5326fdb5943a3567ecc137cff318e4240c0e7", size = 546387, upload-time = "2026-07-08T12:25:28.65Z" },
    { url = "https://files.pythonhosted.org/packages/a2/04/a5d7ce1d1df1afd15ca283dcdf7530ac073e12d69ae8c40879dda96f7868/librt-0.13.0-cp313-cp313-musllinux_1_2_riscv64.whl", hash = "sha256:e4f9b472e7d308d94b62c801982065661158c6ed02790d6c7ddb4337cea0f9c1", size = 535970, upload-time = "2026-07-08T12:25:30.171Z" },
    { url = "https://files.pythonhosted.org/packages/5a/76/927e267a6daa290174ac281b23c9804c8829b042ade9c6f24a065f540958/librt-0.13.0-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:9f836c37478f167a81200d8c8b2c920a22224564bed2c23d7aeec760965c367a", size = 573582, upload-time = "2026-07-08T12:25:31.507Z" },
    { url = "https://files.pythonhosted.org/packages/10/24/b6c5213efe39c19f9e13605644d0cf063b4ddaa33ac2e45b088e23a70e2e/librt-0.13.0-cp313-cp313-pyemscripten_2025_0_wasm32.whl", hash = "sha256:4000d961ff9598ac6ea603c6c836a5ed49bc205ade5fc378b998dfe1e2c36628", size = 82189, upload-time = "2026-07-08T12:25:32.675Z" },
    { url = "https://files.pythonhosted.org/packages/4c/00/d29736be177a906ac0b84a5b04b4fbfa22c776dc2f366de4172b0f968c08/librt-0.13.0-cp313-cp313-win32.whl", hash = "sha256:79e44cff71750d299d61a678e49995b0d5935a9cda238c2574daeca3ba536927", size = 106193, upload-time = "2026-07-08T12:25:33.692Z" },
    { url = "https://files.pythonhosted.org/packages/c8/ac/aff6fb45393cb8912f39dfb156ef6b2d1cadb207ff465fc8f66141054be8/librt-0.13.0-cp313-cp313-win_amd64.whl", hash = "sha256:54dab44a847d5ad1acd05c8a83fe518ae685516ecf4d3f7cc6e3df2a66767650", size = 126962, upload-time = "2026-07-08T12:25:34.769Z" },
    { url = "https://files.pythonhosted.org/packages/d9/3a/d68cb2b334d53fd30fac81d3a489ce4ba0d9506f4df43fcf676b68352b19/librt-0.13.0-cp313-cp313-win_arm64.whl", hash = "sha256:d4cb6fbfdf874340ab5e51450753c0f817b6958a3621125ee695bbc3de866566", size = 112127, upload-time = "2026-07-08T12:25:35.981Z" },
    { url = "https://files.pythonhosted.org/packages/7b/66/f49ae0d592bd45b6941e9a8bafcb6a87cddcd501ee7874707e767f01b585/librt-0.13.0-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:25218d94b1d2cbc0ba1d8a3f9dc9af578d9646e5ed16443a70cde1dfdcce6d71", size = 149818, upload-time = "2026-07-08T12:25:37.203Z" },
    { url = "https://files.pythonhosted.org/packages/3d/50/51c76d74014d04fb95b6506d286808984b78a2f7a41039094e6b2194ac48/librt-0.13.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:f26629539d4893c2957a16c41bb058e1e135c1f150f6a2e25ed047f64cf3f5c6", size = 154071, upload-time = "2026-07-08T12:25:39.399Z" },
    { url = "https://files.pythonhosted.org/packages/b8/fe/f19b0f5f82d5a1f2da736586bc840abd00ce07d6388136ae80b7333883fc/librt-0.13.0-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:a4517d47b2b8af26975a406fba7d314de9696d864252e0257c6ea90238cfe27f", size = 494168, upload-time = "2026-07-08T12:25:40.641Z" },
    { url = "https://files.pythonhosted.org/packages/94/bc/b8550c75775127fd31a5f20e8775997f7b527ad661fc8ddccd7497c064f7/librt-0.13.0-cp314-cp314-manylinux2014_i686.manylinux_2_17_i686.manylinux_2_28_i686.whl", hash = "sha256:f19e181de5b3a1148bb3420b8c4b0b0ea0fce6950099724ad151d6cea5acc180", size = 491054, upload-time = "2026-07-08T12:25:41.905Z" },
    { url = "https://files.pythonhosted.org/packages/30/14/4d0204867623df3f33f86efd3d3692ba5e01321443f4d6eab35a22697618/librt-0.13.0-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:22034924f5b42d5a56371cf271771bfeaabf235a7a8b6264bef2d20013f786c6", size = 523006, upload-time = "2026-07-08T12:25:43.327Z" },
    { url = "https://files.pythonhosted.org/packages/19/0a/c45fc9a260934696bace1ac5df1e148ac92bd71767aee3bf7cd7a4534f4c/librt-0.13.0-cp314-cp314-manylinux_2_34_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:c7897db4e95e22468bdda33d8e012ceacd0182abf001e6389d763f0def6286b9", size = 515058, upload-time = "2026-07-08T12:25:44.541Z" },
    { url = "https://files.pythonhosted.org/packages/13/0a/50c5ce45b326854ef8fa6ae4c36cf5142e5c55315eaf9e51d0ae73ac4da3/librt-0.13.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:1ce61b3746545029d4f5c17d6bd74b676254ad98433086c846ffb5e8fa73f007", size = 534025, upload-time = "2026-07-08T12:25:45.825Z" },
    { url = "https://files.pythonhosted.org/packages/89/2d/08c413c8f93fc13b8103624fce38e5caa86cd08cbbc8465870ab287af54b/librt-0.13.0-cp314-cp314-musllinux_1_2_i686.whl", hash = "sha256:46c330e82565962c761dbce7941be2cff7db674ee807455a8d0cadc5f9b759b0", size = 540557, upload-time = "2026-07-08T12:25:47.059Z" },
    { url = "https://files.pythonhosted.org/packages/b3/c1/93af71fb4a364952210051811dd4e40174e79656b050c89cacac18af3330/librt-0.13.0-cp314-cp314-musllinux_1_2_riscv64.whl", hash = "sha256:375f5af8f99cbaa99dd293af986e3d57caabc9ba81a5d3f021603764854197a1", size = 523201, upload-time = "2026-07-08T12:25:48.392Z" },
    { url = "https://files.pythonhosted.org/packages/c1/6e/9766f07b676a4889d9f8bc2864e9ba5fff165653143ef4dda7df6aa34d16/librt-0.13.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:9320d34c3376ae204b2cd176e8d4883a013934e0aef822f1aed9c536490c275d", size = 565740, upload-time = "2026-07-08T12:25:49.678Z" },
    { url = "https://files.pythonhosted.org/packages/a2/1e/664e3472ce2b6e10e9b83f29d4a36eb982ff6b5a169ae7567bba3a4c4ff5/librt-0.13.0-cp314-cp314-pyemscripten_2026_0_wasm32.whl", hash = "sha256:9af313c66157a69dc69ea0059a66961692250e0dc95af9c385a48ffb770a0d16", size = 81611, upload-time = "2026-07-08T12:25:50.857Z" },
    { url = "https://files.pythonhosted.org/packages/2f/d4/8582a4d65e2234673685e07309d02c230b28a85724eb0acbf13f019b7f6e/librt-0.13.0-cp314-cp314-win32.whl", hash = "sha256:f2a7253458e34f33543551394ae4fe104b497ec2a65ac266074de64c1df82e37", size = 100106, upload-time = "2026-07-08T12:25:52.03Z" },
    { url = "https://files.pythonhosted.org/packages/63/ce/0cb99efe6086b46cd985dc26672166fae312a239690e75871f7fafbd3fc5/librt-0.13.0-cp314-cp314-win_amd64.whl", hash = "sha256:a3dfe4edf10e8ed7e55b026a8bfc2c2a8704218b659cd4bffdf604fab966dc39", size = 121209, upload-time = "2026-07-08T12:25:53.166Z" },
    { url = "https://files.pythonhosted.org/packages/26/85/4f3ccb083a3c9b0d42e223acdb3c3f507953324a59cdcab4826e8e2e3b89/librt-0.13.0-cp314-cp314-win_arm64.whl", hash = "sha256:68a5faee4bba381cb93b5961f684a514cf0053cb92308ff9c792c2fea0b174c6", size = 106404, upload-time = "2026-07-08T12:25:54.253Z" },
    { url = "https://files.pythonhosted.org/packages/b2/77/333191499538c8e8189de7a4cba8e6f49ee949fd6d6e6324b21fd1522466/librt-0.13.0-cp314-cp314t-macosx_10_15_x86_64.whl", hash = "sha256:a38fb81d8376dfa2f8963b265fec07637802b0d01e2a127c19c66cb070fb24f5", size = 159231, upload-time = "2026-07-08T12:25:55.432Z" },
    { url = "https://files.pythonhosted.org/packages/7a/9e/2aa83758f22c278b837a1d8025898434ce2b8bff36678d5330ecaef56dff/librt-0.13.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:d4c8d9bd5abce34b2e75edb3bf37ab0f34e49b1f915a40ae8468eb7c85bc5b46", size = 161300, upload-time = "2026-07-08T12:25:56.585Z" },
    { url = "https://files.pythonhosted.org/packages/bb/c0/86791e936553ca763d6b3c2fb4d31d596cd00e14fa631c283a40ba01559a/librt-0.13.0-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:387e2f1d27e89bffe0d3f520f0da0662c973fd607ca16c1808f8a5085419485e", size = 582056, upload-time = "2026-07-08T12:25:58.144Z" },
    { url = "https://files.pythonhosted.org/packages/a8/d3/a9ec15984a185e000c4d2a16ba28bd623124ad4c38a10974c7ff78e3a893/librt-0.13.0-cp314-cp314t-manylinux2014_i686.manylinux_2_17_i686.manylinux_2_28_i686.whl", hash = "sha256:4f6db193d2e5e0ed60359b9a5a682cd67205d0d3b1e459a867dd4b5c4e7eaa7a", size = 562758, upload-time = "2026-07-08T12:25:59.544Z" },
    { url = "https://files.pythonhosted.org/packages/3c/af/dbe36b78b19c06a55097f99305e4ea9458e2273e6ae16a3cbecaad7ee978/librt-0.13.0-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:0d38604854e8d22faadf683ec6c02bb0f886e2ba56ef981a1c36ee275f21ea22", s

```
ize = 602095, upload-time = "2026-07-08T12:26:00.991Z" },
    { url = "https://files.pythonhosted.org/packages/2a/a8/2966891b4dd2830f5203fbee92ac2c4947653a2390ba73dfa44244fad025/librt-0.13.0-cp314-cp314t-manylinux_2_34_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:371f7ce73026815dafd51c50ce38416e91428b28c4b2ec97cd39271164b0045c", size = 593452, uploaded-time = "2026-07-08T12:26:02.352Z" },
    { url = "https://files.pythonhosted.org/packages/61/f5/4df8bfc8405ecf8c0d525b4d69636f694bdd8620b313ec8b76e54a5926cc/librt-0.13.0-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:3aaedf52171bee90860704c560bc798fe83b76247df47568e0197e9b13c735a0", size = 623729, upload-time = "2026-07-08T12:26:04.294Z" },
    { url = "https://files.pythonhosted.org/packages/d6/13/9ac202dffc8db06f75d06c08c2f9f6ff054be67d21272dcc078fa1cc0c57/librt-0.13.0-cp314-cp314t-musllinux_1_2_i686.whl", hash = "sha256:96bad8725a4f196a798366c25ce075d1f7543a4ec045ffc13e6a7ec095cdab04", size = 617077, upload-time = "2026-07-08T12:26:05.845Z" },
    { url = "https://files.pythonhosted.org/packages/6e/f0/ebe38610716aee5cb28efd95089bb9019209617980277938 1e1c5dcf239c/librt-0.13.0-cp314-cp314t-musllinux_1_2_riscv64.whl", hash = "sha256:6bf6a559ffe4a93bbea6cf31ddf01a7fd9ba342ef51f27beb178e318b74acd61", size = 599561, upload-time = "2026-07-08T12:26:07.21Z" },
    { url = "https://files.pythonhosted.org/packages/4f/5c/c2e72e236fff7abc716d5b1753b8b8cd3ea85ac46fe17d2e7c51d4e1c723/librt-0.13.0-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:301067672387902c55f94b51d5022304b36c966ea9fe1f21caab99a9bef487c9", size = 645511, upload-time = "2026-07-08T12:26:08.562Z" },
    { url = "https://files.pythonhosted.org/packages/0c/99/6203ce619dee940d6bfbe099ec3fe4be00a68e9d60f70abf906cf124fe66/librt-0.13.0-cp314-cp314t-win32.whl", hash = "sha256:5fdcf34f86de8fb66d7dc7589f96ba91c4aa46671200d400e6fd6f109a483f18", size = 104357, upload-time = "2026-07-08T12:26:09.828Z" },
    { url = "https://files.pythonhosted.org/packages/52/dd/843b6314087c41657c7036d7914d8f294bdf9b580aa8513ea0588c8e9a3d/librt-0.13.0-cp314-cp314t-win_amd64.whl", hash = "sha256:260c33e92263fa629b4f6d3c51967a1c2158fe6c33237aaa3ebeac586b085259", size = 126998, upload-time = "2026-07-08T12:26:10.975Z" },
    { url = "https://files.pythonhosted.org/packages/5f/5d/3dcec2884ba1b0806d1408612555c38dd5d68e90156b59f75f6e36435c3a/librt-0.13.0-cp314-cp314t-win_arm64.whl", hash = "sha256:2f281549a4c52ac7bb97997f14353f8bd0e53a34ca0dad1c905cfd0b4a58ae99", size = 110771, upload-time = "2026-07-08T12:26:12.303Z" },
]

[[package]]
name = "llama-index"
version = "0.14.23"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "llama-index-core" },
    { name = "llama-index-embeddings-openai" },
    { name = "llama-index-llms-openai" },
    { name = "nltk" },
]
sdist = { url = "https://files.pythonhosted.org/packages/cb/5c/1f9334c3c0edd880367d0734939256f8536ca94165c6a5fdb455a7bcf180/llama_index-0.14.23.tar.gz", hash = "sha256:eac2049816a7410ff4568490cce4bdff99cda3ab99d59f52f6227dad22cda44b", size = 8566, upload-time = "2026-06-24T19:36:38.241Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e0/92/da2a737bf712fe31c1aaec00f51745797d4570ecdf7e4f5d2b4b93b3d337/llama_index-0.14.23-py3-none-any.whl", hash = "sha256:c205de2442a7186b8e05096f0771f96fa6bdc9603fcad2e07eab1bc96dcf086a", size = 7114, upload-time = "2026-06-24T19:36:37.21Z" },
]

[[package]]
name = "llama-index-core"
version = "0.14.23"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "aiohttp" },
    { name = "aiosqlite" },
    { name = "banks" },
    { name = "dataclasses-json" },
    { name = "deprecated" },
    { name = "dirtyjson" },
    { name = "filetype" },
    { name = "fsspec" },
    { name = "httpx" },
    { name = "llama-index-workflows" },
    { name = "nest-asyncio" },
    { name = "networkx" },
    { name = "nltk" },
    { name = "numpy" },
    { name = "pillow" },
    { name = "platformdirs" },
    { name = "pydantic" },
    { name = "pyyaml" },
    { name = "requests" },
    { name = "setuptools" },
    { name = "sqlalchemy", extra = ["asyncio"] },
    { name = "tenacity" },
    { name = "tiktoken" },
    { name = "tinytag" },
    { name = "tqdm" },
    { name = "typing-extensions" },
    { name = "typing-inspect" },
    { name = "wrapt" },
]
sdist = { url = "https://files.pythonhosted.org/packages/5c/ac/f885ae14317af43a026c909ea4d2083fcee2f0d014f90426b5b9aa1f9912/llama_index_core-0.14.23.tar.gz", hash = "sha256:c4baf2f2ab4f84e95090fe7941e0c87d6c514304f7bd2a749b8fa22164c1822b", size = 11588373, upload-time = "2026-06-24T19:35:55.43Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/ec/d5/05d61f34c01c6578fb758d0a3ddef58d36c6ffa9a9f84a5c9a16262ad94d/llama_index_core-0.14.23-py3-none-any.whl", hash = "sha256:6a54d267826732a8507f81df40785b107f7592af20f451a39a59005147caf84c", size = 11924908, upload-time = "2026-06-24T19:35:52.833Z" },
]

[[package]]
name = "llama-index-embeddings-openai"
version = "0.6.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "llama-index-core" },
    { name = "openai" },
]
sdist = { url = "https://files.pythonhosted.org/packages/06/52/eb56a4887501651fb17400f7f571c1878109ff698efbe0bbac9165a5603d/llama_index_embeddings_openai-0.6.0.tar.gz", hash = "sha256:eb3e6606be81cb89125073e23c97c0a6119dabb4827adbd14697c2029ad73f29", size = 7629, upload-time = "2026-03-12T20:21:27.234Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/4e/d1/4bb0b80f4057903110060f617ef519197194b3ff5dd6153d850c8f5676fa/llama_index_embeddings_openai-0.6.0-py3-none-any.whl", hash = "sha256:039bb1007ad4267e25ddb89a206dfdab862bfb87d58da4271a3919e4f9df4d61", size = 7666, upload-time = "2026-03-12T20:21:28.079Z" },
]

[[package]]
name = "llama-index-instrumentation"
version = "0.5.0"
```

```
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "deprecated" },
    { name = "pydantic" },
]
sdist = { url = "https://files.pythonhosted.org/packages/4e/d0/671b23ccff255c9bce132a84ffd5a6f4541ceefdeab9c1786b08c9722f2e/llama_index_instrumentation-0.5.0.tar.gz", hash = "sha256:eeb724648b25d149de882a5ac9e21c5acb1ce780da214bda2b075341af29ad8e", size = 43831, upload-time = "2026-03-12T20:17:06.742Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/c3/45/6dcaccef44e541ffa138e4b45e33e0d40ab2a7d845338483954fcf77bc75/llama_index_instrumentation-0.5.0-py3-none-any.whl", hash = "sha256:aaab83cddd9dd434278891012d8995f47a3bc7ed1736a371db90965348c56a21", size = 16444, upload-time = "2026-03-12T20:17:05.957Z" },
]

[[package]]
name = "llama-index-llms-openai"
version = "0.7.10"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "llama-index-core" },
    { name = "openai" },
]
sdist = { url = "https://files.pythonhosted.org/packages/74/6e/1ef83a852296c55fe8c529b4476073eccaff150d57aceb3d6319a87dcc89/llama_index_llms_openai-0.7.10.tar.gz", hash = "sha256:aa8dabf08ea0b9740fc9dae9677f05049254be5e929794a886692eaf26973def", size = 27586, upload-time = "2026-07-21T23:16:18.087Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/8b/89/854cb7b2e5be3b3c16561e93f3edd52d8a1b6e031be1aca4519b9fdd3917/llama_index_llms_openai-0.7.10-py3-none-any.whl", hash = "sha256:b76a0c3d4aaa9aba57dc0366ca6b2346c6819bed22973e334676d7a34704e46a", size = 28673, upload-time = "2026-07-21T23:16:17.062Z" },
]

[[package]]
name = "llama-index-workflows"
version = "2.23.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "llama-index-instrumentation" },
    { name = "pydantic" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/cd/6c/14113a4da3ed61f5f07ac87ad1a643ab5d5f2902678ecbf0cc1f8bee4ce7/llama_index_workflows-2.23.0.tar.gz", hash = "sha256:96056dfead58bd33c869d219c8ef56c4abf976c04226d7f6f113ea38db19e56d", size = 136758, upload-time = "2026-08-05T15:21:22.203Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/4a/26/da04a96662ff70f24c9ad4a4ca1a3f41dc521a61bc9c80414d390a860a87/llama_index_workflows-2.23.0-py3-none-any.whl", hash = "sha256:0a038c962294cddb441286075ce3f0a1d4cd21cd899545ab2ad5ef31a0ce5d00", size = 164623, upload-time = "2026-08-05T15:21:20.926Z" },
]

[[package]]
name = "loguru"
version = "0.7.3"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "colorama", marker = "sys_platform == 'win32'" },
    { name = "win32-setctime", marker = "sys_platform == 'win32'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/3a/05/a1dae3dffd1116099471c643b8924f5aa6524411dc6c63fdae648c4f1aca/loguru-0.7.3.tar.gz", hash = "sha256:19480589e77d47b8d85b2c827ad95d49bf31b0dcde16593892eb51dd18706eb6", size = 63559, upload-time = "2024-12-06T11:20:56.608Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/0c/29/0348de65b8cc732daa3e33e67806420b2ae89bdce2b04af740289c5c6c8c/loguru-0.7.3-py3-none-any.whl", hash = "sha256:31a33c10c8e1e10422bfd431aeb5d351c7cf7fa671e3c4df004162264b28220c", size = 61595, upload-time = "2024-12-06T11:20:54.538Z" },
]

[[package]]
name = "mako"
version = "1.4.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "markupsafe" },
]
sdist = { url = "https://files.pythonhosted.org/packages/2a/12/b5fa2353e2754cd67fb9f83793fa48ff42c213a5da7e719869d2301f6ab8/mako-1.4.1.tar.gz", hash = "sha256:d7904710b662996425a21627710c4777c45053146942cf8a7aebf757c92b8c27", size = 410165, upload-time = "2026-08-05T06:10:56.611Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a5/54/12ed58d458474aaab5c3d180173e745a4fe131bb330370596876d19ff60f/mako-1.4.1-py3-none-any.whl", hash = "sha256:a359d9a94a541213958742b2698d0a7757bb83551767bc468a74b9905aba9617", size = 80010, upload-time = "2026-08-05T06:10:58.248Z" },
]

[[package]]
name = "markupsafe"
version = "3.0.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/7e/99/7690b6d4034fffd95959cbe0c02de8deb3098cc577c67bb6a24fe5d7caa7/markupsafe-3.0.3.tar.gz", hash = "sha256:722695808f4b6457b320fdc131280796bdceb04ab50fe1795cd540799ebe1698", size = 80313, upload-time = "2025-09-27T18:37:40.426Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/5a/72/147da192e38635ada20e0a2e1a51cf8823d2119ce8883f7053879c2199b5/markupsafe-3.0.3-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:d53197da72cc091b024dd97249dfc7794d6a56530370992a5e1a08983ad9230e", size = 11615, upload-time = "2025-09-27T18:36:30.854Z" },
    { url = "https://files.pythonhosted.org/packages/9a/81/7e4e08678a1f98521201c3079f77db69fb552acd56067661f8c2f534a718/markupsafe-3.0.3-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:1872df69a4de6aead3491198eaf13810b565bdbeec3ae2dc8780f14458ec73ce", size = 12020, upload-time = "2025-09-27T18:36:31.971Z" },
    { url = "https://files.pythonhosted.org/packages/1e/2c/799f4742efc39633a1b54a92eec4082e4f815314869865d876824c257c1e/markupsafe-3.0.3-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:3a7e8ae81ae39e62a41ec302f972ba6ae23a5c5396c8e60113e9066ef893da0d", size = 24332, upload-time = "2025-09-27T18:36:32.813Z" },
    { url = "https://files.pythonhosted.org/packages/3c/2e/8d0c2ab90a8c1d9a24f0399058ab8519a3279d1bd4289511d74e909f060e/markupsafe-3.0.3-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:d6dd0be5b5b189d31db7cda48b91d7e0a9795f31430b7f271219ab30f1d3ac9d", size = 22947, upload-time = "2025-09-27T18:36:33.86Z" },
    { url = "https://files.pythonhosted.org/packages/2c/54/887f3092a85238093a0b2154bd629c89444f395618842e8b0c41783898ea/markupsafe-3.0.3-cp312-cp312-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:94c6f0bb423f739146aec64595853541634bde58b2135f27f61c1ffd1cd4d16a", size = 21962, upload-time = "2025-09-27T18:36:35.099Z" },
    { url = "https://files.pythonhosted.org/packages/c9/2f/336b8c7b6f4a4d95e91119dc8521402461b74a485558d8f238a68312f11c/markupsafe-3.0.3-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:be8813b57049a7dc738189df53d69395eba14fb99345e0a5994914a3864c8a4b", size = 23760, upload-time = "2025-09-27T18:36:36.001Z" },
    { url = "https://files.pythonhosted.org/packages/32/43/67935f2b7e4982ffb50a4d169b724d74b62a3964bc1a9a527f5ac4f1ee2b/markupsafe-3.0.3-cp312-cp312-musllinux_1_2_riscv64.whl", hash = "sha256:83891d0e9fb81a825d9a6d61e3f07550ca70a076484292a70fde82c4b807286f", size = 21529, upload-time = "2025-09-27T1
```

8:36:36.906Z" },
    { url = "https://files.pythonhosted.org/packages/89/e0/4486f11e51bbba8b0c041098859e869e304d1c261e59244baa3d295d47b7/markupsafe-3.0.3-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:77f0643abe7495da77fb436f50f8dab76dbc6e5fd25d39589a0f1fe6548bfa2b", size = 23015, upload-time = "2025-09-27T18:36:37.868Z" },
    { url = "https://files.pythonhosted.org/packages/2f/e1/78ee7a023dac597a5825441ebd17170785a9dab23de95d2c7508ade94e0e/markupsafe-3.0.3-cp312-cp312-win32.whl", hash = "sha256:d88b440e37a16e651bda4c7c2b930eb586fd15ca7406cb39e211fcff3bf3017d", size = 14540, upload-time = "2025-09-27T18:36:38.761Z" },
    { url = "https://files.pythonhosted.org/packages/aa/5b/bec5aa9bbbb2c946ca2733ef9c4ca91c91b6a24580193e891b5f7dbe8e1e/markupsafe-3.0.3-cp312-cp312-win_amd64.whl", hash = "sha256:26a5784ded40c9e318cfc2bdb30fe164bdb8665ded9cd64d500a34fb42067b1c", size = 15105, upload-time = "2025-09-27T18:36:39.701Z" },
    { url = "https://files.pythonhosted.org/packages/e5/f1/216fc1bbfd74011693a4fd837e7026152e89c4bcf3e77b6692fba9923123/markupsafe-3.0.3-cp312-cp312-win_arm64.whl", hash = "sha256:35add3b638a5d900e807944a078b51922212fb3dedb01633a8defc4b01a3c85f", size = 13906, upload-time = "2025-09-27T18:36:40.689Z" },
    { url = "https://files.pythonhosted.org/packages/38/2f/907b9c7bbba283e68f20259574b13d005c121a0fa4c175f9bed27c4597ff/markupsafe-3.0.3-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:e1cf1972137e83c5d4c136c43ced9ac51d0e124706ee1c8aa8532c1287fa8795", size = 11622, upload-time = "2025-09-27T18:36:41.777Z" },
    { url = "https://files.pythonhosted.org/packages/9c/d9/5f7756922cdd676869eca1c4e3c0cd0df60ed30199ffd775e319089cb3ed/markupsafe-3.0.3-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:116bb52f642a37c115f517494ea5feb03889e04df47eeff5b130b1808ce7c219", size = 12029, upload-time = "2025-09-27T18:36:43.257Z" },
    { url = "https://files.pythonhosted.org/packages/00/07/575a68c754943058c78f30db02ee03a64b3c638586fba6a6dd56830b30a3/markupsafe-3.0.3-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:133a43e73a802c5562be9bbcd03d090aa5a1fe899db609c29e8c8d815c5f6de6", size = 24374, upload-time = "2025-09-27T18:36:44.508Z" },
    { url = "https://files.pythonhosted.org/packages/a9/21/9b05698b46f218fc0e118e1f8168395c65c8a2c750ae2bab54fc4bd4e0e8/markupsafe-3.0.3-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:ccfcd093f13f0f0b7fdd0f198b90053bf7b2f02a3927a30e63f3ccc9df56b676", size = 22980, upload-time = "2025-09-27T18:36:45.385Z" },
    { url = "https://files.pythonhosted.org/packages/7f/71/544260864f893f18b6827315b988c146b559391e6e7e8f7252839b1b846a/markupsafe-3.0.3-cp313-cp313-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:509fa21c6deb7a7a273d629cf5ec029bc209d1a51178615ddf718f5918992ab9", size = 21990, upload-time = "2025-09-27T18:36:46.916Z" },
    { url = "https://files.pythonhosted.org/packages/c2/28/b50fc2f74d1ad761af2f5dcce7492648b983d00a65b8c0e0cb457c82ebbe/markupsafe-3.0.3-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:a4afe79fb3de0b7097d81da19090f4df4f8d3a2b3adaa8764138aac2e44f3af1", size = 23784, upload-time = "2025-09-27T18:36:47.884Z" },
    { url = "https://files.pythonhosted.org/packages/ed/76/104b2aa106a208da8b17a2fb72e033a5a9d7073c68f7e508b94916ed47a9/markupsafe-3.0.3-cp313-cp313-musllinux_1_2_riscv64.whl", hash = "sha256:795e7751525cae078558e679d646ae45574b47ed6e7771863fcc079a6171a0fc", size = 21588, upload-time = "2025-09-27T18:36:48.82Z" },
    { url = "https://files.pythonhosted.org/packages/b5/99/16a5eb2d140087ebd97180d95249b00a03aa87e29cc224056274f2e45fd6/markupsafe-3.0.3-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:8485f406a96febb5140bfeca44a73e3ce5116b2501ac54fe953e488fb1d03b12", size = 23041, upload-time = "2025-09-27T18:36:49.797Z" },
    { url = "https://files.pythonhosted.org/packages/19/bc/e7140ed90c5d61d77cea142eed9f9c303f4c4806f60a1044c13e3f1471d0/markupsafe-3.0.3-cp313-cp313-win32.whl", hash = "sha256:bdd37121970bfd8be76c5fb069c7751683bdf373db1ed6c010162b2a130248ed", size = 14543, upload-time = "2025-09-27T18:36:51.584Z" },
    { url = "https://files.pythonhosted.org/packages/05/73/c4abe620b841b6b791f2edc248f556900667a5a1cf023a6646967ae98335/markupsafe-3.0.3-cp313-cp313-win_amd64.whl", hash = "sha256:9a1abfdc021a164803f4d485104931fb8f8c1efd55bc6b748d2f5774e78b62c5", size = 15113, upload-time = "2025-09-27T18:36:52.537Z" },
    { url = "https://files.pythonhosted.org/packages/f0/3a/fa34a0f7cfef23cf9500d68cb7c32dd64ffd58a12b09225fb03dd37d5b80/markupsafe-3.0.3-cp313-cp313-win_arm64.whl", hash = "sha256:7e68f88e5b8799aa49c85cd116c932a1ac15caaa3f5db09087854d218359e485", size = 13911, upload-time = "2025-09-27T18:36:53.513Z" },
    { url = "https://files.pythonhosted.org/packages/e4/d7/e05cd7efe43a88a17a37b3ae96e79a19e846f3f456fe79c57ca61356ef01/markupsafe-3.0.3-cp313-cp313t-macosx_10_13_x86_64.whl", hash = "sha256:218551f6df4868a8d527e3062d0fb968682fe92054e89978594c28e642c43a73", size = 11658, upload-time = "2025-09-27T18:36:54.819Z" },
    { url = "https://files.pythonhosted.org/packages/99/9e/e412117548182ce2148bdeacdda3bb494260c0b0184360fe0d56389b523b/markupsafe-3.0.3-cp313-cp313t-macosx_11_0_arm64.whl", hash = "sha256:3524b778fe5cfb3452a09d31e7b5adefeea8c5be1d43c4f810ba09f2ceb29d37", size = 12066, upload-time = "2025-09-27T18:36:55.714Z" },
    { url = "https://files.pythonhosted.org/packages/bc/e6/fa0ffcda717ef64a5108eaa7b4f5ed28d56122c9a6d70ab8b72f9f715c80/markupsafe-3.0.3-cp313-cp313t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:4e885a3d1efa2eadc93c894a21770e4bc67899e3543680313b09f139e149ab19", size = 25639, upload-time = "2025-09-27T18:36:56.908Z" },
    { url = "https://files.pythonhosted.org/packages/96/ec/2102e881fe9d25fc16cb4b25d5f5cde50970967ffa5dddafdb771237062d/markupsafe-3.0.3-cp313-cp313t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:8709b08f4a89aa7586de0aadc8da56180242ee0ada3999749b183aa23df95025", size = 23569, upload-time = "2025-09-27T18:36:57.913Z" },
    { url = "https://files.pythonhosted.org/packages/4b/30/6f2fce1f1f205fc9323255b216ca8a235b15860c34b6798f810f05828e32/markupsafe-3.0.3-cp313-cp313t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:b8512a91625c9b3da6f127803b166b629725e68af71f8184ae7e7d54686a56d6", size = 23284, upload-time = "2025-09-27T18:36:58.833Z" },
    { url = "https://files.pythonhosted.org/packages/58/47/4a0ccea4ab9f5dcb6f79c0236d954acb382202721e704223a8aafa38b5c8/markupsafe-3.0.3-cp313-cp313t-musllinux_1_2_aarch64.whl", hash = "sha256:9b79b7a16f7fedff2495d684f2b59b0457c3b493778c9eed31111be64d58279f", size = 24801, upload-time = "2025-09-27T18:36:59.739Z" },
    { url = "https://files.pythonhosted.org/packages/6a/70/3780e9b72180b6fecb83a4814d84c3bf4b4ae4bf0b19c27196104149734c/markupsafe-3.0.3-cp313-cp313t-musllinux_1_2_riscv64.whl", hash = "sha256:12c63dfb4a98206f045aa9563db46507995f7ef6d83b2f68eda65c307c6829eb", size = 22769, upload-time = "2025-09-27T18:37:00.719Z" },
    { url = "https://files.pythonhosted.org/packages/98/c5/c03c7f4125180fc215220c035beac6b9cb684bc7a067c84fc69414d315f5/markupsafe-3.0.3-cp313-cp313t-musllinux_1_2_x86_64.whl", hash = "sha256:8f71bc33915be5186016f675cd83a1e08523649b0e33efdb898db577ef5bb009", size = 23642, upload-time = "2025-09-27T18:37:01.673Z" },
    { url = "https://files.pythonhosted.org/packages/80/d6/2d1b89f6ca4bff1036499b1e29a1d02d282259f3681540e16563f27ebc23/markupsafe-3.0.3-cp313-cp313t-win32.whl", hash = "sha256:69c0b73548bc525c8cb9a251cddf1931d1db4d2258e9599c28c07ef3580ef354", size = 14612, upload-time = "2025-09-27T18:37:02.639Z" },
    { url = "https://files.pythonhosted.org/packages/2b/98/e48a4bfba0a0ffcf9925fe2d69240bfaa19c6f7507b8cd09c70684a53c1e/markupsafe-3.0.3-cp313-cp313t-win_amd64.whl", hash = "sha256:1b4b79e8ebf6b55351f0d91fe80f893b4743f104bff22e90697db1590e47a218", size = 15200, upload-time = "2025-09-27T18:37:03.582Z" },
    { url = "https://files.pythonhosted.org/packages/0e/72/e3cc540f351f316e9ed0f092757459afbc595824ca724cbc5a5d4263713f/markupsafe-3.0.3-cp313-cp313t-win_arm64.whl", hash = "sha256:ad2cf8aa28b8c020ab2fc8287b0f823d0a7d8630784c31e9ee5edea20f406287", size = 13973, upload-time = "2025-09-27T18:37:04.929Z" },
    { url = "https://files.pythonhosted.org/packages/33/8a/8e42d4838cd89b7dde187011e97fe6c3af66d8c044997d2183fbd6d31352/markupsafe-3.0.3-cp314-cp314-macosx_10_13_x86_64.whl", hash = "sha256:eaa9599de571d72e2daf60164784109f19978b327a3910d3e9de8c97b5b70cfe", size = 11619, upload-time = "2025-09-27T18:37:06.342Z" },
    { url = "https://files.pythonhosted.org/packages/b5/64/7660f8a4a8e53c924d0fa05dc3a55c9cee10bbd82b11c5afb27d44b096ce/markupsafe-3.0.3-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:c47a551199eb8eb2121d4f0f15ae0f923d31350ab9280078d1e5f12b249e0026", size = 12029, upload-time = "2025-09-27T18:37:07.213Z" },
    { url = "https://files.pythonhosted.org/packages/da/ef/e648bfd021127bef5fa12e1720ffed0c6cbb8310c8d9bea7266337ff06de/markupsafe-3.0.3-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:f34c41761022dd093b4b6896d4810782ffbabe30f2d443ff5f083e0cbbb8c737", size = 24408, upload-time = "2025-09-27T18:37:09.572Z" },
    { url = "https://files.pythonhosted.org/packages/41/3c/a36c2450754618e62008bf7435ccb0f88053e07592e6028a34776213d877/markupsafe-3.0.3-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:457a69a9577064c05a97c41f4e65148652db078a3a509039e64d3467b9e7ef97", size = 23005, upload-time = "2025-09-27T18:37:10.58Z" },
    { url = "https://files.pythonhosted.org/packages/bc/20/b7fdf89a8456b099837cd1dc21974632a02a999ec9bf7ca3e490aacd98e7/markupsafe-3.0.3-cp314-cp314-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:e8afc3f2ccfa24215f8cb28dcf43f0113ac3c37c2f0f0806d8c70e4228c5cf4d", size = 22048, upload-time = "2025-09-27T18:37:11.547Z" },
    { url = "https://files.pythonhosted.org/packages/9a/a7/591f592afdc734f47db08a75793a55d7fbcc6902a723ae4cfbab61010cc5/markupsafe-3.0.3-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:ec15a59cf5af7be74194f7ab02d0f59a62bdcf1a537677ce67a2537c9b87fcda", size = 23821, upload-time = "2025-09-27T18:37:12.48Z" },
    { url = "https://files.pythonhosted.org/packages/7d/33/45b24e4f44195b26521bc6f1a82197118f74df348556594bd2262bda1038/markupsafe-3.0.3-cp314-cp314-musllinux_1_2_riscv64.whl", hash = "sha256:0eb9ff8191e8498cca014656ae6b8d61f39da5f95b488805da4bb029cccbfbaf", size = 21606, upload-time = "2025-09-27T18:37:13.485Z" },
    { url = "https://files.pythonhosted.org/packages/ff/0e/53dfaca23a69fbfbbf17a4b64072090e70717344c52eaaaa9c5ddff1e5f0/markupsafe-3.0.3-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:2713baf880df847f2bece4230d4d094280f4e67b1e813eec43b4c0e144a34ffe", size = 23043, upload-time = "2025-09-27T18:37:14.408Z" },
    { url = "https://files.pythonhosted.org/packages/46/11/f333a06fc16236d5238bfe74daccbca41459dcd8d1fa952e8fbd5dccfb70/markupsafe-3.0.3-cp314-cp314-win32.whl", hash = "sha256:729586769a26dbceff69f7a7dbbf59ab6572b99d94576a5592625d5b411576b9", size = 14747, upload-time = "2025-09-27T18:37:15.36Z" },
    { url = "https://files.pythonhosted.org/packages/28/52/182836104b33b444e400b14f797212f720cbc9ed6ba34c800639d154e821/markupsafe-3.0.3-cp314-cp314-win_amd64.whl", hash = "sha256:bdc919ead48f234740ad807933cdf545180bfbe9342c2bb451556db2ed958581", size = 15341, upload-time = "2025-09-27T18:37:16.496Z

```
" },
    { url = "https://files.pythonhosted.org/packages/6f/18/acf23e91bd94fd7b3031558b1f013adfa21a8e407a3fdb32745538730382/markupsafe-3.0.3-cp314-cp314-win_arm64.whl", hash = "sha256:5a7d5dc5140555cf21a6fefbdbf8723f06fcd2f63ef108f2854de715e4422cb4", size = 14073, upload-time = "2025-09-27T18:37:17.476Z" },
    { url = "https://files.pythonhosted.org/packages/3c/f0/57689aa4076e1b43b15fdfa646b04653969d50cf30c32a102762be2485da/markupsafe-3.0.3-cp314-cp314t-macosx_10_13_x86_64.whl", hash = "sha256:1353ef0c1b138e1907ae78e2f6c63ff67501122006b0f9abad68fda5f4ffc6ab", size = 11661, upload-time = "2025-09-27T18:37:18.453Z" },
    { url = "https://files.pythonhosted.org/packages/89/c3/2e67a7ca217c6912985ec766c6393b636fb0c2344443ff9d91404dc4c79f/markupsafe-3.0.3-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:1085e7fbddd3be5f89cc898938f42c0b3c711fdcb37d75221de2666af647c175", size = 12069, upload-time = "2025-09-27T18:37:19.332Z" },
    { url = "https://files.pythonhosted.org/packages/f0/00/be561dce4e6ca66b15276e184ce4b8aec61fe83662cce2f7d72bd3249d28/markupsafe-3.0.3-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:1b52b4fb9df4eb9ae465f8d0c228a00624de2334f216f178a995ccdcf82c4634", size = 25670, upload-time = "2025-09-27T18:37:20.245Z" },
    { url = "https://files.pythonhosted.org/packages/50/09/c419f6f5a92e5fadde27efd190eca90f05e1261b10dbd8cbcb39cd8ea1dc/markupsafe-3.0.3-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:fed51ac40f757d41b7c48425901843666a6677e3e8eb0abcff09e4ba6e664f50", size = 23598, upload-time = "2025-09-27T18:37:21.177Z" },
    { url = "https://files.pythonhosted.org/packages/22/44/a0681611106e0b2921b3033fc19bc53323e0b50bc70cffdd19f7d679bb66/markupsafe-3.0.3-cp314-cp314t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:f190daf01f13c72eac4efd5c430a8de82489d9cff23c364c3ea822545032993e", size = 23261, upload-time = "2025-09-27T18:37:22.167Z" },
    { url = "https://files.pythonhosted.org/packages/5f/57/1b0b3f100259dc9fffe780cfb60d4be71375510e435efec3d116b6436d43/markupsafe-3.0.3-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:e56b7d45a839a697b5eb268c82a71bd8c7f6c94d6fd50c3d577fa39a9f1409f5", size = 24835, upload-time = "2025-09-27T18:37:23.296Z" },
    { url = "https://files.pythonhosted.org/packages/26/6a/4bf6d0c97c4920f1597cc14dd720705eca0bf7c787aebc6bb4d1bead5388/markupsafe-3.0.3-cp314-cp314t-musllinux_1_2_riscv64.whl", hash = "sha256:f3e98bb3798ead92273dc0e5fd0f31ade220f59a266ffd8a4f6065e0a3ce0523", size = 22733, upload-time = "2025-09-27T18:37:24.237Z" },
    { url = "https://files.pythonhosted.org/packages/14/c7/ca723101509b518797fedc2fdf79ba57f886b4aca8a7d31857ba3ee8281f/markupsafe-3.0.3-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:5678211cb9333a6468fb8d8be0305520aa073f50d17f089b5b4b477ea6e67fdc", size = 23672, upload-time = "2025-09-27T18:37:25.271Z" },
    { url = "https://files.pythonhosted.org/packages/fb/df/5bd7a48c256faecd1d36edc13133e51397e41b73bb77e1a69deab746ebac/markupsafe-3.0.3-cp314-cp314t-win32.whl", hash = "sha256:915c04ba3851909ce68ccc2b8e2cd691618c4dc4c4232fb7982bca3f41fd8c3d", size = 14819, upload-time = "2025-09-27T18:37:26.285Z" },
    { url = "https://files.pythonhosted.org/packages/1a/8a/0402ba61a2f16038b48b39bccca271134be00c5c9f0f623208399333c448/markupsafe-3.0.3-cp314-cp314t-win_amd64.whl", hash = "sha256:4faffd047e07c38848ce017e8725090413cd80cbc23d86e55c587bf979e579c9", size = 15426, upload-time = "2025-09-27T18:37:27.316Z" },
    { url = "https://files.pythonhosted.org/packages/70/bc/6f1c2f612465f5fa89b95bead1f44dcb607670fd42891d8fdcd5d039f4f4/markupsafe-3.0.3-cp314-cp314t-win_arm64.whl", hash = "sha256:32001d6a8fc98c8cb5c947787c5d08b0a50663d139f1305bac5885d98d9b40fa", size = 14146, upload-time = "2025-09-27T18:37:28.327Z" },
]

[[package]]
name = "marshmallow"
version = "3.26.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "packaging" },
]
sdist = { url = "https://files.pythonhosted.org/packages/55/79/de6c16cc902f4fc372236926b0ce2ab7845268dcc30fb2fbb7f71b418631/marshmallow-3.26.2.tar.gz", hash = "sha256:bbe2adb5a03e6e3571b573f42527c6fe926e17467833660bebd11593ab8dfd57", size = 222095, upload-time = "2025-12-22T06:53:53.309Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/be/2f/5108cb3ee4ba6501748c4908b908e55f42a5b66245b4cfe0c99326e1ef6e/marshmallow-3.26.2-py3-none-any.whl", hash = "sha256:013fa8a3c4c276c24d26d84ce934dc964e2aa794345a0f8c7e5a7191482c8a73", size = 50964, upload-time = "2025-12-22T06:53:51.801Z" },
]

[[package]]
name = "mcp"
version = "2.0.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "httpx2" },
    { name = "jsonschema" },
    { name = "mcp-types" },
    { name = "opentelemetry-api" },
    { name = "pydantic" },
    { name = "pyjwt", extra = ["crypto"] },
    { name = "python-multipart" },
    { name = "pywin32", marker = "sys_platform == 'win32'" },
    { name = "sse-starlette" },
    { name = "starlette" },
    { name = "typing-extensions" },
    { name = "typing-inspection" },
    { name = "uvicorn", marker = "sys_platform != 'emscripten'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/74/33/32d4dff2c95bb5d897c3ef4c83649a08996b17b58f0a326d2495d4c81179/mcp-2.0.0.tar.gz", hash = "sha256:0f440e735c13ece8bb19bc62cf0b86f4313448432fbb77d35e14034f4e050728", size = 1662284, upload-time = "2026-07-28T13:45:32.346Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/67/72/7d7897418912c1d12e87556630dfb7bf0eac71160e9bef8b447960804ee3/mcp-2.0.0-py3-none-any.whl", hash = "sha256:1cb4c75d2d2c7b8c1d756355e5d82a39f2822cc7f13e22a2051d7ca3592349d6", size = 349980, upload-time = "2026-07-28T13:45:28.853Z" },
]

[[package]]
name = "mcp-types"
version = "2.0.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "pydantic" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/bb/56/9b8e1c152f61f6c6b07c4b5896c88c7d0ae90bac6ee6306f852fcc5c1eb0/mcp_types-2.0.0.tar.gz", hash = "sha256:d7d939b9285c9961ae8866ba75ef85da34d12bafe276efbf4eb6a131786d8379", size = 66632, upload-time = "2026-07-28T13:45:33.804Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/f5/4c/c78d78c3d52b0ac594ad7cc8ef5972adfe070e3597a8a4c6ce0cd39196ea/mcp_types-2.0.0-py3-none-any.whl", hash = "sha256:6b2de797ca2797f568b79529e1b25948e34de511bcc0bd82fef1039a6d1b8eb0", size = 69649, upload-time = "2026-07-28T13:45:30.713Z" },
]

[[package]]
name = "multidict"
version = "6.7.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/1a/c2/c2d94cbe6ac1753f3fc980da97b3d930efe1da3af3c9f5125354436c073d/multidict-6.7.1.tar.gz", hash = "sha256:ec6652a1bee61c53a3e5776b6049172c53b6aaba34f18c9ad04f82712bac623d", size = 102010, upload-time = "2026-01-26T02:46:45.979Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/8d/9c/f20e0e2cf80e4b2e4b1c365bf5fe104ee633c751a724246262db8f1a0b13/multidict-6.7.1-cp312-cp312-macosx_10_13_universal2.whl", hash = "sha256:a90f75c956e32891a4eda3639ce6dd86e87105271f43d43442a3aedf3cddf172", size = 76893, upload-time = "2026-01-26T02:43:52.754Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/fe/cf/18ef143a81610136d3da8193da9d80bfe1cb548a1e2d1c775f26b23d024a/multidict-6.7.1-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:3fccb473e87eaa1382689053e4a4618e7ba7b9b9b8d6adf2027ee474597128cd", size = 45456, upload-time = "2026-01-26T02:43:53.893Z" },
    { url = "https://files.pythonhosted.org/packages/a9/65/1caac9d4cd32e843390868344 6eebc953e82d22b03d10d41a5f0fefe991b/multidict-6.7.1-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:b0fa96985700739c4c7853a43c0b3e169360d6857780021bfc6d0f1ce7c123e7", size = 43872, upload-time = "2026-01-26T02:43:55.041Z" },
    { url = "https://files.pythonhosted.org/packages/cf/3b/d6bd75dc4f3ff7c73766e04e705b00ed6dbbaccf670d9e05a12b006f5a21/multidict-6.7.1-cp312-cp312-manylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:cb2a55f408c3043e42b40cc8eecd575afa27b7e0b956dfb190de0f8499a57a53", size = 251018, upload-time = "2026-01-26T02:43:56.198Z" },
    { url = "https://files.pythonhosted.org/packages/fd/80/c959c5933adedb9ac15152e4067c702a808ea183a8b64cf8f31af8ad3155/multidict-6.7.1-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:eb0ce7b2a32d09892b3dd6cc44877a0d02a33241fafca5f25c8b6b62374f8b75", size = 258883, upload-time = "2026-01-26T02:43:57.499Z" },
    { url = "https://files.pythonhosted.org/packages/86/85/7ed40adafea3d4f1c8b916e3b5cc3a8e07dfcdcb9cd72800f4ed3ca1b387/multidict-6.7.1-cp312-cp312-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:c3a32d23520ee37bf327d1e1a656fec76a2edd5c038bf43eddfa0572ec49c60b", size = 242413, upload-time = "2026-01-26T02:43:58.755Z" },
    { url = "https://files.pythonhosted.org/packages/d2/57/b8565ff533e48595503c785f8361ff9a4fde4d67de25c207cd0ba3befd03/multidict-6.7.1-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:9c90fed18bffc0189ba814749fdcc102b536e83a9f738a9003e569acd540a733", size = 268404, upload-time = "2026-01-26T02:44:00.216Z" },
    { url = "https://files.pythonhosted.org/packages/e0/50/9810c5c29350f7258180dfdcb2e52783a0632862eb334c4896ac717cebcb/multidict-6.7.1-cp312-cp312-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:da62917e6076f512daccfbbde27f46fed1c98fee202f0559adec8ee0de67f71a", size = 269456, upload-time = "2026-01-26T02:44:02.202Z" },
    { url = "https://files.pythonhosted.org/packages/f3/8d/5e5be3ced1d12966fefb5c4ea3b2a5b480afcea36406559442c6e31d4a48/multidict-6.7.1-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:bfde23ef6ed9db7eaee6c37dcec08524cb43903c60b285b172b6c094711b3961", size = 256322, upload-time = "2026-01-26T02:44:03.56Z" },
    { url = "https://files.pythonhosted.org/packages/31/6e/d8a26d81ac166a5592782d208dd90dfdc0a7a218adaa52b45a672b46c122/multidict-6.7.1-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:3758692429e4e32f1ba0df23219cd0b4fc0a52f476726fff9337d1a57676a582", size = 253955, upload-time = "2026-01-26T02:44:04.845Z" },
    { url = "https://files.pythonhosted.org/packages/59/4c/7c672c8aad41534ba619bcd4ade7a0dc87ed6b8b5c06149b85d3dd03f0cd/multidict-6.7.1-cp312-cp312-musllinux_1_2_armv7l.whl", hash = "sha256:398c1478926eca669f2fd6a5856b6de9c0acf23a2cb59a14c0ba5844fa38077e", size = 251254, upload-time = "2026-01-26T02:44:06.133Z" },
    { url = "https://files.pythonhosted.org/packages/7b/bd/84c24de512cbafbdbc39439f74e967f19570ce7924e3007174a29c348916/multidict-6.7.1-cp312-cp312-musllinux_1_2_i686.whl", hash = "sha256:c102791b1c4f3ab36ce4101154549105a53dc828f016356b3e3bcae2e3a039d3", size = 252059, upload-time = "2026-01-26T02:44:07.518Z" },
    { url = "https://files.pythonhosted.org/packages/fa/ba/f5449385510825b73d01c2d4087bf6d2fccc20a2d42ac34df93191d3dd03/multidict-6.7.1-cp312-cp312-musllinux_1_2_ppc64le.whl", hash = "sha256:a088b62bd733e2ad12c50dad01b7d0166c30287c166e137433d3b410add807a6", size = 263588, upload-time = "2026-01-26T02:44:09.382Z" },
    { url = "https://files.pythonhosted.org/packages/d7/11/afc7c677f68f75c84a69fe37184f0f82fce13ce4b92f49f3db280b7e92b3/multidict-6.7.1-cp312-cp312-musllinux_1_2_s390x.whl", hash = "sha256:3d51ff4785d58d3f6c91bdbffcb5e1f7ddfda557727043aa20d20ec4f65e324a", size = 259642, upload-time = "2026-01-26T02:44:10.73Z" },
    { url = "https://files.pythonhosted.org/packages/2b/17/ebb9644da78c4ab36403739e0e6e0e30ebb135b9caf3440825001a0bddcb/multidict-6.7.1-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:fc5907494fccf3e7d3f94f95c91d6336b092b5fc83811720fae5e2765890dfba", size = 251377, upload-time = "2026-01-26T02:44:12.042Z" },
    { url = "https://files.pythonhosted.org/packages/ca/a4/840f5b97339e27846c46307f2530a2805d9d537d8b8bd416af031cad7fa0/multidict-6.7.1-cp312-cp312-win32.whl", hash = "sha256:28ca5ce2fd9716631133d0e9a9b9a745ad7f60bac2bccafb56aa380fc0b6c511", size = 41887, upload-time = "2026-01-26T02:44:14.245Z" },
    { url = "https://files.pythonhosted.org/packages/80/31/0b2517913687895f5904325c2069d6a3b78f66cc641a86a2baf75a05dcbb/multidict-6.7.1-cp312-cp312-win32_amd64.whl", hash = "sha256:fcee94dfbd638784645b066074b338bc9cc155d4b4bffa4adce1615c5a426c19", size = 46053, upload-time = "2026-01-26T02:44:15.371Z" },
    { url = "https://files.pythonhosted.org/packages/0c/5b/aba28e4ee4006ae4c7df8d327d31025d760ffa992ea23812a601d226e682/multidict-6.7.1-cp312-cp312-win32_arm64.whl", hash = "sha256:ba0a9fb644d0c1a2194cf7ffb043bd852cea63a57f66fbd33959f7dae18517bf", size = 43307, upload-time = "2026-01-26T02:44:16.852Z" },
    { url = "https://files.pythonhosted.org/packages/f2/22/929c141d6c0dba87d3e1d38fbdf1ba8baba86b7776469f2bc2d3227a1e67/multidict-6.7.1-cp313-cp313-macosx_10_13_universal2.whl", hash = "sha256:2b41f5fed0ed563624f1c17630cb9941cf2309d4df00e494b551b5f3e3d67a23", size = 76174, upload-time = "2026-01-26T02:44:18.509Z" },
    { url = "https://files.pythonhosted.org/packages/c7/75/bc704ae15fee974f8fccd871305e254754167dce5f9e42d88a2def741a1d/multidict-6.7.1-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:84e61e3af5463c19b67ced91f6c634effb89ef8bfc5ca0267f954451ed4bb6a2", size = 45116, upload-time = "2026-01-26T02:44:19.745Z" },
    { url = "https://files.pythonhosted.org/packages/79/76/55cd7186f498ed080a18440c9013011eb548f77ae1b297206d030eb1180a/multidict-6.7.1-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:935434b9853c7c112eee7ac891bc4cb86455aa631269ae35442cb316790c1445", size = 43524, upload-time = "2026-01-26T02:44:21.571Z" },
    { url = "https://files.pythonhosted.org/packages/e9/3c/414842ef8d5a1628d68edee29ba0e5bcf235dbfb3ccd3ea303a7fe8c72ff/multidict-6.7.1-cp313-cp313-manylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:432feb25a1cb67fe82a9680b4d65fb542e4635cb3166cd9c01560651ad60f177", size = 249368, upload-time = "2026-01-26T02:44:22.803Z" },
    { url = "https://files.pythonhosted.org/packages/f6/32/befed7f74c458b4a525e60519fe8d87eef72bb1e99924fa2b0f9d97a221e/multidict-6.7.1-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:e82d14e3c948952a1a85503817e038cba5905a3352de76b9a465075d072fba23", size = 256952, upload-time = "2026-01-26T02:44:24.306Z" },
    { url = "https://files.pythonhosted.org/packages/03/d6/c878a44ba877f366630c860fdf74bfb203c33778f12b6ac274936853c451/multidict-6.7.1-cp313-cp313-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:4cfb48c6ea66c83bcaaf7e4dfa7ec1b6bbcf751b7db85a328902796dfde4c060", size = 240317, upload-time = "2026-01-26T02:44:25.772Z" },
    { url = "https://files.pythonhosted.org/packages/68/49/57421b4d7ad2e9e60e25922b08ceb37e077b90444bde6ead629095327a6f/multidict-6.7.1-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:1d540e51b7e8e170174555edecddbd5538105443754539193e3e1061864d444d", size = 267132, upload-time = "2026-01-26T02:44:27.648Z" },
    { url = "https://files.pythonhosted.org/packages/b7/fe/ec0edd52ddbcea2a2e89e174f0206444a61440b40f39704e64dc807a70bd/multidict-6.7.1-cp313-cp313-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:273d23f4b40f3dce4d6c8a821c741a86dec62cded82e1175ba3d99be128147ed", size = 268140, upload-time = "2026-01-26T02:44:29.588Z" },
    { url = "https://files.pythonhosted.org/packages/b0/73/6e1b01cbeb458807aa0831742232dbdd1fa92bfa33f52a3f176b4ff3dc11/multidict-6.7.1-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9d624335fd4fa1c08a53f8b4be7676ebde19cd092b3895c421045ca87895b429", size = 254277, upload-time = "2026-01-26T02:44:30.902Z" },
    { url = "https://files.pythonhosted.org/packages/6a/b2/5fb8c124d7561a4974c342bc8c778b471ebbeb3cc17df696f034a7e9afe7/multidict-6.7.1-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:12fad252f8b267cc75b66e8fc51b3079604e8d43a75428ffe193cd9e2195dfd6", size = 252291, upload-time = "2026-01-26T02:44:32.31Z" },
    { url = "https://files.pythonhosted.org/packages/5a/96/51d4e4e06bcce92577fcd488e22600bd38e4fd59c20cb49434d054903bd2/multidict-6.7.1-cp313-cp313-musllinux_1_2_armv7l.whl", hash = "sha256:03ede2a6ffbe8ef936b92cb4529f27f42be7f56afcdab5ab739cd5f27fb1cbf9", size = 250156, upload-time = "2026-01-26T02:44:33.734Z" },
    { url = "https://files.pythonhosted.org/packages/db/6b/420e173eec5fba721a50e2a9f89eda89d9c98fded1124f8d5c675f7a0c0f/multidict-6.7.1-cp313-cp313-musllinux_1_2_i686.whl", hash = "sha256:90efbcf47dbe33dcf643a1e400d67d59abeac5db07dc3f27d6bdeae497a2198c", size = 249742, upload-time = "2026-01-26T02:44:35.222Z" },
    { url = "https://files.pythonhosted.org/packages/44/a3/ec5b5bd98f306bc2aa297b8c6f11a46714a56b1e6ef5ebda50a4f5d7c5fb/multidict-6.7.1-cp313-cp313-musllinux_1_2_ppc64le.whl", hash = "sha256:5c4b9bfc148f5a91be9244d6264c53035c8a0dcd2f51f1c3c6e30e30ebaa1c84", size = 262221, upload-time = "2026-01-26T02:44:36.604Z" },
    { url = "https://files.pythonhosted.org/packages/cd/f7/e8c0d0da0cd1e28d10e624604e1a36bcc3353aaebdfdc3a43c72bc683a12/multidict-6.7.1-cp313-cp313-musllinux_1_2_s390x.whl", hash = "sha256:401c5a650f3add2472d1d288c26deebc540f99e2fb83e9525007a74cd2116f1d", size = 258664, upload-time = "2026-01-26T02:44:38.008Z" },
    { url = "https://files.pythonhosted.org/packages/52/da/151a44e8016dd33feed44f730bd856a66257c1ee7aed4f44b649fb7edeb3/multidict-6.7.1-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:97891f3b1b3ffbded884e2916cacf3c6fc87b66bb0dde46f7357404750559f33", size = 249490, upload-time = "2026-01-26T02:44:39.386Z" },
    { url = "https://files.pythonhosted.org/packages/87/af/a3b86bf9630b732897f6fc3f4c4714b90aa4361983ccbdcd6c0339b21b0c/multidict-6.7.1-cp313-cp313-win32.whl", hash = "sha256:e1c5988359516095535c4301af38d8a8838534158f649c05dd1050222321bcb3", size = 41695, upload-time = "2026-01-26T02:44:41.318Z" },
    { url = "https://files.pythonhosted.org/packages/b2/35/e994121b0e90e46134673422dd564623f93304614f5d11886b1b3e06f503/multidict-6.7.1-cp313-cp313-win32_amd64.whl", hash = "sha256:960c83bf01a95b12b08fd54324a4eb1d5b52c88932b5cba5d6e712bb3ed12eb5", size = 45884, upload-time = "2026-01-26T02:44:42.488Z" },
    { url = "https://files.pythonhosted.org/packages/ca/61/42d3e5dbf661242a69c97ea363f2d7b46c567da8eadef8890022be6e2ab0/multidict-6.7.1-cp313-cp313-win32_arm64.whl", hash = "sha256:563fe25c678aaba333d5399408f5ec3c383ca5b663e7f774dd179a520b8144df", size = 43122, upload-time = "2026-01-26T02:44:43.664Z" },
    { url = "https://files.pythonhosted.org/packages/6d/b3/e6b21c6c4f314bb956016b0b3ef2162590a529b84cb831c257519e7fde44/multidict-6.7.1-cp313-cp313t-macosx_10_13_universal2.whl", hash = "sha256:c76c4bec1538375dad9d452d246ca5368ad6e1c9039dadcf007ae59c70619ea1", size = 83175, upload-time = "2026-01-26
```

T02:44:44.894Z" },
    { url = "https://files.pythonhosted.org/packages/fb/76/23ecd2abfe0957b234f6c960f4ade497f55f2c16aeb684d4ecdbf1c95791/multidict-6.7.1-cp313-cp313t-macosx_10_13_x86_64.whl", hash = "sha256:57b46b24b5d5ebcc978da4ec23a819a9402b4228b8a90d9c656422b4bdd8a963", size = 48460, upload-time = "2026-01-26T02:44:46.106Z" },
    { url = "https://files.pythonhosted.org/packages/c4/57/a0ed92b23f3a042c36bc4227b72b97eca803f5f1801c1ab77c8a212d455e/multidict-6.7.1-cp313-cp313t-macosx_11_0_arm64.whl", hash = "sha256:e954b24433c768ce78ab7929e84ccf3422e46deb45a4dc9f93438f8217fa2d34", size = 46930, upload-time = "2026-01-26T02:44:47.278Z" },
    { url = "https://files.pythonhosted.org/packages/b5/66/02ec7ace29162e447f6382c495dc95826bf931d3818799bbef11e8f7df1a/multidict-6.7.1-cp313-cp313t-manylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:3bd231490fa7217cc832528e1cd8752a96f0125ddd2b5749390f7c3ec8721b65", size = 242582, upload-time = "2026-01-26T02:44:48.604Z" },
    { url = "https://files.pythonhosted.org/packages/58/18/64f5a795e7677670e872673aca234162514696274597b3708b2c0d276cce/multidict-6.7.1-cp313-cp313t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:253282d70d67885a15c8a7716f3a73edf2d635793ceda8173b9ecc21f2fb8292", size = 250031, upload-time = "2026-01-26T02:44:50.544Z" },
    { url = "https://files.pythonhosted.org/packages/c8/ed/e192291dbbe51a8290c5686f482084d31bcd9d09af24f63358c3d42fd284/multidict-6.7.1-cp313-cp313t-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:0b4c48648d7649c9335cf1927a8b87fa692de3dcb15faa676c6a6f1f1aabda43", size = 228596, upload-time = "2026-01-26T02:44:51.951Z" },
    { url = "https://files.pythonhosted.org/packages/1e/7e/3562a15a60cf747397e7f2180b0a11dc0c38d9175a650e75fa1b4d325e15/multidict-6.7.1-cp313-cp313t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:98bc624954ec4d2c7cb074b8eefc2b5d0ce7d482e410df446414355d158fe4ca", size = 257492, upload-time = "2026-01-26T02:44:53.902Z" },
    { url = "https://files.pythonhosted.org/packages/24/02/7d0f9eae92b5249bb50ac1595b295f10e263dd0078ebb55115c31e0eaccd/multidict-6.7.1-cp313-cp313t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:1b99af4d9eec0b49927b4402bcbb58dea89d3e0db8806a4086117019939ad3dd", size = 255899, upload-time = "2026-01-26T02:44:55.316Z" },
    { url = "https://files.pythonhosted.org/packages/00/e3/9b60ed9e23e64c73a5cde95269ef1330678e9c6e34dd4eb6b431b85b5a10/multidict-6.7.1-cp313-cp313t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:6aac4f16b472d5b7dc6f66a0d49dd57b0e0902090be16594dc9ebfd3d17c47e7", size = 247970, upload-time = "2026-01-26T02:44:56.783Z" },
    { url = "https://files.pythonhosted.org/packages/3e/06/538e58a63ed5cfb0bd4517e346b91da32fde409d839720f664e9a4ae4f9d/multidict-6.7.1-cp313-cp313t-musllinux_1_2_aarch64.whl", hash = "sha256:21f830fe223215dffd51f538e78c172ed7c7f60c9b96a2bf05c4848ad49921c3", size = 245060, upload-time = "2026-01-26T02:44:58.195Z" },
    { url = "https://files.pythonhosted.org/packages/b2/2f/d743a3045a97c895d401e9bd29aaa09b94f5cbdf1bd561609e5a6c431c70/multidict-6.7.1-cp313-cp313t-musllinux_1_2_armv7l.whl", hash = "sha256:f5dd81c45b05518b9aa4da4aa74e1c93d715efa234fd3e8a179df611cc85e5f4", size = 235888, upload-time = "2026-01-26T02:44:59.57Z" },
    { url = "https://files.pythonhosted.org/packages/38/83/5a325cac191ab28b63c52f14f1131f3b0a55ba3b9aa65a6d0bf2a9b921a0/multidict-6.7.1-cp313-cp313t-musllinux_1_2_i686.whl", hash = "sha256:eb304767bca2bb92fb9c5bd33cedc95baee5bb5f6c88e63706533a1c06ad08c8", size = 243554, upload-time = "2026-01-26T02:45:01.054Z" },
    { url = "https://files.pythonhosted.org/packages/20/1f/9d2327086bd15da2725ef6aae624208e2ef828ed99892b17f60c344e57ed/multidict-6.7.1-cp313-cp313t-musllinux_1_2_ppc64le.whl", hash = "sha256:c9035dde0f916702850ef66460bc4239d89d08df4d02023a5926e7446724212c", size = 252341, upload-time = "2026-01-26T02:45:02.484Z" },
    { url = "https://files.pythonhosted.org/packages/e8/2c/2a1aa0280cf579d0f6eed8ee5211c4f1730bd7e06c636ba2ee6aafda302e/multidict-6.7.1-cp313-cp313t-musllinux_1_2_s390x.whl", hash = "sha256:af959b9beeb66c822380f222f0e0a1889331597e81f1ded7f374f3ecb0fd6c52", size = 246391, upload-time = "2026-01-26T02:45:03.862Z" },
    { url = "https://files.pythonhosted.org/packages/e5/03/7ca022ffc36c5a3f6e03b179a5ceb829be9da5783e6fe395f347c0794680/multidict-6.7.1-cp313-cp313t-musllinux_1_2_x86_64.whl", hash = "sha256:41f2952231456154ee479651491e94118229844dd7226541788be783be2b5108", size = 243422, upload-time = "2026-01-26T02:45:05.296Z" },
    { url = "https://files.pythonhosted.org/packages/dc/1d/b31650eab6c5778aceed46ba735bd97f7c7d2f54b319fa916c0f96e7805b/multidict-6.7.1-cp313-cp313t-win32.whl", hash = "sha256:df9f19c28adcb40b6aae30bbaa1478c389efd50c28d541d76760199fc1037c32", size = 47770, upload-time = "2026-01-26T02:45:06.754Z" },
    { url = "https://files.pythonhosted.org/packages/ac/5b/2d2d1d522e51285bd61b1e20df8f47ae1a9d80839db0b24ea783b3832832/multidict-6.7.1-cp313-cp313t-win_amd64.whl", hash = "sha256:d54ecf9f301853f2c5e802da559604b3e95bb7a3b01a9c295c6ee591b9882de8", size = 53109, upload-time = "2026-01-26T02:45:08.044Z" },
    { url = "https://files.pythonhosted.org/packages/3d/a3/cc409ba012c83ca024a308516703cf339bdc4b696195644a7215a5164a24/multidict-6.7.1-cp313-cp313t-win_arm64.whl", hash = "sha256:5a37ca18e360377cfda1d62f5f382ff41f2b8c4ccb329ed974cc2e1643440118", size = 45573, upload-time = "2026-01-26T02:45:09.349Z" },
    { url = "https://files.pythonhosted.org/packages/91/cc/db74228a8be41884a567e88a62fd589a913708fcf180d029898c17a9a371/multidict-6.7.1-cp314-cp314-macosx_10_15_universal2.whl", hash = "sha256:8f333ec9c5eb1b7105e3b84b53141e66ca05a19a605368c55450b6ba208cb9ee", size = 75190, upload-time = "2026-01-26T02:45:10.651Z" },
    { url = "https://files.pythonhosted.org/packages/d5/22/492f2246bb5b534abd44804292e81eeaf835388901f0c574bac4eeec73c5/multidict-6.7.1-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:a407f13c188f804c759fc6a9f88286a565c242a76b27626594c133b82883b5c2", size = 44486, upload-time = "2026-01-26T02:45:11.938Z" },
    { url = "https://files.pythonhosted.org/packages/f1/4f/733c48f270565d78b4544f2baddc2fb2a245e5a8640254b12c36ac7ac68e/multidict-6.7.1-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:0e161ddf326db5577c3a4cc2d8648f81456e8a20d40415541587a71620d7a7d1", size = 43219, upload-time = "2026-01-26T02:45:14.346Z" },
    { url = "https://files.pythonhosted.org/packages/24/bb/2c0c2287963f4259c85e8bcbba9182ced8d7fca65c780c38e99e61629d11/multidict-6.7.1-cp314-cp314-manylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:1e3a8bb24342a8201d178c3b4984c26ba81a577c80d4d525727427460a50c22d", size = 245132, upload-time = "2026-01-26T02:45:15.712Z" },
    { url = "https://files.pythonhosted.org/packages/a7/f9/44d4b3064c65079d2467888794dea218d1601898ac50222ab8a9a8094460/multidict-6.7.1-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:97231140a50f5d447d3164f994b86a0bed7cd016e2682f8650d6a9158e14fd31", size = 252420, upload-time = "2026-01-26T02:45:17.293Z" },
    { url = "https://files.pythonhosted.org/packages/8b/13/78f7275e73fa17b24c9a51b0bd9d73ba64bb32d0ed51b02a746eb876abe7/multidict-6.7.1-cp314-cp314-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:6b10359683bd8806a200fd2909e7c8ca3a7b24ec1d8132e483d58e791d881048", size = 233510, upload-time = "2026-01-26T02:45:19.356Z" },
    { url = "https://files.pythonhosted.org/packages/4b/25/8167187f62ae3cbd52da7893f58cb036b47ea3fb67138787c76800158982/multidict-6.7.1-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:283ddac99f7ac25a4acadbf004cb5ae34480bbeb063520f70ce397b281859362", size = 264094, upload-time = "2026-01-26T02:45:20.834Z" },
    { url = "https://files.pythonhosted.org/packages/a1/e7/69a3a83b7b030cf283fb06ce074a05a02322359783424d7edf0f15fe5022/multidict-6.7.1-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:538cec1e18c067d0e6103aa9a74f9e832904c957adc260e61cd9d8cf0c3b3d37", size = 260786, upload-time = "2026-01-26T02:45:22.818Z" },
    { url = "https://files.pythonhosted.org/packages/fe/3b/8ec5074bcfc450fe84273713b4b0a0dd47c0249358f5d82eb8104ffe2520/multidict-6.7.1-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:7eee46ccb30ff48a1e35bb818cc90846c6be2b68240e42a78599166722cea709", size = 248483, upload-time = "2026-01-26T02:45:24.368Z" },
    { url = "https://files.pythonhosted.org/packages/48/5a/d5a99e3acbca0e29c5d9cba8f92ceb15dce78bab963b308ae692981e3a5d/multidict-6.7.1-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:fa263a02f4f2dd2d11a7b1bb4362aa7cb1049f84a9235d31adf63f30143469a0", size = 248403, upload-time = "2026-01-26T02:45:25.982Z" },
    { url = "https://files.pythonhosted.org/packages/35/48/e58cd31f6c7d5102f2a4bf89f96b9cf7e00b6c6f3d04ecc44417c00a5a3c/multidict-6.7.1-cp314-cp314-musllinux_1_2_armv7l.whl", hash = "sha256:2e1425e2f99ec5bd36c15a01b690a1a2456209c5deed58f95469ffb46039ccbb", size = 240315, upload-time = "2026-01-26T02:45:27.487Z" },
    { url = "https://files.pythonhosted.org/packages/94/33/1cd210229559cb90b6786c30676bb0c58249ff42f942765f88793b41fdce/multidict-6.7.1-cp314-cp314-musllinux_1_2_i686.whl", hash = "sha256:497394b3239fc6f0e13a78a3e1b61296e72bf1c5f94b4c4eb80b265c37a131cd", size = 245528, upload-time = "2026-01-26T02:45:28.991Z" },
    { url = "https://files.pythonhosted.org/packages/64/f2/6e1107d226278c876c783056b7db43d800bb64c6131cec9c8dfb6903698e/multidict-6.7.1-cp314-cp314-musllinux_1_2_ppc64le.whl", hash = "sha256:233b398c29d3f1b9676b4b6f75c518a06fcb2ea0b925119fb2c1bc35c05e1601", size = 258784, upload-time = "2026-01-26T02:45:30.503Z" },
    { url = "https://files.pythonhosted.org/packages/4d/c1/11f664f14d525e4a1b5327a82d4de61a1db604ab34c6603bb3c2cc63ad34/multidict-6.7.1-cp314-cp314-musllinux_1_2_s390x.whl", hash = "sha256:93b1818e4a6e0930454f0f2af7dfce69307ca03cdcfb3739bf4d91241967b6c1", size = 251980, upload-time = "2026-01-26T02:45:32.603Z" },
    { url = "https://files.pythonhosted.org/packages/e1/9f/75a9ac888121d0c5bbd4ecf4eead45668b1766f6baabfb3b7f66a410e231/multidict-6.7.1-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:f33dc2a3abe9249ea5d8360f969ec7f4142e7ac45ee7014d8f8d5acddf178b7b", size = 243602, upload-time = "2026-01-26T02:45:34.043Z" },
    { url = "https://files.pythonhosted.org/packages/9a/e7/50bf7b004cc8525d80dbbbedfdc7aed3e4c323810890be4413e589074032/multidict-6.7.1-cp314-cp314-win32.whl", hash = "sha256:3ab8b9d8b75aef9df299595d5388b14530839f6422333357af1339443cff777d", size = 40930, upload-time = "2026-01-26T02:45:36.278Z" },
    { url = "https://files.pythonhosted.org/packages/e0/bf/52f25716bbe93745595800f36fb17b73711f14da59ed0bb2eba141bc9f0f/multidict-6.7.1-cp314-cp314-win_amd64.whl", hash = "sha256:5e01429a929600e7dab7b166062d9bb54a5eed752384c7384c968c2afab8f50f", size = 45074, upload-time = "2026-01-26T02:45:37.546Z" },
    { url = "https://files.pythonhosted.org/packages/97/ab/22803b03285fa3a525f48217963da3a65ae40f6a1b6f6cf2768879e208f9/multidict-6.7.1-cp314-cp314-win_arm64.whl", hash = "sha256:4885cb0e817aef5d00a2e8451d4665c1808378dc27c2705f1bf4ef8505c0d2e5", size = 42471, upload-time = "2026-01-26T02:45:38.889Z" },
    { url = "https://files.pythonhosted.org/packages/e0/6d/f9293baa6146ba9507e360ea0292b6422b016907c393e2f63fc40ab7b7b5/multidict-6.7.1-cp314-cp314t-m

```
acosx_10_15_universal2.whl", hash = "sha256:0458c978acd8e6ea53c81eefaddbbee9c6c5e591f41b3f5e8e194780fe026581", size = 82401, upload-time = "2026-01-26
T02:45:40.254Z" },
    { url = "https://files.pythonhosted.org/packages/7a/68/53b5494738d83558d87c3c71a486504d8373421c3e0dbb6d0db48ad42ee0/multidict-6.7.1-cp314-cp314t-m
acosx_10_15_x86_64.whl", hash = "sha256:c0abd12629b0af3cf590982c0b413b1e7395cd4ec026f30986818ab95bfaa94a", size = 48143, upload-time = "2026-01-26T02:
45:41.635Z" },
    { url = "https://files.pythonhosted.org/packages/37/e8/5284c53310dcdc99ce5d66563f6e5773531a9b9fe9ec7a615e9bc306b05f/multidict-6.7.1-cp314-cp314t-m
acosx_11_0_arm64.whl", hash = "sha256:14525a5f61d7d0c94b368a42cff4c9a4e7ba2d52e2672a7b23d84dc86fb02b0c", size = 46507, upload-time = "2026-01-26T02:45
:42.99Z" },
    { url = "https://files.pythonhosted.org/packages/e4/fc/6800d0e5b3875568b4083ecf5f310dcf91d86d52573160834fb4bfcf5e4f/multidict-6.7.1-cp314-cp314t-m
anylinux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:17307b22c217b4cf05033dabefe68255a534d637c6c9b0cc8382718f87be4262", size = 2
39358, upload-time = "2026-01-26T02:45:44.376Z" },
    { url = "https://files.pythonhosted.org/packages/41/75/4ad0973179361cdf3a113905e6e088173198349131be2b390f9fa4da5fc6/multidict-6.7.1-cp314-cp314t-m
anylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:7a7e590ff876a3eaf1c02a4dfe0724b6e69a9e9de6d8f556816f29c496046e
59", size = 246884, upload-time = "2026-01-26T02:45:47.167Z" },
    { url = "https://files.pythonhosted.org/packages/c3/9c/095bb28b5da139bd41fb9a5d5caff412584f377914bd8787c2aa98717130/multidict-6.7.1-cp314-cp314t-m
anylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:5fa6a95dfee63893d80a34758cd0e0c118a30b8dcb46372bf75106c591b77889"
, size = 225878, upload-time = "2026-01-26T02:45:48.698Z" },
    { url = "https://files.pythonhosted.org/packages/07/d0/c0a72000243756e8f5a277b6b514fa005f2c73d481b7d9e47cd4568aa2e4/multidict-6.7.1-cp314-cp314t-m
anylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:a0543217a6a017692aa6ae5cc39adb75e587af0f3a82288b1492eb73dd6cc2
a4", size = 253542, upload-time = "2026-01-26T02:45:50.164Z" },
    { url = "https://files.pythonhosted.org/packages/c0/6b/f69da15289e384ecf2a68837ec8b5ad8c33e973aa18b266f50fe55f24b8c/multidict-6.7.1-cp314-cp314t-m
anylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:f99fe611c312b3c1c0ace793f92464d8cd263cc3b26b5721950d977b006b6c4d", s
ize = 252403, upload-time = "2026-01-26T02:45:51.779Z" },
    { url = "https://files.pythonhosted.org/packages/a2/76/b9669547afa5a1a25cd93eaca91c0da1c095b06b6d2d8ec25b713588d3a1/multidict-6.7.1-cp314-cp314t-m
anylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9004d8386d133b7e6135679424c91b0b854d2d164af6ea3f289f8f2761064609"
, size = 244889, upload-time = "2026-01-26T02:45:53.27Z" },
    { url = "https://files.pythonhosted.org/packages/7e/a9/a50d2669e506dad33cfc45b5d574a205587b7b8a5f426f2fbb2e90882588/multidict-6.7.1-cp314-cp314t-m
usllinux_1_2_aarch64.whl", hash = "sha256:e628ef0e6859ffd8273c69412a2465c4be4a9517d07261b33334b5ec6f3c7489", size = 241982, upload-time = "2026-01-26T
02:45:54.919Z" },
    { url = "https://files.pythonhosted.org/packages/c5/bb/1609558ad8b456b4827d3c5a5b775c93b87878fd3117ed3db3423dfbce1b/multidict-6.7.1-cp314-cp314t-m
usllinux_1_2_armv7l.whl", hash = "sha256:841189848ba629c3552035a6a7f5bf3b02eb304e9fea7492ca220a8eda6b0e5c", size = 232415, upload-time = "2026-01-26T0
2:45:56.981Z" },
    { url = "https://files.pythonhosted.org/packages/d8/59/6f61039d2aa9261871e03ab9dc058a550d240f25859b05b67fd70f80d4b3/multidict-6.7.1-cp314-cp314t-m
usllinux_1_2_i686.whl", hash = "sha256:ce1bbd7d780bb5a0da032e095c951f7014d6b0a205f8318308140f1a6aba159e", size = 240337, upload-time = "2026-01-26T02:
45:58.698Z" },
    { url = "https://files.pythonhosted.org/packages/a1/29/fdc6a43c203890dc2ae9249971ecd0c41deaedfe00d25cb6564b2edd99eb/multidict-6.7.1-cp314-cp314t-m
usllinux_1_2_ppc64le.whl", hash = "sha256:b26684587228afed0d50cf804cc71062cc9c1cdf55051c4c6345d372947b268c", size = 248788, upload-time = "2026-01-26T
02:46:00.862Z" },
    { url = "https://files.pythonhosted.org/packages/a9/14/a153a06101323e4cf086ecee3faadba52ff71633d471f9685c42e3736163/multidict-6.7.1-cp314-cp314t-m
usllinux_1_2_s390x.whl", hash = "sha256:9f9af1130699433539829 3f9958071019e3ab95e9a707dc1383a35613f6abcb9", size = 242842, upload-time = "2026-01-26T02
:46:02.824Z" },
    { url = "https://files.pythonhosted.org/packages/41/5f/604ae839e64a4a6efc80db94465348d3b328ee955e37acb24badbcd24d83/multidict-6.7.1-cp314-cp314t-m
usllinux_1_2_x86_64.whl", hash = "sha256:b4938326284c4f1224178a560987b6cf8b4d38458b113d9b8c1db1a836e640a2", size = 240237, upload-time = "2026-01-26T0
2:46:05.898Z" },
    { url = "https://files.pythonhosted.org/packages/5f/60/c3a5187bf66f6fb546ff4ab8fb5a077cbdd832d7b1908d4365c7f74a1917/multidict-6.7.1-cp314-cp314t-w
in32.whl", hash = "sha256:98655c737850c064a65e006a3df7c997cd3b220be4ec8fe26215760b9697d4d7", size = 48008, upload-time = "2026-01-26T02:46:07.468Z" },
    { url = "https://files.pythonhosted.org/packages/0c/f7/addf1087b860ac60e6f382240f64fb99f8bfb532bb06f7c542b83c29ca61/multidict-6.7.1-cp314-cp314t-w
in_amd64.whl", hash = "sha256:497bde6223c212ba11d462853cfa4f0ae6ef97465033e7dc9940cdb3ab5b48e5", size = 53542, upload-time = "2026-01-26T02:46:08.809Z
" },
    { url = "https://files.pythonhosted.org/packages/4c/81/4629d0aa32302ef7b2ec65c75a728cc5ff4fa410c50096174c1632e70b3e/multidict-6.7.1-cp314-cp314t-w
in_arm64.whl", hash = "sha256:2bbd113e0d4af5db41d5ebfe9ccaff89de2120578164f86a5d17d5a576d1e5b2", size = 44719, upload-time = "2026-01-26T02:46:11.146Z
" },
    { url = "https://files.pythonhosted.org/packages/81/08/7036c080d7117f28a4af526d794aab6a84463126db031b007717c1a6676e/multidict-6.7.1-py3-none-any.w
hl", hash = "sha256:55d97cc6dae627efa6a6e548885712d4864b81110ac76fa4e534c03819fa4a56", size = 12319, upload-time = "2026-01-26T02:46:44.004Z" },
]

[[package]]
name = "mypy"
version = "2.3.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "ast-serialize" },
    { name = "librt", marker = "platform_python_implementation != 'PyPy'" },
    { name = "mypy-extensions" },
    { name = "pathspec" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/12/af/4e516a05d3ca2eb9283e9ec45b2c02225c1514dd6da49fd3c9eaa6639370/mypy-2.3.0.tar.gz", hash =
 "sha256:465965d41cd9a2726694e983e8ce7113259327bec798115d1e1dfa2a52fb666e", size = 3988104, upload-time = "2026-07-13T11:34:53.387Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/dc/94/0e7e592619e2133596a47cdd642534b0456545c218430bd3b9d8fefdd1b1/mypy-2.3.0-cp312-cp312-macosx_
10_13_x86_64.whl", hash = "sha256:2d53fc67b9d28a43c6199077f49fea0f05839e36cf6158500331c9549225e5a5", size = 15026523, upload-time = "2026-07-13T11:34:
49.206Z" },
    { url = "https://files.pythonhosted.org/packages/f6/d2/1e1731df090a857df2807177a4626863e5ac0f0256513c35780efe53986f/mypy-2.3.0-cp312-cp312-macosx_
11_0_arm64.whl", hash = "sha256:fbc00cee7bdbb9291979ddc9d08034a29dfcda4932628c9bbc28c1edd589df0c", size = 14032189, upload-time = "2026-07-13T11:33:57
.168Z" },
    { url = "https://files.pythonhosted.org/packages/44/95/cab921f4a806e171f34113e6181dd23c55358ccf6a80741269ef594a410e/mypy-2.3.0-cp312-cp312-manylin
ux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:04e617030eca5221909c8b7d8d7fd1c637948199aa2100b2ad9813feb07e1491", s
ize = 14198696, upload-time = "2026-07-13T11:32:12.767Z" },
    { url = "https://files.pythonhosted.org/packages/66/80/e6d008bb19fe446e3662d85e0e2717bf9f2d611a2164fb29d6e067dbf46c/mypy-2.3.0-cp312-cp312-manylin
ux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:56c184d2c20ca6b6378d58d1960270a767f41f5e44acbbd27f05effef4f4e1d7", size
 = 15286904, upload-time = "2026-07-13T11:34:27.594Z" },
    { url = "https://files.pythonhosted.org/packages/db/83/94397c9293608a364aa03e8084fb34ede4ae976a260384b9b52929308135/mypy-2.3.0-cp312-cp312-musllin
ux_1_2_x86_64.whl", hash = "sha256:3961a4a34b05f7c74b0f05aa51fbfe99a2d1e126038df40318d15c8f558b7ef3", size = 15528342, upload-time = "2026-07-13T11:34
:07.819Z" },
    { url = "https://files.pythonhosted.org/packages/cf/96/d8b37d819adec6cfccfb1fd3afc1735d94717ddeafb45536db9c6943e09b/mypy-2.3.0-cp312-cp312-win_amd
64.whl", hash = "sha256:b1942b9314d4c784b8ea1dbab497260329 0e5dd5630f06675f13aec97526bc4c", size = 11218346, upload-time = "2026-07-13T11:28:27.745Z" }
,
    { url = "https://files.pythonhosted.org/packages/2b/cd/cd9f725b19b19e5b530a154cf9bcf9e94279c5d55b3c34fb42b3aa48ea1b/mypy-2.3.0-cp312-cp312-win_arm
64.whl", hash = "sha256:be51653d7669d7d7955d613b8d0bb57d5b652eaf71a873ddf65ac87254dd2595", size = 10204525, upload-time = "2026-07-13T11:31:02.552Z" }
,
    { url = "https://files.pythonhosted.org/packages/6e/ae/f7d056eb0294586a572d0d0d89580ec633c064db520f11d37d5a2fb833bd/mypy-2.3.0-cp313-cp313-macosx_
10_13_x86_64.whl", hash = "sha256:91ad22a52ae2c7e621c2f67c94d5a17f66b3209a4cff5cf8a573579835c69e97", size = 14947298, upload-time = "2026-07-13T11:27:
47.734Z" },
    { url = "https://files.pythonhosted.org/packages/32/d5/db3e7af01e7844d21662c6ddc1f7825ec7cb4053f0391ac02faf3638396f/mypy-2.3.0-cp313-cp313-macosx_
11_0_arm64.whl", hash = "sha256:99ac767cc5d3b64c8d0ae226ead10c96694f94e4e7da1668642225dcd4e75aac", size = 13950768, upload-time = "2026-07-13T11:27:57
.726Z" },
    { url = "https://files.pythonhosted.org/packages/d9/fb/43c031f0190513d1ec248ed037eceb742ddd2a4d74bbf406658a28173837/mypy-2.3.0-cp313-cp313-manylin
ux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:de6d2c484742a4d7b0ed6d07b143375624d3b899c5749c7b3c947f56261f48a6", s
ize = 14151586, upload-time = "2026-07-13T11:29:18.615Z" },
    { url = "https://files.pythonhosted.org/packages/ec/c3/f8b2ffc60883084da91be51af58e88a7ffd4ff9795acb7d902ff88d31eb1/mypy-2.3.0-cp313-cp313-manylin
ux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:7da939dd335cfd2ad788bdfd081c9f4e47634ab995e5a45eb15fd1e5bc052f8b", size
 = 15227411, upload-time = "2026-07-13T11:30:29.904Z" },
    { url = "https://files.pythonhosted.org/packages/83/2e/16b917fc7adcf03f1aadddfc93aab804ffb234b1ab09c0ffd6d92a5d34a2/mypy-2.3.0-cp313-cp313-musllin
ux_1_2_x86_64.whl", hash = "sha256:7247eb2824f996722a949530183394921ca71deb9680052a338cf53cff7925c2", size = 15478790, upload-time = "2026-07-13T11:33
```

:14.686Z" },
    { url = "https://files.pythonhosted.org/packages/c0/88/aaa65a93c73d0cdae7e42f8adb302bf6885bb281302084f99d0290a35347/mypy-2.3.0-cp313-cp313-win_amd64.whl", hash = "sha256:75b0984bb3cbd76bb5c9291a8671f7ae66ca3b51c7584c358fc2e923259f0757", size = 11234919, upload-time = "2026-07-13T11:33:39.28Z" },
    { url = "https://files.pythonhosted.org/packages/35/19/b40de63f1a80e63bc2d40f0679a6a8dbd34e95176c8122119bdf406aa552/mypy-2.3.0-cp313-cp313-win_arm64.whl", hash = "sha256:d78fcf900b59cb7e82cb7e3a235e31b462d9333d92285bd1e4952d355b8ffba1", size = 10201510, upload-time = "2026-07-13T11:31:52.619Z" },
    { url = "https://files.pythonhosted.org/packages/a4/58/fa0ae047da911f540284009b4f44b96fe09d83c076d7c103e9d645f46303/mypy-2.3.0-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:ea317b060ce83e26050f8f9e4d7d6bf44ed7597c8ff9990bccffbb9d1d8522db", size = 14941909, upload-time = "2026-07-13T11:32:34.332Z" },
    { url = "https://files.pythonhosted.org/packages/15/14/2ba1d61452d7c2a7fe12741e8d374e52b183476b07aa7f9e2a0d02b0720a/mypy-2.3.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:094af99f92638aa92852326188b85a89e50f4a472f44827c03362228482f0762", size = 13967581, upload-time = "2026-07-13T11:30:00.587Z" },
    { url = "https://files.pythonhosted.org/packages/ed/5a/483fb9e5ffbbb1a28dccc7b0a13d141b17ac769b6c9f488c0a0c63698962/mypy-2.3.0-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:de121747278144fc9ae7caa2e978cf5df12aebc82933182f5b3b86081a30baef", size = 14168807, upload-time = "2026-07-13T11:28:48.6Z" },
    { url = "https://files.pythonhosted.org/packages/ae/77/70d7a10732063beb74ad713682cf871e88f5c5fa39bfc8beff8a524bf9cb/mypy-2.3.0-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:37fa4de896a84e2dc9200d91e614c22563b43d1a266789d4bbac7b22ebe6192b", size = 15200144, upload-time = "2026-07-13T11:31:25.283Z" },
    { url = "https://files.pythonhosted.org/packages/56/72/766218ac783be4fdfcd699b90037b63017348a3e86fb2c1fbfb18302637d/mypy-2.3.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:f1b3a98dfd21058bc759bb3337d5d1f61d0fdf9f3cf9c00f4291790fb5427bff", size = 15460389, upload-time = "2026-07-13T11:29:29.077Z" },
    { url = "https://files.pythonhosted.org/packages/38/4e/8a9db7411ecb8ec0cb1fd05dba432f28bafffcd38b4e887714a4a0506689/mypy-2.3.0-cp314-cp314-pyemscripten_2026_0_wasm32.whl", hash = "sha256:944c665d984157cb96a679dfb7a4a81dd1d36b24b9c284b699514e6e626b82d4", size = 7753664, upload-time = "2026-07-13T11:29:08.147Z" },
    { url = "https://files.pythonhosted.org/packages/65/4c/c3f8bfd6ed0e5e38b5a244403b27f821d433443df5a15a278417c10a3a3c/mypy-2.3.0-cp314-cp314-win_amd64.whl", hash = "sha256:4359424140d985192c778c1ce2c114a10c1ca58a381ed79cfa70d37df94b299f", size = 11417237, upload-time = "2026-07-13T11:33:47.467Z" },
    { url = "https://files.pythonhosted.org/packages/3c/00/89a32eaf5ccf174bc4f90db0eaea5d70636c01b8d49f384bdab2e8834390/mypy-2.3.0-cp314-cp314-win_arm64.whl", hash = "sha256:3dd0bed92c4bdec57c42505b96416fb9e6a5aa7be84d2809bcd5f2ecec2860d7", size = 10389252, upload-time = "2026-07-13T11:31:43.81Z" },
    { url = "https://files.pythonhosted.org/packages/31/56/104f93d69aa9f339b6b9d3b0a7faa699b8b466c942cf3ae86cc2a2ec0915/mypy-2.3.0-cp314-cp314t-macosx_10_15_x86_64.whl", hash = "sha256:691fdc37132b1ae628d834f672e74de83462d9fb4aff621835767fb43a8dd373", size = 16385495, upload-time = "2026-07-13T11:29:49.818Z" },
    { url = "https://files.pythonhosted.org/packages/d2/03/f1d2123313f55efafdd27706960f43a771c62f1b68426c76043f3ab9ebf3/mypy-2.3.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:aec15d465d477558fd842757b487849007311cf3897849cdda0e3162ac0ac556", size = 15098155, upload-time = "2026-07-13T11:30:40.301Z" },
    { url = "https://files.pythonhosted.org/packages/e5/5d/d5f9200399b445e81726c4f23becee33f233aee81c72680b1ef3a258b641/mypy-2.3.0-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:b352b7e49f5e6576009e8df730e1ff4f915cb565b851b396d2ffe2f5a6f5da88", size = 15514155, upload-time = "2026-07-13T11:34:38.569Z" },
    { url = "https://files.pythonhosted.org/packages/cd/ce/69977c555f08faa3190cfde44189b89dbd56861b1ab97aa18fc5f3a2e4a3/mypy-2.3.0-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:1c6c6bf687b17f90dbfcad95b960d32eaa0154c00da45f03ab50bf8952e047fe", size = 16766351, upload-time = "2026-07-13T11:33:29.195Z" },
    { url = "https://files.pythonhosted.org/packages/bc/92/6648b6caa3ab9e00f9ac0c2a78307805f873dd48139b24a6f6f7c3667bbf/mypy-2.3.0-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:f4ed18f111bfe2d599bca7468e7f9251042c1c2118f762c8de2766a56d773c60", size = 17043490, upload-time = "2026-07-13T11:30:53.927Z" },
    { url = "https://files.pythonhosted.org/packages/7c/ab/0dc91d80f3f016634c68d451f294a97320fe903a9b6f90b9e57b3f7f1717/mypy-2.3.0-cp314-cp314t-win_amd64.whl", hash = "sha256:0b025a93cffb9781d231f232be07a17912f35f10a313c24f301c81e842870654", size = 12146869, upload-time = "2026-07-13T11:29:38.874Z" },
    { url = "https://files.pythonhosted.org/packages/85/b5/4c964d02634ba81f4d1c84838e5c5b18ab06d13ed568960f5d6318495ccc/mypy-2.3.0-cp314-cp314t-win_arm64.whl", hash = "sha256:adebc76aab4f3495a88b41d48aa4aff0c03f2822501da76625afcca5975f19e5", size = 10965113, upload-time = "2026-07-13T11:28:07.056Z" },
    { url = "https://files.pythonhosted.org/packages/2c/fa/fdc54fe583ba3cafbcedfb70eeeaf03849f75b1827a07096c7bd996f582d/mypy-2.3.0-py3-none-any.whl", hash = "sha256:6b1cdb579446b60432432b2b2403a6201b4b475a004d7f488511c9ba177c9e88", size = 2753292, upload-time = "2026-07-13T11:33:18.48Z" },
]

[[package]]
name = "mypy-extensions"
version = "1.1.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a2/6e/371856a3fb9d31ca8dac321cda606860fa4548858c0cc45d9d1d4ca2628b/mypy_extensions-1.1.0.tar.gz", hash = "sha256:52e68efc3284861e772bbcd66823fde5ae21fd2fdb51c62a211403730b916558", size = 6343, upload-time = "2025-04-22T14:54:24.164Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/79/7b/2c79738432f5c924bef5071f933bcc9efd0473bac3b4aa584a6f7c1c8df8/mypy_extensions-1.1.0-py3-none-any.whl", hash = "sha256:1be4cccdb0f2482337c4743e60421de3a356cd97508abadd57d47403e94f5505", size = 4963, upload-time = "2025-04-22T14:54:22.983Z" },
]

[[package]]
name = "nest-asyncio"
version = "1.6.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/83/f8/51569ac65d696c8ecbee95938f89d4abf00f47d58d48f6fbabfe8f0baefe/nest_asyncio-1.6.0.tar.gz", hash = "sha256:6f172d5449aca15afd6c646851f4e31e02c598d553a667e38cafa997cfec55fe", size = 7418, upload-time = "2024-01-21T14:25:19.227Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a0/c4/c2971a3ba4c6103a3d10c4b0f24f461ddc027f0f09763220cf35ca1401b3/nest_asyncio-1.6.0-py3-none-any.whl", hash = "sha256:87af6efd6b5e897c81050477ef65c62e2b2f35d51703cae01aff2905b1852e1c", size = 5195, upload-time = "2024-01-21T14:25:17.223Z" },
]

[[package]]
name = "networkx"
version = "3.6.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/6a/51/63fe664f3908c97be9d2e4f1158eb633317598cfa6e1fc14af5383f17512/networkx-3.6.1.tar.gz", hash = "sha256:26b7c357accc0c8cde558ad486283728b65b6a95d85ee1cd66bafab4c8168509", size = 2517025, upload-time = "2025-12-08T17:02:39.908Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/9e/c9/b2622292ea83fbb4ec318f5b9ab867d0a28ab43c5717bb85b0a5f6b3b0a4/networkx-3.6.1-py3-none-any.whl", hash = "sha256:d47fbf302e7d9cbbb9e2555a0d267983d2aa476bac30e90dfbe5669bd57f3762", size = 2068504, upload-time = "2025-12-08T17:02:38.159Z" },
]

[[package]]
name = "nltk"
version = "3.10.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "click" },
    { name = "defusedxml" },
    { name = "joblib" },
    { name = "regex" },
    { name = "tqdm" },
]
sdist = { url = "https://files.pythonhosted.org/packages/72/16/24d639531e73cbc6884fb251d116dfe469df9c595e0dcf24668c54d0e8d3/nltk-3.10.2.tar.gz", hash = "sha256:fcfd80fb77931868cea8357573c79838b8abc609942ef9914d1c9f6070d4645c", size = 3101716, upload-time = "2026-08-05T09:56:20.657Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/1f/2b/bf677eb32ca6684b270c0d19ab133c2271e9f3375997e5a8dd2b08e3152d/nltk-3.10.2-py3-none-any.whl", hash = "sha256:2c7ccacb765c5e26b0cb60fb1b57080af522c6924d12a714a243305ba3637412", size = 1725815, upload-time = "2026-08-05T09:56:09.657Z" },
]

```
[[package]]
name = "nodeenv"
version = "1.10.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/24/bf/d1bda4f6168e0b2e9e5958945e01910052158313224ada5ce1fb2e1113b8/nodeenv-1.10.0.tar.gz", hash = "sha256:996c191ad80897d076bdfba80a41994c2b47c68e224c542b48feba42ba00f8bb", size = 55611, upload-time = "2025-12-20T14:08:54.006Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/88/b2/d0896bdcdc8d28a7fc5717c305f1a861c26e18c05047949fb371034d98bd/nodeenv-1.10.0-py2.py3-none-any.whl", hash = "sha256:5bb13e3eed293615535339b3c620e76779af4cb4c6a90deccc9e36b274d3827", size = 23438, upload-time = "2025-12-20T14:08:52.782Z" },
]

[[package]]
name = "numpy"
version = "2.5.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/22/fd/89965aa4ac08c74998539fcbf24fa3540f3e15237fbeb6bcf9c908f4aade/numpy-2.5.1.tar.gz", hash = "sha256:a48a113e6afea91f5608793bafa7ef2ad481fefbda87ec5069f483de61cb9fa3", size = 20755553, upload-time = "2026-07-04T17:08:00.933Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/62/7b/14687aa674250e5e546f616f486b0d56d3631cd5b2415739141ce40bdcea/numpy-2.5.1-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:2c889b56fe48b1018f764b0eec8df59ab654e9148aa91faa12596043500de277", size = 16801574, upload-time = "2026-07-04T17:06:12.423Z" },
    { url = "https://files.pythonhosted.org/packages/e1/19/cc5bb2a3f2913d27d6dbb2c78d25921fabaedc6741d4a5a615a11f3c5bf3/numpy-2.5.1-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:ab451b59c5643c570974c43aef780703ef1d3b4965d2be07afd530615a9358d1", size = 11772250, upload-time = "2026-07-04T17:06:15.726Z" },
    { url = "https://files.pythonhosted.org/packages/42/77/fdf34a71dd30f54979b18603bee915e0aaf825b07afe79acd60b04b691e2/numpy-2.5.1-cp312-cp312-macosx_14_0_arm64.whl", hash = "sha256:78798bd5b9ad744056af8efa90e3b9ddaa53272a0848a483084a1cc0a13b2dc0", size = 5331516, upload-time = "2026-07-04T17:06:17.913Z" },
    { url = "https://files.pythonhosted.org/packages/ce/e2/eb7efa015b4cce41e2517bf182a7fce0d7d5b9d9ed76a29bfa0f4fe4505c/numpy-2.5.1-cp312-cp312-macosx_14_0_x86_64.whl", hash = "sha256:2ae0ca40bcb22d6ba59c1dfd5446f49940b0f2d821fde133f10dda11f816b84e", size = 6664863, upload-time = "2026-07-04T17:06:20.02Z" },
    { url = "https://files.pythonhosted.org/packages/a9/4b/a2b32dd94ee9ffbeecb28152240042a3949db33b1c834d44090b80e1b3b8/numpy-2.5.1-cp312-cp312-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:61ac47e772e6b8ea489e1d2f441a34c5c3ac17327e7ce294cbdf535795ad4e75", size = 15167977, uploaded-time = "2026-07-04T17:06:21.621Z" },
    { url = "https://files.pythonhosted.org/packages/b8/a9/6e73d68500f80773f65f0654ea932019d6694329a0eb0ed0533de38df376/numpy-2.5.1-cp312-cp312-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:59fda5e192b570217ec2580c96f00e9a7e12ef6866a900eb089b62c1a32545ca", size = 16672469, upload-time = "2026-07-04T17:06:24.064Z" },
    { url = "https://files.pythonhosted.org/packages/24/7d/ad3e59015135f5261c95fd4cafeff159c955febd83a99a1d9250c4233815/numpy-2.5.1-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:f7119ebff1a9829e9f431a4f9d28e703023bb6b9fe7c8f724467dbfc27c94ab3", size = 16527531, upload-time = "2026-07-04T17:06:26.69Z" },
    { url = "https://files.pythonhosted.org/packages/83/d0/a39b2fbcde9cb17a1dac678f254b33a6336298af9df338824c685425d5e8/numpy-2.5.1-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:e824c2acf8862052246be5a44c15da1777940c60d010dd2aab897824d9c430f9", size = 18431940, upload-time = "2026-07-04T17:06:29.521Z" },
    { url = "https://files.pythonhosted.org/packages/04/12/cff070947791c1ed425ff76413189adbdc2fbe215eba7ce7fa454a03c7f8/numpy-2.5.1-cp312-cp312-win32.whl", hash = "sha256:08d60c810432eb83360958dea0999ac4cfb94531ea8efcbf0b7f277c2068aeb2", size = 6066764, upload-time = "2026-07-04T17:06:32.571Z" },
    { url = "https://files.pythonhosted.org/packages/65/66/53f31807a48a750f9d748da273bc3fcedd12b27ff1f3e373bfec55ef2dc0/numpy-2.5.1-cp312-cp312-win_amd64.whl", hash = "sha256:f7d60026c0bdb1380e83bfa7a0419c4577ee4b9a08880afcb6dadeb74c649fa2", size = 12430966, upload-time = "2026-07-04T17:06:34.926Z" },
    { url = "https://files.pythonhosted.org/packages/2b/2a/d1a88066b1c14186f5d3c0d18c94f17b064511982bab0578d49ee9d43c29/numpy-2.5.1-cp312-cp312-win_arm64.whl", hash = "sha256:17a25e09640602e10bc8de0e6fa2b3fd68eedd84ba6d7842dc8f32f9ab87bd0b", size = 10350488, upload-time = "2026-07-04T17:06:37.785Z" },
    { url = "https://files.pythonhosted.org/packages/eb/07/ec2a3f0c91761581d4b7104a740791800025983f9a4dc4e73f91a99aeac4/numpy-2.5.1-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:0bfebd8695f9863592fe744be833a258120b14a9f39da255e8aa8fade2c0ddd1", size = 16796419, upload-time = "2026-07-04T17:06:40.37Z" },
    { url = "https://files.pythonhosted.org/packages/ab/ab/ddb499fc4f8780354395face5b65c7fd107bcd6e1d667a5f07d046956f6f/numpy-2.5.1-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:30b44a6b53a7ae63c54c089a8726e5563ed302716c5b7ccc85afade40b0e7ff6", size = 11765832, upload-time = "2026-07-04T17:06:42.768Z" },
    { url = "https://files.pythonhosted.org/packages/88/b3/3c28c558a09fc72100c646dac6d2fce8e834c471b0edca01a29996706117/numpy-2.5.1-cp313-cp313-macosx_14_0_arm64.whl", hash = "sha256:6165343f81b56ef8f514f396989e529b61d9dc709b99421b07e9f3e698e2287d", size = 5325143, upload-time = "2026-07-04T17:06:45.466Z" },
    { url = "https://files.pythonhosted.org/packages/5e/0e/ce19b985bb15c596f4f05954e76cccc77c845083b3b8f938a6c68e523128/numpy-2.5.1-cp313-cp313-macosx_14_0_x86_64.whl", hash = "sha256:4939237038ada79308dda3204ac6462df056b5672b2e25db1149cf873668b3e1", size = 6659749, upload-time = "2026-07-04T17:06:47.288Z" },
    { url = "https://files.pythonhosted.org/packages/2e/20/1ee6614d64332a1bba6411f38e68cb79eec1b2459e20a623777c5c5492a2/numpy-2.5.1-cp313-cp313-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:1c6759f538fb912fc46de0a6b1758ccf7b57bc7c7ebebc23974fdac3de8db0cd", size = 15164716, uploaded-time = "2026-07-04T17:06:49.494Z" },
    { url = "https://files.pythonhosted.org/packages/ed/a7/2bcd3fdbb87804755c35b729bf8709d62025c5f4cfd7d5b2415997097515/numpy-2.5.1-cp313-cp313-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9726558e8db4a5bf7929a70ae50f63abda4daf0efe810e3bfbab95976f75fc1a", size = 16661440, upload-time = "2026-07-04T17:06:52.061Z" },
    { url = "https://files.pythonhosted.org/packages/fc/d7/a41e3310c886fe457d36e670bbf24fae411aca8a7b6ad92a32afd924077c/numpy-2.5.1-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:3935f3b419b244a02732676fa5317a9193cc596a4c0646db07e5b421229ac9f7", size = 16526305, upload-time = "2026-07-04T17:06:54.605Z" },
    { url = "https://files.pythonhosted.org/packages/53/75/4333a9a707c1edd3a4e1a0c58eca52c0f31e55089fa80db02b5565b24df7/numpy-2.5.1-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:dc932a65ded7ce9013d120845a2514dcccb1a67bfc8deb8d37633762951904a6", size = 18423008, upload-time = "2026-07-04T17:06:57.54Z" },
    { url = "https://files.pythonhosted.org/packages/ee/90/e314a32b1c11a2ffe818ddad3a57b50b4b6e1b6c487192eb50cdef0415d0/numpy-2.5.1-cp313-cp313-win32.whl", hash = "sha256:4b4ff1608417eb7a59da7b967bbb798cacfe071d2caf526a24281cd562072ed9", size = 6063885, upload-time = "2026-07-04T17:07:00.14Z" },
    { url = "https://files.pythonhosted.org/packages/10/70/800b3fca480af32df9e8ea9f3d4a0c8feb4b32d7f195d174eabbda4829ad/numpy-2.5.1-cp313-cp313-win_amd64.whl", hash = "sha256:6c3fe51bc6a16453d452997053454f309e8e0ed7b42d6b361ce4ac8c32913d74", size = 12425674, upload-time = "2026-07-04T17:07:02.387Z" },
    { url = "https://files.pythonhosted.org/packages/8b/0b/196350c122f50f6ca56846f2d71efd5e0d24b7b2e07355e019b2e2c7a11e/numpy-2.5.1-cp313-cp313-win_arm64.whl", hash = "sha256:f7feb014281029e628ba2d5a007407443b06e418b6fe451d1e2adcbc8eba0107", size = 10350256, upload-time = "2026-07-04T17:07:04.878Z" },
    { url = "https://files.pythonhosted.org/packages/db/f4/731b6085a83faf6ca843394cbd5e217280c214399f7e8b21b9f552af0ae2/numpy-2.5.1-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:7c786fe9a5bbe360022e584c5a34cf6b54265c71bd7ec8ac3d8fec38968071f8", size = 16795063, upload-time = "2026-07-04T17:07:07.374Z" },
    { url = "https://files.pythonhosted.org/packages/bf/64/0e215f2048dd11a55bb989ed41b3585ef57452404e638d703a211a3e4157/numpy-2.5.1-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:32985c896d897419ef8da6917872d80b78ad0ea26d85b23245c7366ffde76d75", size = 11776652, upload-time = "2026-07-04T17:07:09.907Z" },
    { url = "https://files.pythonhosted.org/packages/b5/59/2b844c7a6e9deff69b404a66221e1542937734f65d5e6e39411876053862/numpy-2.5.1-cp314-cp314-macosx_14_0_arm64.whl", hash = "sha256:efd736408cc97c79b9e6917338dfc8f06013b2274f992e96b1d9a81a71e2a2c2", size = 5335944, upload-time = "2026-07-04T17:07:12.227Z" },
    { url = "https://files.pythonhosted.org/packages/86/51/9bf7cb2cabcebc9e017e4ec7e6322b378317a542c08b4cb68479c1efc716/numpy-2.5.1-cp314-cp314-macosx_14_0_x86_64.whl", hash = "sha256:ab84dc6b074fa881cae55bea94cc4f68e285181ba7f32497bf7dee6b1496165b", size = 6656266, upload-time = "2026-07-04T17:07:14.368Z" },
    { url = "https://files.pythonhosted.org/packages/83/3e/fb7615b211b82a32f44d5180a6d421b61f84d4fadd578b48ba4ac34e189f/numpy-2.5.1-cp314-cp314-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:caf3e317d33d60c37986b452613f4ab51246d0691350c03d0cb4a898627f4a95", size = 15179720, uploaded-time = "2026-07-04T17:07:16.272Z" },
    { url = "https://files.pythonhosted.org/packages/41/5f/0f992cb24560673496c5d68de61913b57166ce530ffda07c1f280e0cc464/numpy-2.5.1-cp314-cp314-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:54ad769f17bc2d833b620851989f62054fb9ab93c969d9e1dc3c8e3d56beea21", size = 16664835, upload-time = "2026-07-04T17:07:19.021Z" },
    { url = "https://files.pythonhosted.org/packages/a2/2f/97d6475ee91afe2587797d09446f9d3e475ad4cb681662d824809327b75a/numpy-2.5.1-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:c12afb53450fa976d4c681c50a7423729a4c51c0465ed9f32b8a9cabbc472373", size = 16539135, upload-time = "2026-07-04T17:07:22.015Z" },
    { url = "https://files.pythonhosted.org/packages/c4/5b/4db81e4ba0be7e2776b1de68c82aa862c7f8ec27e1b4927d4ae075e20678/numpy-2.5.1-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:e8c11c405efc5ff6816d5983c96cdfa215bab3428961243af3ff59b228490438", size = 18426684, upload-time = "2026-07-04T17:0
```

```
7:24.941Z" },
    { url = "https://files.pythonhosted.org/packages/1f/64/c0ba2d90724d450279a7df8f32057241070250a26a7e2b5337d77347f481/numpy-2.5.1-cp314-cp314-win32.
whl", hash = "sha256:f2479a47f8d5932d1718168a681ad6e536a9df484c83cfcf9de365e164537ace", size = 6116103, upload-time = "2026-07-04T17:07:27.622Z" },
    { url = "https://files.pythonhosted.org/packages/c1/1a/837f9ed7405adcd7a40538792eb169eddd8fa5630c16a1ef49dae71a30f4/numpy-2.5.1-cp314-cp314-win_am
d64.whl", hash = "sha256:24d0eb82c0541d3415a33425db64ae439dffccd7b4dbcb30e7c35120205c506a", size = 12562177, upload-time = "2026-07-04T17:07:29.887Z"
},
    { url = "https://files.pythonhosted.org/packages/22/ed/49707938b6dd0a78a9178dd93227dc89e4c11af47f5c798d70366e8d0483/numpy-2.5.1-cp314-cp314-win_ar
m64.whl", hash = "sha256:5a4c988b38d261deeeaad9954e3deb091ad905c94e8bb6708654ef1d97f286b0", size = 10627739, upload-time = "2026-07-04T17:07:32.568Z"
},
    { url = "https://files.pythonhosted.org/packages/a6/c7/bb4b882cfe7f299cbc8b66e42e7dd78cf9d14e40f9469fc5e3db7e15b3bd/numpy-2.5.1-cp314-cp314t-macos
x_11_0_arm64.whl", hash = "sha256:a33276be12fa045805f477f22482088b66bb758ffbe89a9d21457de863a32e22", size = 11894709, upload-time = "2026-07-04T17:07:
34.941Z" },
    { url = "https://files.pythonhosted.org/packages/40/3f/5af7f4a7f6224aef48017aa82bb6174c7a659d724be0c75017b7e64a55b4/numpy-2.5.1-cp314-cp314t-macos
x_14_0_arm64.whl", hash = "sha256:f089d7b00756190aacf1f5d34bdf38c3c430ac82b4f868f8cede73380460fce7", size = 5453810, upload-time = "2026-07-04T17:07:3
7.495Z" },
    { url = "https://files.pythonhosted.org/packages/20/c9/3474309bc94d634d3f9c3eddf03250ecb8c22cd948ef16fef69a77cc5d7b/numpy-2.5.1-cp314-cp314t-macos
x_14_0_x86_64.whl", hash = "sha256:09e9bfd8d2cf479c7d174804fb3811c53a8e9f20a37444008606b57d6b7a826d", size = 6761189, upload-time = "2026-07-04T17:07:
39.563Z" },
    { url = "https://files.pythonhosted.org/packages/90/8a/558ae39fdd55d7e7f7fef9a84a6e964ac6b23edbd2a07e52bb084500507d/numpy-2.5.1-cp314-cp314t-manyl
inux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:e68d8dd1e7eba712948f2053a29ec86917bc70ba1358df869d9f06649ef9cf09", size = 15225039, uplo
ad-time = "2026-07-04T17:07:41.682Z" },
    { url = "https://files.pythonhosted.org/packages/63/27/ca7392b2d030277bdf0273e7d23255b3ee57d57a7c170a6f4fb3981e1e5d/numpy-2.5.1-cp314-cp314t-manyl
inux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:99d5095fa265a0c4152e7bb12759e14381ef5496152f1ce58f44bdf55c44beb4", size = 16701306, upload
-time = "2026-07-04T17:07:44.611Z" },
    { url = "https://files.pythonhosted.org/packages/02/42/03d53ae7996c44d4374a8262e9dc41671fd56cbb98f7d47ef85cf5da4c6b/numpy-2.5.1-cp314-cp314t-musll
inux_1_2_aarch64.whl", hash = "sha256:ab87a91b3cc3382b8956095bd8f95e00cf679bb81554339be1a2ba404a1473c1", size = 16589955, upload-time = "2026-07-04T17
:07:47.694Z" },
    { url = "https://files.pythonhosted.org/packages/7b/15/6c1784ae469640e65db111e9a34b3d0f14d91e8a38b9ce34810ced370dbb/numpy-2.5.1-cp314-cp314t-musll
inux_1_2_x86_64.whl", hash = "sha256:224ca51130ef7da85bea2191625181cb4f337f9cb64b471f10c1a12aa8b60077", size = 18464252, upload-time = "2026-07-04T17:
07:50.684Z" },
    { url = "https://files.pythonhosted.org/packages/94/a8/f98e50356cf167df656c526c2dfeec2d7dde182f2a3da4b458a5938e2776/numpy-2.5.1-cp314-cp314t-win32
.whl", hash = "sha256:6eab239876581b2b3c5a242281b6007bbdbcd1c7085d7709bb57c5929b11e6bf", size = 6263298, upload-time = "2026-07-04T17:07:53.445Z" },
    { url = "https://files.pythonhosted.org/packages/72/ac/96ae880cdecad0b3275d9359fcec72667b49a4863c9f12942e43679dda02/numpy-2.5.1-cp314-cp314t-win_a
md64.whl", hash = "sha256:83ce9c80d5b521b0d77ddcbe5447c218d247929b6cc056ca5351342accfff0af", size = 12748623, upload-time = "2026-07-04T17:07:55.384Z"
 },
    { url = "https://files.pythonhosted.org/packages/a1/5a/4d2b1601df3602dba7a14f3348ba9bfe94a18adb428e693df6154c293831/numpy-2.5.1-cp314-cp314t-win_a
rm64.whl", hash = "sha256:5a6db61f9aaa57e369905c67d852045d3c4f7126405b29d09b19dec118e9c9cb", size = 10697674, upload-time = "2026-07-04T17:07:58.506Z"
 },
]

[[package]]
name = "openai"
version = "2.53.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "distro" },
    { name = "httpx" },
    { name = "jiter" },
    { name = "pydantic" },
    { name = "sniffio" },
    { name = "tqdm" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/ef/cf/36e3e7235fdf6d125c052acc0970924611b17a20a4fe580596faf4566a65/openai-2.53.0.tar.gz", has
h = "sha256:baf5802ad08980e1d9d561e1b996e800c8bcd14af5847c6d0e7a5cc59e4d4116", size = 1099435, upload-time = "2026-08-03T21:42:01.664Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/78/0f/cc6afea3542a5142c5d8fc8211c5e059a8375105d004a41dfa2c7948dbb0/openai-2.53.0-py3-none-any.whl
", hash = "sha256:c694ffc747a3c4d1663ef2b07b811315a476164ee5efa3a993967349ebca7618", size = 1659829, upload-time = "2026-08-03T21:41:59.581Z" },
]

[[package]]
name = "opentelemetry-api"
version = "1.44.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/ee/8b/aa9e2d8b8dfa7c946f7dec5d1f8f6ba8eca062f43509a06bdb5ce93d26c0/opentelemetry_api-1.44.0.t
ar.gz", hash = "sha256:67647e5e9566edcf421166fdf022b3537f818635daa852b289e34604dc6fb33a", size = 72406, upload-time = "2026-07-16T15:25:32.678Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/ca/6f/a04e900f465ff3221ccc395522503e2d10e79fa21f2723c8e177aae1e0d1/opentelemetry_api-1.44.0-py3-n
one-any.whl", hash = "sha256:94b98c893a91b88657eaac1e3ba89618cdb85be6918196705354f34728b2cdef", size = 60018, upload-time = "2026-07-16T15:25:11.657Z"
 },
]

[[package]]
name = "opentelemetry-exporter-otlp-proto-common"
version = "1.44.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "opentelemetry-proto" },
]
sdist = { url = "https://files.pythonhosted.org/packages/61/09/4d717852c1cf3f854b76c7110a5d00883bc3c99288b9b0dbcbeb9e306eb6/opentelemetry_exporter_otl
p_proto_common-1.44.0.tar.gz", hash = "sha256:dc87a5a5bc58f149a56d1547e4691588fa12994cdc3bc039a694ccb3375862ac", size = 20202, upload-time = "2026-07-
16T15:25:37.658Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/5e/71/65fd9d54c10b860f87c045ccee1264cab7011268895d3528818a29c1172a/opentelemetry_exporter_otlp_pr
oto_common-1.44.0-py3-none-any.whl", hash = "sha256:9a9fe61bba73d802904bc989f1d6b4a7b1ee40f06c40e98d6f85af65aaebb694", size = 17045, upload-time = "20
26-07-16T15:25:18.201Z" },
]

[[package]]
name = "opentelemetry-exporter-otlp-proto-http"
version = "1.44.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "googleapis-common-protos" },
    { name = "opentelemetry-api" },
    { name = "opentelemetry-exporter-otlp-proto-common" },
    { name = "opentelemetry-proto" },
    { name = "opentelemetry-sdk" },
    { name = "requests" },
    { name = "typing-extensions" },
]
```

```
sdist = { url = "https://files.pythonhosted.org/packages/1a/87/95e2a5aaa795b4e2260d74e16df2d5541deb2ea9de010bcd615f4dee2654/opentelemetry_exporter_otlp_proto_http-1.44.0.tar.gz", hash = "sha256:c633d7270ad6b57cd4cfbe8b0007a9e2e7c0cb50bd6c50fe2a7b245f721a09d8", size = 25806, upload-time = "2026-07-16T15:25:39.162Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/cd/d0/fdeb1a98d8d3a6205f5f297c51b4a9bfe65126ab60339669bbe3dd54c2e2/opentelemetry_exporter_otlp_proto_http-1.44.0-py3-none-any.whl", hash = "sha256:838592fce774c1c8bb7b9a0a7facbfa82e17be5a8a4e94cef10cb84ae026bae3", size = 21850, upload-time = "2026-07-16T15:25:20.006Z" },
]

[[package]]
name = "opentelemetry-instrumentation"
version = "0.65b0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "opentelemetry-api" },
    { name = "opentelemetry-semantic-conventions" },
    { name = "packaging" },
    { name = "wrapt" },
]
sdist = { url = "https://files.pythonhosted.org/packages/13/91/3c58961cb0360cd60509064734f0be4275383c8681d73c580a40ca83ddce/opentelemetry_instrumentation-0.65b0.tar.gz", hash = "sha256:071d9d9eced9bd6460444ec3b0c77229870ed05a881c22c84fdede58e4eed09b", size = 42689, upload-time = "2026-07-16T15:25:50.275Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/40/7b/85eab1215f72adf0e68d3dc4a679b9bff993fa679ff34cd8dd378e2659fd/opentelemetry_instrumentation-0.65b0-py3-none-any.whl", hash = "sha256:ea967a72b9939b5fcfdad572753b4306c59dcb99e3f382d95dae04286805e137", size = 36717, upload-time = "2026-07-16T15:24:51.424Z" },
]

[[package]]
name = "opentelemetry-instrumentation-asgi"
version = "0.65b0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "asgiref" },
    { name = "opentelemetry-api" },
    { name = "opentelemetry-instrumentation" },
    { name = "opentelemetry-semantic-conventions" },
    { name = "opentelemetry-util-http" },
]
sdist = { url = "https://files.pythonhosted.org/packages/17/83/8e8e83b7ac285281687c7be2fd305213ccccbb8c0a2dd4fb45a8ccaf12c7/opentelemetry_instrumentation_asgi-0.65b0.tar.gz", hash = "sha256:892bca67c56522ffa85a8a83cf934d7b50b3be2132e45cbee705825f0a5ba426", size = 26140, upload-time = "2026-07-16T15:25:54.544Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/0b/9c/376962840b619d2d55fe8ee2285f8c70971c090e5fff614516fc654a6f3a/opentelemetry_instrumentation_asgi-0.65b0-py3-none-any.whl", hash = "sha256:3a845a8ebd1c4ef0d8263401e6545f5b219b2feee612090d50f578a87e71fd65", size = 15903, upload-time = "2026-07-16T15:24:57.198Z" },
]

[[package]]
name = "opentelemetry-instrumentation-fastapi"
version = "0.65b0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "opentelemetry-api" },
    { name = "opentelemetry-instrumentation" },
    { name = "opentelemetry-instrumentation-asgi" },
    { name = "opentelemetry-semantic-conventions" },
    { name = "opentelemetry-util-http" },
]
sdist = { url = "https://files.pythonhosted.org/packages/30/23/b057f8196d06efdc1b50e3ff11fbc499a7d96b35c87f217eb7885542f4ea/opentelemetry_instrumentation_fastapi-0.65b0.tar.gz", hash = "sha256:10a3a9548603623041 3a58fe4fdf4a83fa6bba46918407e527476994bd92bd97", size = 26236, upload-time = "2026-07-16T15:26:05.954Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/fa/b0/c9b0300d33349ecc3dfd2362516eaffc44877e90970e6a52178ff953fec3/opentelemetry_instrumentation_fastapi-0.65b0-py3-none-any.whl", hash = "sha256:cda2610a0ec1b22d19886f33e4d861e9f5dbb886aeaa3a1263b47aff82c36943", size = 13261, upload-time = "2026-07-16T15:25:12.429Z" },
]

[[package]]
name = "opentelemetry-proto"
version = "1.44.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "protobuf" },
]
sdist = { url = "https://files.pythonhosted.org/packages/64/01/40ac4ae9a149263cc52c2cee200ddd80cb6d8db1a4610abf8eabce0fe771/opentelemetry_proto-1.44.0.tar.gz", hash = "sha256:c547a79c2f8c0c515d31509154682e5921c7cfd5ca67b70e1f9266e2c3e103f3", size = 46488, upload-time = "2026-07-16T15:25:45.34Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d1/7c/8be563d68e93bbefa5c8affb82ddcff91b3ad858ce49957ba7b16fd3e0ab/opentelemetry_proto-1.44.0-py3-none-any.whl", hash = "sha256:898b155a0e1557afd867478fb6158e8122a46329ca0bb8dc53cc55e98f017f56", size = 72483, upload-time = "2026-07-16T15:25:28.429Z" },
]

[[package]]
name = "opentelemetry-sdk"
version = "1.44.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "opentelemetry-api" },
    { name = "opentelemetry-semantic-conventions" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/5d/77/a6592cbc7c8d9bcc9d6757a9df45e04a7c585e3e6e7a13456da522b21109/opentelemetry_sdk-1.44.0.tar.gz", hash = "sha256:cebe7f65dc12f26ead75c6064de12fd2a9052e5060c0272d402cfa203aae123b", size = 208624, upload-time = "2026-07-16T15:25:46.078Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e7/23/ff077e61886ee020a17ce9c8b6fa11c601c8d8345b09ea24f605445df62a/opentelemetry_sdk-1.44.0-py3-none-any.whl", hash = "sha256:df081c4c6bcfdb1211e3e8614037679264 3128a25f8d72d1d27675936e7e96ad", size = 137221, upload-time = "2026-07-16T15:25:29.534Z" },
]

[[package]]
name = "opentelemetry-semantic-conventions"
version = "0.65b0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
```

```
    { name = "opentelemetry-api" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/8f/73/0cbdebcb4cf545fdd328da14f5137e37d0770c3f26185e478b0d15d94f50/opentelemetry_semantic_conventions-0.65b0.tar.gz", hash = "sha256:f9b2b81e9d5b64f11bc952075e7e9c7fb0aab075c7fd1c46d597f1b919852d60", size = 148774, upload-time = "2026-07-16T15:25:46.902Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a6/0e/49df70d9b81fb5cbae4bbf2a49d865b09bcbcbc4eb53f5851b1027738d78/opentelemetry_semantic_conventions-0.65b0-py3-none-any.whl", hash = "sha256:1cacde7b0ad306f84c5ef08c3dbe1bbaf20165bba6f8bff43b670e555a086bcb", size = 204645, upload-time = "2026-07-16T15:25:30.688Z" },
]

[[package]]
name = "opentelemetry-util-http"
version = "0.65b0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/32/a9/d7525a59fdd240e69b5af4a6338e78fafa1b4203394122cbd6701fb5f84a/opentelemetry_util_http-0.65b0.tar.gz", hash = "sha256:84f82d826978bba416ab453460ff6a7391cdc3534c93a786595e4068680016b7", size = 11243, upload-time = "2026-07-16T15:26:27.898Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/23/3f/ab8d29df207ce5f470a07fa96ebb48af4e95b7fab7e7635311b9a32f2fab/opentelemetry_util_http-0.65b0-py3-none-any.whl", hash = "sha256:7553b606f963097cb190536dc30556cce85090692e471a422fff30ca29b04348", size = 8245, upload-time = "2026-07-16T15:25:46.482Z" },
]

[[package]]
name = "orjson"
version = "3.11.9"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/7e/0c/964746fcafbd16f8ff53219ad9f6b412b34f345c75f384ad434ceaadb538/orjson-3.11.9.tar.gz", has h = "sha256:4fef17e1f8722c11587a6ef18e35902450221da0028e65dbaaa543619e68e48f", size = 5599163, upload-time = "2026-05-06T15:11:08.309Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/16/6d/11867a3ffa3a3608d84a4de51ef4dd0896d6b5cc9132fbe1daf593e677bc/orjson-3.11.9-cp312-cp312-macosx_10_15_x86_64.macosx_11_0_arm64.macosx_10_15_universal2.whl", hash = "sha256:9ef6fe90aadef185c7b128859f40beb24720b4ecea95379fc9000931179c3a49", size = 228515, upload-time = "2026-05-06T15:09:57.265Z" },
    { url = "https://files.pythonhosted.org/packages/24/75/05912954c8b288f34fcf5cd4b9b071cb4f6e77b9961e175e56ebb258089f/orjson-3.11.9-cp312-cp312-macosx_15_0_arm64.whl", hash = "sha256:e5c9b8f28e726e97d97696c826bc7bea5d71cecd63576dba92924a32c1961291", size = 128409, upload-time = "2026-05-06T15:09:59.063Z" },
    { url = "https://files.pythonhosted.org/packages/ab/86/1c3a47df3bc8191ea9ac51603bbb872a95167a364320c269f2557911f406/orjson-3.11.9-cp312-cp312-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:26a473dbb4162108b27901492546f83c76fdcea3d0eadff00ae7a07e18dcce09", size = 132106, upload-time = "2026-05-06T15:10:00.798Z" },
    { url = "https://files.pythonhosted.org/packages/d7/cf/b33b5f3e695ae7d63feef9d915c37cc3b8f465493dcd4f8e0b4c697a2366/orjson-3.11.9-cp312-cp312-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:011382e2a60fda9d46f1cdee31068cfc52ffe952b587d683ec0463002802a0f4", size = 127864, upload-time = "2026-05-06T15:10:02.15Z" },
    { url = "https://files.pythonhosted.org/packages/31/6a/6cf69385a58208024fcb8c014e2141b8ce838aba6492b589f8acfff97fab/orjson-3.11.9-cp312-cp312-manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:c2d3dc759490128c5c1711a53eeaa8ee1d437fd0038ffd2b6008abf46db3f882", size = 135213, upload-time = "2026-05-06T15:10:03.515Z" },
    { url = "https://files.pythonhosted.org/packages/e8/f8/0b1bd3e8f2efcdd376af5c8cfd79eaf13f018080c0089c80ebd724e3c7fb/orjson-3.11.9-cp312-cp312-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:d8ea516b3726d190e1b4297e6f4e7a8650347ae053868a18163b4dd3641d1fff", size = 145994, upload-time = "2026-05-06T15:10:05.083Z" },
    { url = "https://files.pythonhosted.org/packages/f3/59/dab79f61044c529d2c81aecdc589b1f833a1c8dec11ba3b1c2498a02ca7e/orjson-3.11.9-cp312-cp312-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:380cdce7ba24989af81d0a7013d0aaec5d0e2a21734c0e2681b1bc4f141957fe", size = 132744, upload-time = "2026-05-06T15:10:06.853Z" },
    { url = "https://files.pythonhosted.org/packages/0e/a4/82b7a2fe5d8a67a59ed831b24d59a3d46ea7d207b66e1602d376541d94a6/orjson-3.11.9-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:be4fa4f0af7fa18951f7ab3fc2148e223af211bf03f59e1c6034ec3f97f21d61", size = 134014, upload-time = "2026-05-06T15:10:08.213Z" },
    { url = "https://files.pythonhosted.org/packages/50/c7/375e83a76851b73b2e39f3bcf0e5a19e2b89bad13e5bca97d0b293d27f24/orjson-3.11.9-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:a8f5f8bc7ce7d59f08d9f99fa510c06496164a24cb5f3d34537dbd9ca30132e2", size = 141509, upload-time = "2026-05-06T15:10:09.595Z" },
    { url = "https://files.pythonhosted.org/packages/7f/7c/49d5d82a3d3097f641f094f552131f1e2723b0b8cb0fa2874ab65ecfffa6/orjson-3.11.9-cp312-cp312-musllinux_1_2_armv7l.whl", hash = "sha256:4d7fde5501b944f83b3e665e1b31343ff6e154b15560a16b7130ea1e594a4206", size = 415127, upload-time = "2026-05-06T15:10:11.049Z" },
    { url = "https://files.pythonhosted.org/packages/3a/dc/7446c538590d55f455647e5f3c61fc33f7108714e7afcffa6a2a033f8350/orjson-3.11.9-cp312-cp312-musllinux_1_2_i686.whl", hash = "sha256:cde1a448023ba7d5bb4c01c5afb48894380b5e4956e0627266526587ef4e535f", size = 148025, upload-time = "2026-05-06T15:10:12.842Z" },
    { url = "https://files.pythonhosted.org/packages/df/e5/4d2d8af06f788329b4f78f8cc3679bb395392fcaa1e4d8d3c33e85308fa4/orjson-3.11.9-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:71e63adb0e1f1ed5d9e168f50a91ceb93ae6420731d222dc7da5c69409aa47aa", size = 136943, upload-time = "2026-05-06T15:10:14.405Z" },
    { url = "https://files.pythonhosted.org/packages/06/69/850264ccf6d80f6b174620d30a87f65c9b1490aba33fe6b62798e618cad3/orjson-3.11.9-cp312-cp312-win32.whl", hash = "sha256:2d057a602cdd19a0ad680417527c45b6961a095081c0f46fe0e03e304aac6470", size = 131606, upload-time = "2026-05-06T15:10:15.791Z" },
    { url = "https://files.pythonhosted.org/packages/b9/d5/973a43fc9c55e20f2051e9830997649f669be0cb3ca52192087c0143f118/orjson-3.11.9-cp312-cp312-win_amd64.whl", hash = "sha256:59e403b1cc5a676da8eaf31f6254801b7341b3e29efa85f92b48d272637e77be", size = 127101, upload-time = "2026-05-06T15:10:17.129Z" },
    { url = "https://files.pythonhosted.org/packages/fe/ae/495470f0e4a18f73fa10b7f6b84b464ec4cc5291c4e0c7c2a6c400bef006/orjson-3.11.9-cp312-cp312-win_arm64.whl", hash = "sha256:9af678d6488357948f1f84c6cd1c1d397c014e1ae2f98ae082a44eb48f602624", size = 126736, upload-time = "2026-05-06T15:10:18.645Z" },
    { url = "https://files.pythonhosted.org/packages/32/33/93fcc25907235c344ae73122f8a4e01d2d393ef062b4af7d2e2487a32c37/orjson-3.11.9-cp313-cp313-macosx_10_15_x86_64.macosx_11_0_arm64.macosx_10_15_universal2.whl", hash = "sha256:4bab1b2d6141fe7b32ae71dac905666ece4f94936efbfb13d55bb7739a3a6021", size = 228458, upload-time = "2026-05-06T15:10:20.079Z" },
    { url = "https://files.pythonhosted.org/packages/8f/27/b1e6dadb3c080313c03fdd8067b85e6a0460c7d8d6a1c3984ef77b904e4d/orjson-3.11.9-cp313-cp313-macosx_15_0_arm64.whl", hash = "sha256:844417969855fc7a41be124aafe83dc424592a7f77cd4501900c67307122b92c", size = 128368, upload-time = "2026-05-06T15:10:21.549Z" },
    { url = "https://files.pythonhosted.org/packages/21/0f/c9ede0bf052f6b4051e64a7d4fa91b725cccf8321a6a786e86eb03519f00/orjson-3.11.9-cp313-cp313-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:ffe02797b5e9f3a9d8292ddcd289b474ad13e81ad83cd1891a240811f1d2cb81", size = 132070, upload-time = "2026-05-06T15:10:23.371Z" },
    { url = "https://files.pythonhosted.org/packages/fd/26/d398e28048dc18205bbe812f2c88cb9b40313db2470778e25964796458fe/orjson-3.11.9-cp313-cp313-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:0e4eed3b200023042814d2fc8a5d2e880f13b52e1ed2485e83da4f3962f7dc1a", size = 127892, upload-time = "2026-05-06T15:10:24.714Z" },
    { url = "https://files.pythonhosted.org/packages/66/60/52b0054c4c700d5aa7fc5b7ca96917400d8f061307778578e67a10e25852/orjson-3.11.9-cp313-cp313-manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:8aff7da9952a5ad1cef8e68017724d96c7b9a66e99e91d6252e1b133d67a7b10", size = 135217, upload-time = "2026-05-06T15:10:26.084Z" },
    { url = "https://files.pythonhosted.org/packages/d5/97/1e3dc2b2a28b7b2528f403d2fc1d79ec5f39af3bc143ab65d3ec26426385/orjson-3.11.9-cp313-cp313-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:4d4e98d6f3b8afed8bc8cd9718ec0cdf46661826beefb53fe8eafb37f2bf0362", size = 145980, upload-time = "2026-05-06T15:10:28.062Z" },
    { url = "https://files.pythonhosted.org/packages/fc/39/31fbfe7850f2de32dee7e7e5c09f26d403ab01e440ac96001c6b01ad3c99/orjson-3.11.9-cp313-cp313-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:3a81d52442a7c99b3662333235b3adf96a1715864658b35bb797212be7bddb97", size = 132738, upload-time = "2026-05-06T15:10:29.727Z" },
    { url = "https://files.pythonhosted.org/packages/a1/08/dca0082dd2a194acb93e5457e73455388e2e2ca464a2672449a9ddbb679d/orjson-3.11.9-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:4e39364e726a8fff737309aff059ff67d8a8c8d5b677be7bb49a8b3e84b7e218", size = 134033, upload-time = "2026-05-06T15:10:31.152Z" },
    { url = "https://files.pythonhosted.org/packages/11/d4/5bdb06268012301399873855 54c5d4c42255218ac906525bf4347f22cd95/orjson-3.11.9-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:4fd66214623f1b17501df9f0543bef0b833979ab5b6ded1e1d123222866aa8c9", size = 141492, upload-time = "2026-05-06T15:10:32.641Z" },
    { url = "https://files.pythonhosted.org/packages/fa/88/a21fb53b3ede6703aede6dce4710ed4111e5b201cfa6bbff5e544f9d47d7/orjson-3.11.9-cp313-cp313-musllinux_1_2_armv7l.whl", hash = "sha256:8ecc30f10465fa1e0ce13fd01d9e22c316e5053a719a8d915d4545a09a5ff677", size = 415087, upload-time = "2026-05-06T15:1
```

0:34.438Z" },
    { url = "https://files.pythonhosted.org/packages/3d/57/1b30daf70f0d8180e9a73cefbfbdd99e4bf19eb020466502b01fba7e0e50/orjson-3.11.9-cp313-cp313-musl
linux_1_2_i686.whl", hash = "sha256:97db4c94a7db398a5bd636273324f0b3fd58b350bbbac8bb380ceb825a9b40f4", size = 148031, upload-time = "2026-05-06T15:10:
36.358Z" },
    { url = "https://files.pythonhosted.org/packages/04/83/45fbb6d962e260807f99441db9613cee868ceda4baceda59b3720a563f97/orjson-3.11.9-cp313-cp313-musl
linux_1_2_x86_64.whl", hash = "sha256:9f78cf8fec5bd627f4082b8dfeac7871b43d7f327490449 2a43dab39f18a19a0", size = 136915, upload-time = "2026-05-06T15:1
0:38.013Z" },
    { url = "https://files.pythonhosted.org/packages/5f/cc/2d10025f9056d376e4127ec05a5808b218d46f035fdc08178a5411b34250/orjson-3.11.9-cp313-cp313-win3
2.whl", hash = "sha256:d4087e5c0209a0a8efe4de3303c234b9c44d1174161dcd851e8eea07c7560b32", size = 131613, upload-time = "2026-05-06T15:10:39.569Z" },
    { url = "https://files.pythonhosted.org/packages/67/bd/2775ff28bfe883b9aa1ff348300542eb2ef1ee18d8ae0e3a49846817a865/orjson-3.11.9-cp313-cp313-win_
amd64.whl", hash = "sha256:051b102c93b4f634e89f3866b07b9a9a98915ada541f4ec30f177067b2694979", size = 127086, upload-time = "2026-05-06T15:10:41.262Z"
},
    { url = "https://files.pythonhosted.org/packages/91/2b/d26799e580939e32a7da9a39531bc9e58e15ca32ffaa6a8cb3e9bb0d22cd/orjson-3.11.9-cp313-cp313-win_
arm64.whl", hash = "sha256:cce9127885941bd28f080cecf1f1d288336b7e0d812c345b08be88b572796254", size = 126696, upload-time = "2026-05-06T15:10:42.651Z"
},
    { url = "https://files.pythonhosted.org/packages/8e/eb/5da01e356015aee6ecfa1187ced87aef51364e306f5e695dd52719bf0e78/orjson-3.11.9-cp314-cp314-maco
sx_10_15_x86_64.macosx_11_0_arm64.macosx_10_15_universal2.whl", hash = "sha256:b6ef1979adc4bc243523f1a2ba91418030a8e29b0a99cbe7e0e2d6807d4dce6e", size
 = 228465, upload-time = "2026-05-06T15:10:44.097Z" },
    { url = "https://files.pythonhosted.org/packages/64/62/3e0e0c14c957133bcd855395c62b55ed4e3b0af23ffea11b032cb1dcbdb1/orjson-3.11.9-cp314-cp314-maco
sx_15_0_arm64.whl", hash = "sha256:f36b7f32c7c0db4a719f1fc5824db4a9c6f8bd1a354debb91faf26ebf3a4c71e", size = 128364, upload-time = "2026-05-06T15:10:4
5.839Z" },
    { url = "https://files.pythonhosted.org/packages/5a/5a/07d8aa117211a8ed7630bda80c8c0b14d04e0f8dcf99bcf49656e4a710eb/orjson-3.11.9-cp314-cp314-many
linux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:08f4d8ebb44925c794e535b2bebc507cebf32209df81de22ae285fb0d8d66de0", size = 132063, upload
-time = "2026-05-06T15:10:47.267Z" },
    { url = "https://files.pythonhosted.org/packages/d6/ec/4acaf21483e18aa945be74a474c74b434f284b549f275a0a39b9f98956e9/orjson-3.11.9-cp314-cp314-many
linux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:6cc7923789694fd58f001cbcac7e47abc13af4d560ebbfcf3b41a8b1a0748124", size = 122356, upload-t
ime = "2026-05-06T15:10:48.765Z" },
    { url = "https://files.pythonhosted.org/packages/13/d8/5f0555e7638801323b7a75850f92e7dfa891bc84fe27a1ba4449170d1200/orjson-3.11.9-cp314-cp314-many
linux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:ea5c46eb2d3af39e806b986f4b09d5c2706a1f5afde3cbf7544ce6616127173c", size = 129592, upload-time
= "2026-05-06T15:10:50.13Z" },
    { url = "https://files.pythonhosted.org/packages/b6/30/ed9860412a3603ceb3c5955bfd72d28b9d0e7ba6ed81add14f83d7114236/orjson-3.11.9-cp314-cp314-many
linux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:f5d89a2ed90731df3be64bab0aa44f78bff39fdc9d71c291f4a8023aa46425b7", size = 140491, upload
-time = "2026-05-06T15:10:51.582Z" },
    { url = "https://files.pythonhosted.org/packages/d0/17/adc514dea7ac7c505527febf884934b815d34f0c7b8693c1a8b39c5c4a57/orjson-3.11.9-cp314-cp314-many
linux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:25e4aed0312d292c09f61af25bba34e0b2c88546041472b09088c39a4d828af1", size = 127309, upload-tim
e = "2026-05-06T15:10:53.329Z" },
    { url = "https://files.pythonhosted.org/packages/76/3e/c0b690253f0b82d86e99949af13533363acfb5432ecb5d53dd5b3bce9c34/orjson-3.11.9-cp314-cp314-many
linux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:aaea64f3f467d22e70eeed68bdccb3bc4f83f650446c4a03c59f2cba28a108db", size = 134030, upload-t
ime = "2026-05-06T15:10:54.988Z" },
    { url = "https://files.pythonhosted.org/packages/c1/7a/bc82a0bb25e9faaf92dc4d9ef002732efc09737706af83e346788641d4a7/orjson-3.11.9-cp314-cp314-musl
linux_1_2_aarch64.whl", hash = "sha256:a028425d1b440c5d92a6be1e1a020739dfe67ea87d96c6dbe828c1b30041728b", size = 141482, upload-time = "2026-05-06T15:
10:56.663Z" },
    { url = "https://files.pythonhosted.org/packages/01/55/e69188b939f77d5d32a9833745ace31ea5ccae3ab613a1ec185d3cd2c4fb/orjson-3.11.9-cp314-cp314-musl
linux_1_2_armv7l.whl", hash = "sha256:5b192c6cf397e4455b11523c5cf2b18ed084c1bbd61b6c0926344d2129481972", size = 415178, upload-time = "2026-05-06T15:1
0:58.446Z" },
    { url = "https://files.pythonhosted.org/packages/2e/1a/b8a5a7ac527e80b9cb11d51e3f6689b709279183264b9ec5c7bc680bb8b5/orjson-3.11.9-cp314-cp314-musl
linux_1_2_i686.whl", hash = "sha256:ea407d4ccf5891d667d045fecae97a7a1e5e87b3b97f97ae1803c2e741130be0", size = 148089, upload-time = "2026-05-06T15:11:
00.441Z" },
    { url = "https://files.pythonhosted.org/packages/97/4e/00503f64204bf859b37213a63927028f30fb6268cd8677fb0a5ad48155e1/orjson-3.11.9-cp314-cp314-musl
linux_1_2_x86_64.whl", hash = "sha256:5f63aaf97afd9f6dec5b1a68e1b8da12bfccb4cb9a9a65c3e0b6c847849e7586", size = 136921, upload-time = "2026-05-06T15:1
1:02.176Z" },
    { url = "https://files.pythonhosted.org/packages/0d/ba/a23b82a0a8d0ed7bed4e5f5035aae751cad4ff6a1e8d2ecd14d8860f5929/orjson-3.11.9-cp314-cp314-win3
2.whl", hash = "sha256:e30ab17845bb9fa54ccf67fa4f9f5282652d54faa6d17452f47d0f369d038673", size = 131638, upload-time = "2026-05-06T15:11:03.696Z" },
    { url = "https://files.pythonhosted.org/packages/f3/c3/0c6798456bade745c75c452342dabacce5798196483e77e643be1f53877d/orjson-3.11.9-cp314-cp314-win_
amd64.whl", hash = "sha256:32ef5f4283a3be81913947d19608eacb7c6608026851123790cd9cc8982af34b", size = 127078, upload-time = "2026-05-06T15:11:05.123Z"
},
    { url = "https://files.pythonhosted.org/packages/16/21/5a3f1e8913103b703a436a5664238e5b965ec392b555fe68943ea3691e6b/orjson-3.11.9-cp314-cp314-win_
arm64.whl", hash = "sha256:eebdbdeef0094e4f5aefa20dcd4eb2368ab5e7a3b4edea27f1e7b2892e009cf9", size = 126687, upload-time = "2026-05-06T15:11:06.602Z"
},
]

[[package]]
name = "ormsgpack"
version = "1.12.2"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/12/0c/f1761e21486942ab9bb6feaebc610fa074f7c5e496e6962dea5873348077/ormsgpack-1.12.2.tar.gz",
hash = "sha256:944a2233640273bee67521795a73cf1e959538e0dfb7ac635505010455e53b33", size = 39031, upload-time = "2026-01-18T20:55:28.023Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/4c/36/16c4b1921c308a92cef3bf6663226ae283395aa0ff6e154f925c32e91ff5/ormsgpack-1.12.2-cp312-cp312-m
acosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:7a29d09b64b9694b588ff2f80e9826bdceb3a2b91523c5beae1fab27d5c940e7", s
ize = 378618, upload-time = "2026-01-18T20:55:50.835Z" },
    { url = "https://files.pythonhosted.org/packages/c0/68/468de634079615abf66ed13bb5c34ff71da237213f29294363beeeca5306/ormsgpack-1.12.2-cp312-cp312-m
anylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:0b39e629fd2e1c5b2f46f99778450b59454d1f901bc507963168985e79f09c5d", size = 203186, upl
oad-time = "2026-01-18T20:56:11.163Z" },
    { url = "https://files.pythonhosted.org/packages/73/a9/d756e01961442688b7939bacd87ce13bfad7d26ce24f910f6028178b2cc8/ormsgpack-1.12.2-cp312-cp312-m
anylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:958dcb270d30a7cb633a45ee62b9444433fa571a752d2ca484efdac07480876e", size = 210738, uploa
d-time = "2026-01-18T20:56:09.181Z" },
    { url = "https://files.pythonhosted.org/packages/7b/ba/795b1036888542c9113269a3f5690ab53dd2258c6fb17676ac4bd44fcf94/ormsgpack-1.12.2-cp312-cp312-m
anylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:58d379d72b6c5e964851c77cfedfb386e474adee4fd39791c2c5d9efb53505cc", size = 212569, uploa
d-time = "2026-01-18T20:56:06.135Z" },
    { url = "https://files.pythonhosted.org/packages/6c/aa/bff73c57497b9e0cba8837c7e4bcab584b1a6dbc91a5dd5526784a5030c8/ormsgpack-1.12.2-cp312-cp312-m
usllinux_1_2_aarch64.whl", hash = "sha256:8463a3fc5f09832e67bdb0e2fda6d518dc4281b133166146a67f54c08496442e", size = 387166, upload-time = "2026-01-18T
20:55:36.738Z" },
    { url = "https://files.pythonhosted.org/packages/d3/cf/f8283cba44bcb7b14f97b6274d449db276b3a86589bdb363169b51bc12de/ormsgpack-1.12.2-cp312-cp312-m
usllinux_1_2_armv7l.whl", hash = "sha256:eddffb77eff0bad4e67547d67a130604e7e2dfbb7b0cde0796045be4090f35c6", size = 482498, upload-time = "2026-01-18T2
0:55:29.626Z" },
    { url = "https://files.pythonhosted.org/packages/05/be/71e37b852d723dfcbe952ad04178c030df60d6b78eba26bfd14c9a40575e/ormsgpack-1.12.2-cp312-cp312-m
usllinux_1_2_x86_64.whl", hash = "sha256:fcd55e5f6ba0dbce624942adf9f152062135f991a0126064889f68eb850de0dd", size = 425518, upload-time = "2026-01-18T2
0:55:49.556Z" },
    { url = "https://files.pythonhosted.org/packages/7a/0c/9803aa883d18c7ef197213cd2cbf73ba76472a11fe100fb7dab2884edf48/ormsgpack-1.12.2-cp312-cp312-w
in_amd64.whl", hash = "sha256:d024b40828f1dde5654faebd0d824f9cc29ad46891f626272dd5bfd7af2333a4", size = 117462, upload-time = "2026-01-18T20:55:47.726
Z" },
    { url = "https://files.pythonhosted.org/packages/c8/9e/029e898298b2cc662f10d7a15652a53e3b525b1e7f07e21fef8536a09bb8/ormsgpack-1.12.2-cp312-cp312-w
in_arm64.whl", hash = "sha256:da538c542bac7d1c8f3f2a937863dba36f013108ce63e55745941dda4b75dbb6", size = 111559, upload-time = "2026-01-18T20:55:54.273
Z" },
    { url = "https://files.pythonhosted.org/packages/eb/29/bb0eba3288c0449efbb013e9c6f58aea79cf5cb9ee1921f8865f04c1a9d7/ormsgpack-1.12.2-cp313-cp313-m
acosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:5ea60cb5f210b1cfbad8c002948d73447508e629ec375acb82910e3efa8ff355", s
ize = 378661, upload-time = "2026-01-18T20:55:57.765Z" },
    { url = "https://files.pythonhosted.org/packages/6e/31/5efa31346affdac489acade2926989e019e8ca98129658a183e3add7af5e/ormsgpack-1.12.2-cp313-cp313-m
anylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:f3601f19afdbea273ed70b06495e5794606a8b690a568d6c996a90d7255e51c1", size = 203194, upl
oad-time = "2026-01-18T20:56:08.252Z" },
    { url = "https://files.pythonhosted.org/packages/eb/56/d0087278beef833187e0167f8527235ebe6f6ffc2a143e9de12a98b1ce87/ormsgpack-1.12.2-cp313-cp313-m
anylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:29a9f17a3dac6054c0dce7925e0f4995c727f7c41859adf9b5572180f640d172", size = 210778, uploa
d-time = "2026-01-18T20:55:17.694Z" },
    { url = "https://files.pythonhosted.org/packages/1c/a2/072343e1413d9443e5a252a8eb591c2d5b1bffbe5e7bfc78c069361b92eb/ormsgpack-1.12.2-cp313-cp313-m

```
anylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:39c1bd2092880e413902910388be8715f70b9f15f20779d44e673033a6146f2d", size = 212592, uploa
d-time = "2026-01-18T20:55:32.747Z" },
    { url = "https://files.pythonhosted.org/packages/a2/8b/a0da3b98a91d41187a63b02dda14267eefc2a74fcb43cc2701066cf1510e/ormsgpack-1.12.2-cp313-cp313-m
usllinux_1_2_aarch64.whl", hash = "sha256:50b7249244382209877deedeee838aef1542f3d0fc28b8fe71ca9d7e1896a0d7", size = 387164, upload-time = "2026-01-18T
20:55:40.853Z" },
    { url = "https://files.pythonhosted.org/packages/19/bb/6d226bc4cf9fc20d8eb1d976d027a3f7c3491e8f08289a2e76abe96a65f3/ormsgpack-1.12.2-cp313-cp313-m
usllinux_1_2_armv7l.whl", hash = "sha256:5af04800d844451cf102a59c74a841324868d3f1625c296a06cc655c542a6685", size = 482516, upload-time = "2026-01-18T2
0:55:42.033Z" },
    { url = "https://files.pythonhosted.org/packages/fb/f1/bb2c7223398543dedb3dbf8bb93aaa737b387de61c5feaad6f908841b782/ormsgpack-1.12.2-cp313-cp313-m
usllinux_1_2_x86_64.whl", hash = "sha256:cec70477d4371cd524534cd16472d8b9cc187e0e3043a8790545a9a9b296c258", size = 425539, upload-time = "2026-01-18T2
0:55:24.727Z" },
    { url = "https://files.pythonhosted.org/packages/7b/e8/0fb45f57a2ada1fed374f7494c8cd55e2f88ccd0ab0a669aa3468716bf5f/ormsgpack-1.12.2-cp313-cp313-w
in_amd64.whl", hash = "sha256:21f4276caca5c03a818041d637e4019bc84f9d6ca8baa5ea03e5cc8bf56140e9", size = 117459, upload-time = "2026-01-18T20:55:56.876
Z" },
    { url = "https://files.pythonhosted.org/packages/7a/d4/0cfeea1e960d550a131001a7f38a5132c7ae3ebde4c82af1f364ccc5d904/ormsgpack-1.12.2-cp313-cp313-w
in_arm64.whl", hash = "sha256:baca4b6773d20a82e36d6fd25f341064244f9f86a13dead95dd7d7f996f51709", size = 111577, upload-time = "2026-01-18T20:55:43.605
Z" },
    { url = "https://files.pythonhosted.org/packages/94/16/24d18851334be09c25e87f74307c84950f18c324a4d3c0b41dabdbf19c29/ormsgpack-1.12.2-cp314-cp314-m
acosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:bc68dd5915f4acf66ff2010ee47c8906dc1cf07399b16f4089f8c71733f6e36c", s
ize = 378717, upload-time = "2026-01-18T20:55:26.164Z" },
    { url = "https://files.pythonhosted.org/packages/b5/a2/88b9b56f83adae8032ac6a6fa7f080c65b3baf9b6b64fd3d37bd202991d4/ormsgpack-1.12.2-cp314-cp314-m
anylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:46d084427b4132553940070ad95107266656cb646ea9da4975f85cb1a6676553", size = 203183, upl
oad-time = "2026-01-18T20:55:18.815Z" },
    { url = "https://files.pythonhosted.org/packages/a9/80/43e4555963bf602e5bdc79cbc8debd8b6d5456c00d2504df9775e74b450b/ormsgpack-1.12.2-cp314-cp314-m
anylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:c010da16235806cf1d7bc4c96bf286bfa91c686853395a299b3ddb49499a3e13", size = 210814, uploa
d-time = "2026-01-18T20:55:33.973Z" },
    { url = "https://files.pythonhosted.org/packages/78/e1/7cfbf28de8bca6efe7e525b329c31277d1b64ce08dcba723971c241a9d60/ormsgpack-1.12.2-cp314-cp314-m
anylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:18867233df592c997154ff942a6503df274b5ac1765215bceba7a231bea2745d", size = 212634, uploa
d-time = "2026-01-18T20:55:28.634Z" },
    { url = "https://files.pythonhosted.org/packages/95/f8/30ae5716e88d792a4e879debee195653c26ddd3964c968594ddef0a3cc7e/ormsgpack-1.12.2-cp314-cp314-m
usllinux_1_2_aarch64.whl", hash = "sha256:b009049086ddc6b8f80c76b3955df1aa22a5fbd7673c525cd63bf91f23122ede", size = 387139, upload-time = "2026-01-18T
20:56:02.013Z" },
    { url = "https://files.pythonhosted.org/packages/dc/81/aee5b18a3e3a0e52f718b37ab4b8af6fae0d9d6a65103036a90c2a8ffb5d/ormsgpack-1.12.2-cp314-cp314-m
usllinux_1_2_armv7l.whl", hash = "sha256:1dcc17d92b6390d4f18f937cf0b99054824a7815818012ddca925d6e01c2e49e", size = 482578, upload-time = "2026-01-18T2
0:55:35.117Z" },
    { url = "https://files.pythonhosted.org/packages/bd/17/71c9ba472d5d45f7546317f467a5fc941929cd68fb32796ca3d13dcbaec2/ormsgpack-1.12.2-cp314-cp314-m
usllinux_1_2_x86_64.whl", hash = "sha256:f04b5e896d510b07c0ad733d7fce2d44b260c5e6c402d272128f8941984e4285", size = 425539, upload-time = "2026-01-18T2
0:56:04.009Z" },
    { url = "https://files.pythonhosted.org/packages/2e/a6/ac99cd7fe77e822fed5250ff4b86fa66dd4238937dd178d2299f10b69816/ormsgpack-1.12.2-cp314-cp314-w
in_amd64.whl", hash = "sha256:ae3aba7eed4ca7cb79fd3436eddd29140f17ea254b91604aa1eb19bfcedb990f", size = 117493, upload-time = "2026-01-18T20:56:07.343
Z" },
    { url = "https://files.pythonhosted.org/packages/3a/67/339872846a1ae4592535385a1c1f93614138566d7af094200c9c3b45d1e5/ormsgpack-1.12.2-cp314-cp314-w
in_arm64.whl", hash = "sha256:118576ea6006893aea811b17429bfc561b4778fad393f5f538c84af70b01260c", size = 111579, upload-time = "2026-01-18T20:55:21.161
Z" },
    { url = "https://files.pythonhosted.org/packages/49/c2/6feb972dc87285ad381749d3882d8aecbde9f6ecf908dd717d33d66df095/ormsgpack-1.12.2-cp314-cp314t-
macosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:7121b3d355d3858781dc40dafe25a32ff8a8242b9d80c692fd548a4b1f7fd3c8",
size = 378721, upload-time = "2026-01-18T20:55:52.12Z" },
    { url = "https://files.pythonhosted.org/packages/a3/9a/900a6b9b413e0f8a471cf07830f9cf65939af039a362204b36bd5b581d8b/ormsgpack-1.12.2-cp314-cp314t-
manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:4ee766d2e78251b7a63daf1cddfac36a73562d3ddef68cacfb41b2af64698033", size = 203170, up
load-time = "2026-01-18T20:55:44.469Z" },
    { url = "https://files.pythonhosted.org/packages/87/4c/27a95466354606b256f24fad464d7c97ab62bce6cc529dd4673e1179b8fb/ormsgpack-1.12.2-cp314-cp314t-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:292410a7d23de9b40444636b9b8f1e4e4b814af7f1ef476e44887e52a123f09d", size = 212816, uplo
ad-time = "2026-01-18T20:55:23.501Z" },
    { url = "https://files.pythonhosted.org/packages/73/cd/29cee6007bddf7a834e6cd6f536754c0535fcb939d384f0f37a38b1cddb8/ormsgpack-1.12.2-cp314-cp314t-
win_amd64.whl", hash = "sha256:837dd316584485b72ef451d08dd3e96c4a11d12e4963aedb40e08f89685d8ec2", size = 117232, upload-time = "2026-01-18T20:55:45.44
8Z" },
]

[[package]]
name = "packaging"
version = "26.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/7d/fa/3944b40b07da9ce895c0e6303a5ab7d53da063554f534556b134a54d6093/packaging-26.3.tar.gz", ha
sh = "sha256:94edc256424af38762eb31306eed28beb9f0efc50a8837492c9d6fd6004aed79", size = 313412, upload-time = "2026-08-04T18:15:28.737Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/63/34/ba1c580383c9eada3711951fef0795c80b829a078d72188184bcab9dd527/packaging-26.3-py3-none-any.wh
l", hash = "sha256:d7193f7c8e4e93f444fde0262bf90af30e16fa0ad0ad44cb553c87339b23cd1c", size = 129956, upload-time = "2026-08-04T18:15:27.159Z" },
]

[[package]]
name = "pandas"
version = "3.0.5"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "numpy" },
    { name = "python-dateutil" },
    { name = "tzdata", marker = "sys_platform == 'emscripten' or sys_platform == 'win32'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/be/4f/5f3422a2afec5ffc46308b79e53291365a93748b498ac2e58bead0197916/pandas-3.0.5.tar.gz", hash
 = "sha256:dca3734d6ab7c906e6730f0788b0a1dbb9f2467731f9711f77995c8e9d62d712", size = 4658219, upload-time = "2026-07-22T22:19:28.819Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/1c/54/1dc810ea558d1320b597aa140a514f2fdf1d2ea09c38cf556f13ea712ec9/pandas-3.0.5-cp312-cp312-macos
x_10_13_x86_64.whl", hash = "sha256:fa290c16964d4963fbfbc358928239cf3bd755b20e988ce944877def2f44471d", size = 10411717, upload-time = "2026-07-22T22:1
8:08.307Z" },
    { url = "https://files.pythonhosted.org/packages/68/56/fbe81c09195924d8b7b8d4461a20458fe80a6a5ed6b24f0314da684277e1/pandas-3.0.5-cp312-cp312-macos
x_11_0_arm64.whl", hash = "sha256:c2e26bb46934b8a2ca0c3de1d3d606fc5f6746584791b2db264d58cf370e08dc", size = 9957095, upload-time = "2026-07-22T22:18:1
0.6Z" },
    { url = "https://files.pythonhosted.org/packages/e0/51/fac252f4a913ed5eabf3c11b880a9e8d5a6c10f0b2129d0462212d238b4d/pandas-3.0.5-cp312-cp312-manyl
inux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:73fa87b08a7ef706f8aafda39ddaccf2a99047bea62d8c88a0361bcafb2237bc", size = 10485458, uplo
ad-time = "2026-07-22T22:18:12.834Z" },
    { url = "https://files.pythonhosted.org/packages/12/98/e976540c1addf70442be7842a18cf70884a964abbf69442504f4d2939989/pandas-3.0.5-cp312-cp312-manyl
inux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:d373ce03ffd84010ed9839fa73672a9c8256990532e158440c0085db7d914b34", size = 10998091, upload
-time = "2026-07-22T22:18:15.209Z" },
    { url = "https://files.pythonhosted.org/packages/a4/8c/1f29b5be8d3fc47dd7567eb167fabba2085879b31e0287ce7cba6d3d2ff4/pandas-3.0.5-cp312-cp312-musll
inux_1_2_aarch64.whl", hash = "sha256:2a29c53d85ea98c5e792c59ef82ee9fbe6ca902c0d0adb6b23f45ef894cd7bf6", size = 11499501, upload-time = "2026-07-22T22
:18:17.689Z" },
    { url = "https://files.pythonhosted.org/packages/9d/e2/bd9c98ad2df7b38bde002adde4cdf353519da51881634323b126c55997f9/pandas-3.0.5-cp312-cp312-musll
inux_1_2_x86_64.whl", hash = "sha256:a5ad3b02ed6bc7d7ae9b70804b2c6aa31827489d150f8e623ce82491b82085d7", size = 12060559, upload-time = "2026-07-22T22:
18:20.147Z" },
    { url = "https://files.pythonhosted.org/packages/f3/9a/ffbd852d58bd74a617fe2f8ee6a58a96982271ce41cf981eab22190b4a4b/pandas-3.0.5-cp312-cp312-pyems
cripten_2024_0_wasm32.whl", hash = "sha256:b2acb4650527eec6822c3dadb2b771277b65e7dae7a267d4bccf65fd1bb3fbce", size = 7197652, upload-time = "2026-07-2
2T22:18:22.502Z" },
    { url = "https://files.pythonhosted.org/packages/70/b5/d2d3e9ae73362ba4229651b0ee1455cf78073a1ce585f6ff693782ce263e/pandas-3.0.5-cp312-cp312-win_a
md64.whl", hash = "sha256:80a611068e8a3ac23f7398c6c14eb46dc974e5cc9997f653e2dcfd1da74edd41", size = 9831691, upload-time = "2026-07-22T22:18:24.534Z"
},
```

```
    { url = "https://files.pythonhosted.org/packages/52/51/dea1e89d6a6796b9c43f85a09b484ee03edb8a4c4842e73e200a8c11301c/pandas-3.0.5-cp312-cp312-win_arm64.whl", hash = "sha256:25ff585b972a18ef1fe9ffa3ac6544d9950508aa76832e5147640b6022821e49", size = 9105796, upload-time = "2026-07-22T22:18:27.064Z" },
    { url = "https://files.pythonhosted.org/packages/bf/09/7b95c4a0025227d6f118c4039b423412ac6a982db02864166185d812fbc7/pandas-3.0.5-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:c1c05a767fe8e5b4fe9e1c29806829c582052eaedb9120a3da83ba3f69e24a5b", size = 10385742, upload-time = "2026-07-22T22:18:29.346Z" },
    { url = "https://files.pythonhosted.org/packages/8d/0c/dc78fd8c4da477b4b5e8ad37295af352190d21ef63a9ee1bc071753074cc/pandas-3.0.5-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:b86765f268b56f7e665b93bce9d5df69dee7f99e595cf8fb839483ab315942a3", size = 9932067, upload-time = "2026-07-22T22:18:31.833Z" },
    { url = "https://files.pythonhosted.org/packages/3e/71/3592c055cf44df9808550f9368ceda80ff2b224d355ef73fe251dcda1802/pandas-3.0.5-cp313-cp313-manylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:c597ecf5616b5c420372c1d4d4c00dbbfba7398bea857dcc984347e1ea48417b", size = 10466756, upload-time = "2026-07-22T22:18:34.195Z" },
    { url = "https://files.pythonhosted.org/packages/e3/70/4363150359f95b4cb4bcbb34ca23572bb5495749a621a8f3d5a1ddfd293c/pandas-3.0.5-cp313-cp313-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:4b11c36e218331d0387cbe3a0a5f75162357a1d92d57b2b08a336ff94b19b2be", size = 10938525, upload-time = "2026-07-22T22:18:36.81Z" },
    { url = "https://files.pythonhosted.org/packages/f7/d0/317e7a0c67c0e69fa905a0161409397a7dc2d46ff611f6ca4803352c042b/pandas-3.0.5-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:cf52e1f61d229496da17dc7ab54acdee627357e7008fd4fecba3d0ba2937fa58", size = 11489303, upload-time = "2026-07-22T22:18:39.287Z" },
    { url = "https://files.pythonhosted.org/packages/f1/8d/36dade89b49e4f9d5cbdbe863772581f98c0c6d78fc39ad4c557f6f2e17e/pandas-3.0.5-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:db172144bb56422bd157812f3b021eacc255451470b31e2c633c349490a1cfee", size = 11989004, upload-time = "2026-07-22T22:18:42.208Z" },
    { url = "https://files.pythonhosted.org/packages/9c/ba/18c4ec8a746e177da05a9e7a7963781d8ea195780724f854601b6ebd6b78/pandas-3.0.5-cp313-cp313-win_amd64.whl", hash = "sha256:0d298e951f23016ce4699951d044ae6418dbc91bf68cefca0f77666fcbb4e5c6", size = 9826896, upload-time = "2026-07-22T22:18:44.539Z" },
    { url = "https://files.pythonhosted.org/packages/de/ec/28a57266b753799a87b8bc79e7887ac6fd981b8c6d2978a0b7e7b6bd708c/pandas-3.0.5-cp313-cp313-win_arm64.whl", hash = "sha256:66266d3442a5e8b3c90274c2b8b230bee42dd1c286bc822cc2f9f2c7e12b883e", size = 9094790, upload-time = "2026-07-22T22:18:47.468Z" },
    { url = "https://files.pythonhosted.org/packages/51/2f/cf6aae281264f4463f0875bcbb15fd2bb6d291cc535187dad1732475e4a9/pandas-3.0.5-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:2f264fc46911cc8131a7322a16199bbf8e353d27c10bb211f5bd0c814324dc36", size = 10390034, upload-time = "2026-07-22T22:18:49.818Z" },
    { url = "https://files.pythonhosted.org/packages/06/ec/5189518c7a7659c4bdcc6b1eb32c46c6f3c86b0661ffd84143d1112c7732/pandas-3.0.5-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:53730687fcd161883b24e10411c06d6a4c0f2275d2faf3bb2bc25deb4ba8007c", size = 9980065, upload-time = "2026-07-22T22:18:52.249Z" },
    { url = "https://files.pythonhosted.org/packages/ea/f1/598503ce8d7e3c35601e0747ba288c7864baae66380725bc12f13f884dfe/pandas-3.0.5-cp314-cp314-manylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:960d3ebcf249f75206899fcd2c6de53f736b7265759ced0d3e559df0b8b709b0", size = 10545532, upload-time = "2026-07-22T22:18:54.813Z" },
    { url = "https://files.pythonhosted.org/packages/fa/de/ceae2adf7034e07e9910299fe412e1819c4f0dd520700a888bcb03625448/pandas-3.0.5-cp314-cp314-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9e94c2c5ca43bd3ca32bf64d32308887b65e5f9bfd8023ea52755107a999f93b", size = 10963120, upload-time = "2026-07-22T22:18:57.42Z" },
    { url = "https://files.pythonhosted.org/packages/66/25/86e0f4451874eb79e688deeebe3c451fec4557f8952005818d800ee8ac7e/pandas-3.0.5-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:e819dd5f62966b481a8cb649d3299ebd886a1ea91ed5a99bf7ce77c98d18ab94", size = 11563178, upload-time = "2026-07-22T22:18:59.729Z" },
    { url = "https://files.pythonhosted.org/packages/f3/45/8643daa3b4147e433adfcccefdd0380d3aad79d86b15d8999730fe1944d5/pandas-3.0.5-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:3c5ed2e7c06e91d340dfd091d7934f9bc82e4a36b95f647f090b9d1c9ac649da", size = 12028708, upload-time = "2026-07-22T22:19:02.164Z" },
    { url = "https://files.pythonhosted.org/packages/96/58/ad979ae617615576e8aafd569c9d4b62f1191d896e38f51d66ba06f3b89a/pandas-3.0.5-cp314-cp314-win_amd64.whl", hash = "sha256:cd8f7c6dc98527058ee6264219343f5392240a6f1bfa654fc5d79023020d0c92", size = 9951806, upload-time = "2026-07-22T22:19:04.596Z" },
    { url = "https://files.pythonhosted.org/packages/69/32/7ac03886b304049a9d2625ee88f59af760d8a93bd30ed9239bce7b9869a8/pandas-3.0.5-cp314-cp314-win_arm64.whl", hash = "sha256:5183427f5a8156d480f30333777bc978be93650a49a7c01db26adffe95b31e85", size = 9238297, upload-time = "2026-07-22T22:19:06.836Z" },
    { url = "https://files.pythonhosted.org/packages/be/ed/1d1f2ee5547d5167face2376d11c8b2a4c7bfff5a416ee7a9046891fab1e/pandas-3.0.5-cp314-cp314t-macosx_10_15_x86_64.whl", hash = "sha256:303da736987d481074ca720ada325f8bd80c64ebc2d45ed79b29df3aaa4a26ca", size = 10849690, upload-time = "2026-07-22T22:19:09.391Z" },
    { url = "https://files.pythonhosted.org/packages/57/55/17e17152e98fbb0c4b1e562bc65387a2f20a80db0f4a86bf8d3a0e4248d4/pandas-3.0.5-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:3b2801bbb049d0136f6c213eae02b5fca969384fc2064dd728d8620552aa49da", size = 10509945, upload-time = "2026-07-22T22:19:11.773Z" },
    { url = "https://files.pythonhosted.org/packages/88/90/817d44dbf83facf9556f33576d9af0a241981e7bb5c00606c0bcb5df8dda/pandas-3.0.5-cp314-cp314t-manylinux_2_24_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:cce3a9d11d2b1f82c69a27ec1f4948a170e2c403c4bbfa8cca62e3fdebe2ef3a", size = 10392197, upload-time = "2026-07-22T22:19:14.024Z" },
    { url = "https://files.pythonhosted.org/packages/f1/da/889f00c0a6f5aa1545add70abbf01502dff87ab577adb855bd631c54d2f2/pandas-3.0.5-cp314-cp314t-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:ef01af4d8dc6cd2c8d6c7736f149574ef93fe043811eeb5e445f2647154b5040", size = 10862726, uploaded-time = "2026-07-22T22:19:16.351Z" },
    { url = "https://files.pythonhosted.org/packages/bc/98/f1e934fb3c98fce859c6147c6785816c7b5b9ab7821115c5d8c4de9842b9/pandas-3.0.5-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:e2759e890db96dfcffdbd9b86c3c2cb6afaf58def482820317e06163ec1066cd", size = 11414864, upload-time = "2026-07-22T22:19:18.981Z" },
    { url = "https://files.pythonhosted.org/packages/fe/be/d448af7d657d82e1888dd8551f79c6d6fb161080b5b9752d84d910ec2319/pandas-3.0.5-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:b58b1b39d46a5862e3fb18f50d1a201398619d16a0f9f73f57eea5583cf0e63c", size = 11925105, upload-time = "2026-07-22T22:19:21.515Z" },
    { url = "https://files.pythonhosted.org/packages/29/c1/ccb4238212c8c4f496c584f3044d94e0c030ed8e1d68999db46c91c2242f/pandas-3.0.5-cp314-cp314t-win_amd64.whl", hash = "sha256:1c10461f6eeb35d8f05b6184c65c8b9991663b66c46b1d559b682cb34ae7c6ea", size = 10387612, upload-time = "2026-07-22T22:19:24.257Z" },
    { url = "https://files.pythonhosted.org/packages/d2/cf/6a51b2c38980e04c279fd2fa908a1b0982064e860444acfca4ec2e2c8359/pandas-3.0.5-cp314-cp314t-win_arm64.whl", hash = "sha256:3c5015fd1730fbf883647e88068176c839c102cea883ba1769a6f4593bfc1f8c", size = 9509776, upload-time = "2026-07-22T22:19:26.694Z" },
]

[[package]]
name = "pathspec"
version = "1.1.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/5a/82/42f767fc1c1143d6fd36efb827202a2d997a375e160a71eb2888a925aac1/pathspec-1.1.1.tar.gz", hash = "sha256:17db5ecd524104a120e173814c90367a96a98d07c45b2e10c2f3919fff91bf5a", size = 135180, upload-time = "2026-04-27T01:46:08.907Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/f1/d9/7fb5aa316bc299258e68c73ba3bddbc499654a07f151cba08f6153988714/pathspec-1.1.1-py3-none-any.whl", hash = "sha256:a00ce642f577bf7f473932318056212bc4f8bfdf53128c78bbd5af0b9b20b189", size = 57328, upload-time = "2026-04-27T01:46:07.06Z" },
]

[[package]]
name = "pgvector"
version = "0.5.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a7/ec/6eb80aebc728200f95229219882994c1b0585b956ca47da5edb9d062627a/pgvector-0.5.0.tar.gz", hash = "sha256:07a9dcf735696879406983afc6eba9a787cef7c0cf6c367ca1a5779f036dee74", size = 35170, upload-time = "2026-07-06T18:27:27.767Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/5c/e4/a5573f2c579ca9ad133293bfb624148ba0893674ca4a6eeec85ced9a6a09/pgvector-0.5.0-py3-none-any.whl", hash = "sha256:fedc9800894e6da2be51358d7b7c574bf34f247ca741a5a09513622135f5964f", size = 30958, upload-time = "2026-07-06T18:27:26.797Z" },
]

[[package]]
name = "pillow"
version = "12.3.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/1c/3d/bb7fca845737cf9d7dbde16ed1843984665ff2e0a518f5db43e77ec540b9/pillow-12.3.0.tar.gz", hash = "sha256:3b8182a766685eaa002637e28b4ec8d6b18819a0c71f579bf0dbaa5830297cce", size = 47025035, upload-time = "2026-07-01T11:56:38.965Z" }
```

```
wheels = [
    { url = "https://files.pythonhosted.org/packages/37/bf/fb3ebff8ddcb76aac5a01389251bbbb9519922a9b520d8247c1ca864a25d/pillow-12.3.0-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:ba09209fbe443b4acccebe845d8a138b89a8f4fbaeedd44953490b5315d5e965", size = 5345969, upload-time = "2026-07-01T11:54:06.397Z" },
    { url = "https://files.pythonhosted.org/packages/d8/66/9a386a92561f402389a4fc70c18838bf6d35eb5eb5c6850b4b2dc64f5048/pillow-12.3.0-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:ffd0c5368496f41b0944be820fcb7a838aa6e623d250b01acf2643939c3f99d7", size = 4780323, upload-time = "2026-07-01T11:54:09.351Z" },
    { url = "https://files.pythonhosted.org/packages/25/27/ac8f99618ffd3dde21db0f4d4b1d2ab00c0880595bfd17df103f7f39fd0c/pillow-12.3.0-cp312-cp312-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:d9c7f76c0673154f044e9d78c8655fb4213f6ca31a836df48b40fe5d187717b9", size = 6266838, upload-time = "2026-07-01T11:54:11.71Z" },
    { url = "https://files.pythonhosted.org/packages/84/21/a35af28dcc61f37ed850a2d64c65c701321dfbf25085e469d5559360cbbf/pillow-12.3.0-cp312-cp312-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:78cb2c6865a35ab8ff8b75fd122f6033b92a62c82801110e48ddd6c936a45d91", size = 6940830, upload-time = "2026-07-01T11:54:13.732Z" },
    { url = "https://files.pythonhosted.org/packages/eb/51/8b08617af3ad95e33ce6d7dd2c99ed6c8298f7fb131636303956be022e25/pillow-12.3.0-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:e491916b378fba47242221bb9ead245211b70d504f495d105d17b14a24b4907c", size = 6344383, upload-time = "2026-07-01T11:54:15.756Z" },
    { url = "https://files.pythonhosted.org/packages/1d/72/cf78ac9780bb93c28328f408973845a309d4d145041665f734572ced1b52/pillow-12.3.0-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:0dd2064cbc55aaec028ef5fbb60fa47bb6c3e7918e07ff17935284b227a9d2df", size = 7052934, upload-time = "2026-07-01T11:54:17.721Z" },
    { url = "https://files.pythonhosted.org/packages/20/20/25e0f4dc178a6bc069679372005551 9a0de89e7661dae886992decbd2f81/pillow-12.3.0-cp312-cp312-win32.whl", hash = "sha256:dbce0b29841537a2fa4a214c2bbf14de3587c9680caa9b4e217568472490b28f", size = 6472684, upload-time = "2026-07-01T11:54:19.839Z" },
    { url = "https://files.pythonhosted.org/packages/45/89/da2f7971a317f83d807fdd4065c0af40208e59e692cc43d315a71a0e96d1/pillow-12.3.0-cp312-cp312-win_amd64.whl", hash = "sha256:a2b55dd6b2a4c4b7d87ffa56bdb33fdc5fdb9a462173861a7bc097f17d91cb09", size = 7227137, upload-time = "2026-07-01T11:54:22.025Z" },
    { url = "https://files.pythonhosted.org/packages/de/47/4845a0a6c0dbf1db8456bd9fc791f13c5ced7ced20606d08a0aacfd25b49/pillow-12.3.0-cp312-cp312-win_arm64.whl", hash = "sha256:331b624368d4f1d069149002f25f44bc61c8919ce8ddb3c45bdad8f6e2d89510", size = 2568267, upload-time = "2026-07-01T11:54:24.051Z" },
    { url = "https://files.pythonhosted.org/packages/9d/ac/31fb64e1e7efb5a4b50cd3d92049ba89ac6e4d8d3bb6a74e15048ca3353e/pillow-12.3.0-cp313-cp313-ios_13_0_arm64_iphoneos.whl", hash = "sha256:21900ce7ba264168cd50defae43cd75d25c833ad4ad6e73ffc5596d12e25ac89", size = 4161684, upload-time = "2026-07-01T11:54:25.934Z" },
    { url = "https://files.pythonhosted.org/packages/87/b4/9805e23d2b4d77842b468513841fda254ee42f0289d25088340e4ff46e2d/pillow-12.3.0-cp313-cp313-ios_13_0_arm64_iphonesimulator.whl", hash = "sha256:4e8c2a84d977f50b9daed6eeaf3baef67d00d5d74d932288f02cb94518ee3ace", size = 4255487, upload-time = "2026-07-01T11:54:27.935Z" },
    { url = "https://files.pythonhosted.org/packages/df/39/ecf519435a200c693fe053a6ee4d835b41cf963a4dfc2551c4e637cb2a71/pillow-12.3.0-cp313-cp313-ios_13_0_x86_64_iphonesimulator.whl", hash = "sha256:ae26d61dfa7a47befdc7572b521024e8745f3d809bd95ca9505a7bba9ef849ec", size = 3696433, upload-time = "2026-07-01T11:54:29.813Z" },
    { url = "https://files.pythonhosted.org/packages/42/92/2fc3ffad878ae8dd5469ec1bc8eb83b71f48e13efdf68f02709003982a32/pillow-12.3.0-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:7a743ff716f746fc19a9557f60dab1600d4613255f8a7aeb3cdde4db7eb15a66", size = 5345889, upload-time = "2026-07-01T11:54:31.97Z" },
    { url = "https://files.pythonhosted.org/packages/10/76/8803c13605b763d33d156c4678fc77f8443389c0c51c8aef707bb02015f4/pillow-12.3.0-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:d69141514cc30b774ceea5e3ed3a6635c8d8a96edf664689b890f4089111fb35", size = 4780109, upload-time = "2026-07-01T11:54:34.026Z" },
    { url = "https://files.pythonhosted.org/packages/1f/01/e18aff37cb0b4aac47ac90f016d347a49aca667ef97f190b06ac2aabc928/pillow-12.3.0-cp313-cp313-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:f7401aebd7f581d7f83a439d87d474999317ee099218e5ad25d125290990ba65", size = 6263736, upload-time = "2026-07-01T11:54:36.131Z" },
    { url = "https://files.pythonhosted.org/packages/f7/62/de5bdd77d935331f4f802edc11e4d82950f642caad6cb2f949837b8560e2/pillow-12.3.0-cp313-cp313-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:0847a763afefb695bc912d7c131e7e0632d4edc1d8698f58ddabec8e46b8b6d3", size = 6937129, upload-time = "2026-07-01T11:54:38.216Z" },
    { url = "https://files.pythonhosted.org/packages/70/4d/105627a13300c5e0df1d174230b32fd1273062c96f7745fd552b945d1e1d/pillow-12.3.0-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:571b9fcb07b97ef3a492028fb3d2dc0993ca23a06138b0315286566d29ef718a", size = 6339562, upload-time = "2026-07-01T11:54:40.354Z" },
    { url = "https://files.pythonhosted.org/packages/6b/1d/f13de01a553988ab895ba1c722e06cf3144d4f57656fd5b81b6d881f1179/pillow-12.3.0-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:756c768d0c9c2955feb7a56c37ea24aea2e369f8d36a88da270b6a9f19e62b5e", size = 7049439, upload-time = "2026-07-01T11:54:42.489Z" },
    { url = "https://files.pythonhosted.org/packages/c9/f9/066794cca041b969964f779ee5fa66a9498bbf34248ac39c5d7954e4198f/pillow-12.3.0-cp313-cp313-win32.whl", hash = "sha256:a876864214e136f0eb367788dbd7df045f4806801518e2cfe9e13229cfe06d8f", size = 6473287, upload-time = "2026-07-01T11:54:44.9Z" },
    { url = "https://files.pythonhosted.org/packages/a6/9b/7a58e61d62be561da3a356fe2384d4059a6345fc130e23ef1c36a5b81d24/pillow-12.3.0-cp313-cp313-win_amd64.whl", hash = "sha256:1cca606cd25738df4ed873d5ad46bbdb3d83b5cbca291f6b4ff13a4df6b0bbe8", size = 7239691, upload-time = "2026-07-01T11:54:47.141Z" },
    { url = "https://files.pythonhosted.org/packages/aa/b0/c4ed4f0ef8f8fa5ee8351537db6650bb8189f7e118842978dd6589065692/pillow-12.3.0-cp313-cp313-win_arm64.whl", hash = "sha256:b629de27fda84b42cde7edef0d85f13b958b47f6e9bbcbba9b673c562a89bd8b", size = 2568185, upload-time = "2026-07-01T11:54:49.137Z" },
    { url = "https://files.pythonhosted.org/packages/dc/01/001f65b68192f0228cc1dbbc8d2530ab5d58b61037ba0587f946fea607cd/pillow-12.3.0-cp314-cp314-ios_13_0_arm64_iphoneos.whl", hash = "sha256:9cf95fe4d0f84c82d282745d9bb08ad9f926efa00be4697e767b814ce40d4330", size = 4161736, upload-time = "2026-07-01T11:54:51.156Z" },
    { url = "https://files.pythonhosted.org/packages/1a/d2/0219746d0fd16fc8a84498e79452375be3797d3ce4044596ce565164b84f/pillow-12.3.0-cp314-cp314-ios_13_0_arm64_iphonesimulator.whl", hash = "sha256:8728f216dcdb6e6d555cf971cb34076139ad74b31fc2c14da4fafc741c5f6217", size = 4255435, upload-time = "2026-07-01T11:54:53.414Z" },
    { url = "https://files.pythonhosted.org/packages/c8/02/8d0bc62ef0302318c46ff2a512822d2610e81c7aa46c9b3abe6cbaca5ad0/pillow-12.3.0-cp314-cp314-ios_13_0_x86_64_iphonesimulator.whl", hash = "sha256:a45650e8ce7fafffd731db8550230db6b0d306d181a90b67d3e6bca2f1990930", size = 3696262, upload-time = "2026-07-01T11:54:55.739Z" },
    { url = "https://files.pythonhosted.org/packages/85/e2/73c77d218410b14f5f2d565e8a998d5317b7b9c75368d29985139f7a46f0/pillow-12.3.0-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:ba54cfebe86920a559a7c4d6b9050791c20513650a1952ebe3368c7dc70306f8", size = 5350344, upload-time = "2026-07-01T11:54:57.657Z" },
    { url = "https://files.pythonhosted.org/packages/c7/da/32c752228ae345f489e3a42499d817b6c3996da7e8a3bc7a04fc806b243b/pillow-12.3.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:e158cb00350dc278f3b91551101aa7d12415a66ebf2c91d8d5ac14e56ddd3ad0", size = 4780131, upload-time = "2026-07-01T11:54:59.713Z" },
    { url = "https://files.pythonhosted.org/packages/b1/9d/8b2c807dbef61a5197c047afe99823787eb66f63daf9fb2432f91d6f0462/pillow-12.3.0-cp314-cp314-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:e9aeb04d6aef139de265b29683e119b638208f88cf73cdd1658aa07221165321", size = 6263757, upload-time = "2026-07-01T11:55:01.778Z" },
    { url = "https://files.pythonhosted.org/packages/5c/44/c85361f65dbe00eea8576ee467c768d25129989efb76e94f205e9ca9bb46/pillow-12.3.0-cp314-cp314-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:251bf95b67017e27b13d82f5b326234ca62d70f9cf4c2b9032de2358a3b12c7b", size = 6936962, upload-time = "2026-07-01T11:55:03.93Z" },
    { url = "https://files.pythonhosted.org/packages/18/7e/e483414b35800b86b6f08dbbc7803fb5cd52c4d6f897f47d53ea2c7e6f65/pillow-12.3.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:fe3cca2e4e8a592be0f269a1ca4835c25199d9f3ce815c8491048f785b0a0198", size = 6339171, upload-time = "2026-07-01T11:55:05.989Z" },
    { url = "https://files.pythonhosted.org/packages/f0/f4/68c491844841ede6bed70189546b3ee9731cf9f2cbad396faff5e1ccba45/pillow-12.3.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:23aceaa007d6172b02c277f0cd359c79492bbb14f7072b4ede9fbcaf20648130", size = 7048116, upload-time = "2026-07-01T11:55:08.131Z" },
    { url = "https://files.pythonhosted.org/packages/a3/34/77f3f793fed8efc7d243f21b33c5a3f0d1c97ee70346d3db855587e155ff/pillow-12.3.0-cp314-cp314-win32.whl", hash = "sha256:af8d94b0db561cf68b88a267c5c44b49e134f525d0dc2cb7ed413a66bc23559a", size = 6467209, upload-time = "2026-07-01T11:55:10.408Z" },
    { url = "https://files.pythonhosted.org/packages/f1/e0/492879f69d94f91f60fc8cd05ba03650e9520afebb2fb7aa12777d7c7f38/pillow-12.3.0-cp314-cp314-win_amd64.whl", hash = "sha256:fdafc9cce40277e0f7a0feabce0ee50dd2fa1800f3b38015e51296b5e814048d", size = 7237707, upload-time = "2026-07-01T11:55:12.745Z" },
    { url = "https://files.pythonhosted.org/packages/c9/ac/6b11f2875f1c2ac040d84e1bbf9cf22a88038f901ca1037898b280b38365/pillow-12.3.0-cp314-cp314-win_arm64.whl", hash = "sha256:e91206ee562682b51b98ef4b26a6ef48fd84e15fd4c4bc5ec768eb641d206838", size = 2565995, upload-time = "2026-07-01T11:55:14.736Z" },
    { url = "https://files.pythonhosted.org/packages/52/69/c2208e56af9bfc1913afb24020297a691eb1d4ef688474c8a04913f65e04/pillow-12.3.0-cp314-cp314t-macosx_10_15_x86_64.whl", hash = "sha256:164b31cd1a0490ab6efae01aa5df49da7061be0af1b30e035b6e9a1bfe34ee6e", size = 5352503, upload-time = "2026-07-01T11:55:17.076Z" },
    { url = "https://files.pythonhosted.org/packages/07/70/e5686d753e898a45d778ff1718dba8516ead6ab6b95d85fc8c4b70650cf2/pillow-12.3.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:5afb51d599ea772b8365ae807ae557f18bccfe46ab261fd1c2a9ed700fc6eb17", size = 4782956, upload-time = "2026-07-01T11:55:19.448Z" },
    { url = "https://files.pythonhosted.org/packages/d5/37/25c6692f06927ee973ff18c8d9ee98ad0b4d84ee67a09610c2dd1447958e/pillow-12.3.0-cp314-cp314t-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:3edce1d53195db527e0191f84b71d02022de0540bf43a16ed734ed7537b07385", size = 6322855, upl
```

```
oad-time = "2026-07-01T11:55:21.613Z" },
    { url = "https://files.pythonhosted.org/packages/cc/91/420637fcb8f1bc11029e403b4538e6694744428d8246118e45719f944556/pillow-12.3.0-cp314-cp314t-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:bf16ba1b4d0b6b7c8e534936632270cf70eb00dbe09005bc345b2677b726855c", size = 6989642, upload-time = "2026-07-01T11:55:24.006Z" },
    { url = "https://files.pythonhosted.org/packages/10/08/b94d7811281ccf0d143a1cf768d1c49e1e54af63e7b708ab2ee3eb87face/pillow-12.3.0-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:24870b09b224f7ae3c39ed07d10e819d06f8720bc551847b1d623832b5b0e28d", size = 6391281, upload-time = "2026-07-01T11:55:26.252Z" },
    { url = "https://files.pythonhosted.org/packages/d2/87/24233f785f55474dc02ce3e739c5528a77e3a862e9333d1dd7a25cc31f70/pillow-12.3.0-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:30f2aa603c41533cc25c05acd0da21636e84a315768feb631c937177db558931", size = 7096716, upload-time = "2026-07-01T11:55:28.318Z" },
    { url = "https://files.pythonhosted.org/packages/23/26/fcb2f6e37175b04f53570b59937867e2b80ee1685e744023153028fc14f9/pillow-12.3.0-cp314-cp314t-win32.whl", hash = "sha256:4b0a7fe987b14c31ebda6083f74f22b561fd3739bc0ac51e019622e3d72668c7", size = 6474125, upload-time = "2026-07-01T11:55:30.956Z" },
    { url = "https://files.pythonhosted.org/packages/90/de/3634abee5f1c9e13c56787b7d5517b0ba8d6de51700b95578cf338349c9f/pillow-12.3.0-cp314-cp314t-win_amd64.whl", hash = "sha256:962864dc93511324d51ddbb5b9f8731bf71675b93ca612a07441896f4688fb8c", size = 7242939, upload-time = "2026-07-01T11:55:34.044Z" },
    { url = "https://files.pythonhosted.org/packages/ce/2a/fd13f8eb24de5714a6eb444a3d67e2842c6c576e159a43793adf23051351/pillow-12.3.0-cp314-cp314t-win_arm64.whl", hash = "sha256:0740a512dc522224c77d9aa5a8d70d8b7d73fb91f2c21125d8d025d3b8990e45", size = 2567506, upload-time = "2026-07-01T11:55:35.988Z" },
    { url = "https://files.pythonhosted.org/packages/5d/dc/8fdce34ec725a33c81c6ba122b904d6b9024e50ea9ac7bede62fab54506c/pillow-12.3.0-cp315-cp315-ios_13_0_arm64_iphoneos.whl", hash = "sha256:0feb2e9d6ad6c9e3c06effe9d00f3f1e618a6643273576b016f591e9315a7139", size = 4162063, upload-time = "2026-07-01T11:55:37.941Z" },
    { url = "https://files.pythonhosted.org/packages/76/66/2044b9a63d3b84ff048228dfcb7cd9bf0df983e8470971bf7d4c57b693de/pillow-12.3.0-cp315-cp315-ios_13_0_arm64_iphonesimulator.whl", hash = "sha256:9e881fca225083806662a5c43d627d215f258ff43c890f831966c7d7ba9c7402", size = 4255549, upload-time = "2026-07-01T11:55:40.022Z" },
    { url = "https://files.pythonhosted.org/packages/52/7e/1f67e6f4ece6b582ee4b539decbcc9f848dc245a93ed8cd7338bafef72f1/pillow-12.3.0-cp315-cp315-ios_13_0_x86_64_iphonesimulator.whl", hash = "sha256:4998562bf62a445225f22e07c896bb04b35b1b1f2eb6d760584c9c51d7a5f78c", size = 3696331, upload-time = "2026-07-01T11:55:41.98Z" },
    { url = "https://files.pythonhosted.org/packages/12/40/d306fc2c8e4d45d7f175c77edca7063be7b86fe7fe6e68f4353bf71d808c/pillow-12.3.0-cp315-cp315-macosx_10_15_x86_64.whl", hash = "sha256:dc624f6bc473dacdf7ef7eb8678d0d08edf15cd94fad6ae5c7d6cc67a4e4902f", size = 5350370, upload-time = "2026-07-01T11:55:44.028Z" },
    { url = "https://files.pythonhosted.org/packages/dd/44/668fb1437e8ce420f62d6106eb66e44a5971602a4d794615bdf79315d82d/pillow-12.3.0-cp315-cp315-macosx_11_0_arm64.whl", hash = "sha256:71d6097b330eea8fd15097780c8e89cb1a8ce7838669f48c5bacd6f663dd4701", size = 4780147, upload-time = "2026-07-01T11:55:46.073Z" },
    { url = "https://files.pythonhosted.org/packages/0c/08/93fa2e70e30a2d81547e481b6ee2bb9522117221fb1e0ce4b5df70967677/pillow-12.3.0-cp315-cp315-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:28ce87c5ab450a9dd970b52e5aca5fe63ed432d18a2eaddd1979a00a1ba24ace", size = 6273659, upload-time = "2026-07-01T11:55:48.264Z" },
    { url = "https://files.pythonhosted.org/packages/f8/6d/043e96ff814fc31a33077e4cba86082167db520c93632afdf2042febbb0c/pillow-12.3.0-cp315-cp315-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:6b02afb9b97f65fbca5f31db6a2a3ba21aa93030225f150fa3f249717e938fb4", size = 6947439, upload-time = "2026-07-01T11:55:50.503Z" },
    { url = "https://files.pythonhosted.org/packages/af/92/ba71d2ee2ac0edf3fa33bd9d5ee9ee080da70b1766f3ca3934f9938ddac9/pillow-12.3.0-cp315-cp315-musllinux_1_2_aarch64.whl", hash = "sha256:1182d52bc2d5e5d7d0949503aa7e36d12f42205dc287e4883f407b1988820d39", size = 6353577, upload-time = "2026-07-01T11:55:52.697Z" },
    { url = "https://files.pythonhosted.org/packages/0f/ce/e63064e2122923ff687c8ad792d0d736a7b3920a56a46982e81a7fdd25d6/pillow-12.3.0-cp315-cp315-musllinux_1_2_x86_64.whl", hash = "sha256:e795b7eb908249c4e43c7c99fac7c2c75dab0c43566e37db472a355f63693d71", size = 7060394, upload-time = "2026-07-01T11:55:55.149Z" },
    { url = "https://files.pythonhosted.org/packages/54/76/a09cc3ccc8d773a7283d34c38bec1708f9e3cc932093cbc4c5e71ac4060b/pillow-12.3.0-cp315-cp315-win32.whl", hash = "sha256:57b3d78c95ba9059768b10e28b813002261d3f3dfc55cc48b0c988f625175827", size = 6467375, upload-time = "2026-07-01T11:55:57.769Z" },
    { url = "https://files.pythonhosted.org/packages/3e/03/1846c49ba3b1d5550392a4bbd06d6fb4578e1cd91a803198b5c90f5f7d53/pillow-12.3.0-cp315-cp315-win_amd64.whl", hash = "sha256:fa4ecea169a355be7a3ade2c783e2ed12f0e40d2c5621cda8b3297faf7fbb9f5", size = 7237048, upload-time = "2026-07-01T11:55:59.975Z" },
    { url = "https://files.pythonhosted.org/packages/fb/bb/89f35dcc79610423f9f195504d7def7f0d1416a711541b42867e25fe3412/pillow-12.3.0-cp315-cp315-win_arm64.whl", hash = "sha256:877c3f311ff35410f690861c4409e7ccbf0cd2f878e50628a28e5a0bb689e658", size = 2566006, upload-time = "2026-07-01T11:56:02.143Z" },
    { url = "https://files.pythonhosted.org/packages/30/88/707027ba09942dfa2c28759b5c222d769290a41c6d20ea60ec250801941f/pillow-12.3.0-cp315-cp315t-macosx_10_15_x86_64.whl", hash = "sha256:e9871b1ffbfa9656b60aeee92ed5136a5742696006fa322b29ea3d8da0ecc9cf", size = 5352509, upload-time = "2026-07-01T11:56:04.2Z" },
    { url = "https://files.pythonhosted.org/packages/b0/6d/00352fa25332c2569cd387851f568cc5a4b75a9adbfb37ac4fbce4c02eec/pillow-12.3.0-cp315-cp315t-macosx_11_0_arm64.whl", hash = "sha256:53aa02d20d10c3d814d536aa4e5ac9b84ca0ff5a88377963b085ad6822f93e64", size = 4783167, upload-time = "2026-07-01T11:56:06.631Z" },
    { url = "https://files.pythonhosted.org/packages/13/4f/9e049dfa21af7c22427275720e2490267ba8138120add5c4c574deb69782/pillow-12.3.0-cp315-cp315t-manylinux_2_27_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:446c34dcc4324b084a53b705127dc15717b22c5e140ae0a3c38349d4efec071e", size = 6329237, upload-time = "2026-07-01T11:56:08.868Z" },
    { url = "https://files.pythonhosted.org/packages/36/16/cf6eeaae8d0fce8dd390a33437cf68c5d5bd73834a2bc6e2f14efda0ab45/pillow-12.3.0-cp315-cp315t-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:cf1845d02ad822a369a49f2bb9345b1614744267682e7a03527dc3bf6eea1777", size = 6997047, upload-time = "2026-07-01T11:56:11.379Z" },
    { url = "https://files.pythonhosted.org/packages/1e/69/dbf769bdd55f48bf5733cac28edc6364ffaa072ec9ba336266e4fe66be55/pillow-12.3.0-cp315-cp315t-musllinux_1_2_aarch64.whl", hash = "sha256:186941b6aef820ad110fb01fb06eb925374dc3a21b17e37ec9a53b250c6fe2d1", size = 6400440, upload-time = "2026-07-01T11:56:13.908Z" },
    { url = "https://files.pythonhosted.org/packages/a0/e1/ffc9cfc2eea0d178da8018e18e959301ad9d6bc9f3edb7181e748a474b97/pillow-12.3.0-cp315-cp315t-musllinux_1_2_x86_64.whl", hash = "sha256:f13c32a3abd6079a66d9526e18dad9b6d280384d49d7c54040cd57b6424041d9", size = 7105895, upload-time = "2026-07-01T11:56:16.575Z" },
    { url = "https://files.pythonhosted.org/packages/18/f0/a5595c1e8c3ae44b9828cb2f0fa8155e5095ef04d6327b8f61cf44a3df85/pillow-12.3.0-cp315-cp315t-win32.whl", hash = "sha256:1657923d2d45afb66526e5b933e5b3052e6bdea196c90d3abb2424e18c77dae8", size = 6474384, upload-time = "2026-07-01T11:56:18.855Z" },
    { url = "https://files.pythonhosted.org/packages/e4/04/62bcd9f844984c5938d3b05264a61d797a29d3e0812341a8204af70bbdee/pillow-12.3.0-cp315-cp315t-win_amd64.whl", hash = "sha256:8cd2f7bdda092d99c9fc2fb7391354f306d01443d22785d0cbfafa2e2c8bb418", size = 7243537, upload-time = "2026-07-01T11:56:21.214Z" },
    { url = "https://files.pythonhosted.org/packages/3d/68/1f3066acedf37673694a7141381d8f811ae97f30d34413d236abe7d489f1/pillow-12.3.0-cp315-cp315t-win_arm64.whl", hash = "sha256:06ff022112bc9cbf83b60f8e028d94ad87b60621706487e65f673de61610ab59", size = 2567491, upload-time = "2026-07-01T11:56:23.506Z" },
]

[[package]]
name = "platformdirs"
version = "4.11.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/78/9b/560e4be8e26f6fd133a03630a8df0c663b9e8d61b4ade152b72005aec83b/platformdirs-4.11.0.tar.gz", hash = "sha256:0555d18370482847566ffabcaa53ad7c6c1c29f195989ae1ed634a05f76ea1e0", size = 31953, upload-time = "2026-07-21T13:09:36.565Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/7d/68/d8d58938dfb1370b266a1a729e6d77a985be23689a0496498ee17b2cbf90/platformdirs-4.11.0-py3-none-any.whl", hash = "sha256:360ccded2b7fce0af0ff80cc8f5942a1c5d99b0e856033acb030bfc634709e74", size = 23247, upload-time = "2026-07-21T13:09:35.422Z" },
]

[[package]]
name = "pluggy"
version = "1.6.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/f9/e2/3e91f31a7d2b083fe6ef3fa267035b518369d9511ffab804f839851d2779/pluggy-1.6.0.tar.gz", hash = "sha256:7dcc130b76258d33b90f61b658791dede3486c3e6bfb003ee5c9bfb396dd22f3", size = 69412, upload-time = "2025-05-15T12:30:07.975Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/54/20/4d324d65cc6d9205fabedc306948156824eb9f0ee1633355a8f7ec5c66bf/pluggy-1.6.0-py3-none-any.whl", hash = "sha256:e920276dd6813095e9377c0bc5566d94c932c33b27a3e3945d8389c374dd4746", size = 20538, upload-time = "2025-05-15T12:30:06.134Z" },
]

[[package]]
name = "pre-commit"
```

```
version = "4.6.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "cfgv" },
    { name = "identify" },
    { name = "nodeenv" },
    { name = "pyyaml" },
    { name = "virtualenv" },
]
sdist = { url = "https://files.pythonhosted.org/packages/25/3a/ddb78f32a0814e66b18a099377a106a2dcdce92d86a034d69d65df9b256e/pre_commit-4.6.1.tar.gz",
hash = "sha256:03e809865c7d178b9979d06c761fcbfe6808fdaded8581a745bb110e52050421", size = 198646, upload-time = "2026-07-21T20:56:58.225Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/fb/49/bc925106abcdac498074f2cbe6137e94e09f418dd2b7775df5b577dc0313/pre_commit-4.6.1-py2.py3-none-
any.whl", hash = "sha256:0e3b2942510d1fb34eec167a3ec57331bf8442122f1153a9fb8b58f5c49b2717", size = 226186, upload-time = "2026-07-21T20:56:57.064Z" },
]

[[package]]
name = "prometheus-client"
version = "0.26.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/52/73/f1334c29c2af4cd9dba6c7817e61b611bd0215e2eb5565c6064a4de18802/prometheus_client-0.26.0.t
ar.gz", hash = "sha256:04a91bcf94e2cf74a44a1a874d651a2e853ed354b6e822f3b7487751465d5c2b", size = 92910, upload-time = "2026-07-24T19:36:41.893Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/eb/a3/b69efbf4143b5b9859b977770bbbabcc2796b702fa69dc40271e45cd5a56/prometheus_client-0.26.0-py3-n
one-any.whl", hash = "sha256:fa93d06737aa02bacd05794768508bb97d2fbee28cb3bca04eaae92f0ca953d6", size = 64494, upload-time = "2026-07-24T19:36:40.854Z"
 },
]

[[package]]
name = "propcache"
version = "0.5.2"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/ec/44/c87281c333769159c50594f22610f77398a47ccbfbbf23074e744e86f87c/propcache-0.5.2.tar.gz", h
ash = "sha256:01c4fc7480cd0598bb4b57022df55b9ca296da7fc5a8760bd8451a7e63a7d427", size = 50208, upload-time = "2026-05-08T21:02:12.199Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/4a/cb/e27bc2b2737a0bb49962b275efa051e8f1c35a936df7d5139b6b658b7dc9/propcache-0.5.2-cp312-cp312-ma
cosx_10_13_universal2.whl", hash = "sha256:806719138ecd720339a12410fb9614ac9b2b2d3a5fdf8235d56981c36f4039ba", size = 95887, upload-time = "2026-05-08T
21:00:11.277Z" },
    { url = "https://files.pythonhosted.org/packages/e6/13/b8ae04c59392f8d11c6cd9fb4011d1dc7c86b81225c770280300e259ffe1/propcache-0.5.2-cp312-cp312-ma
cosx_10_13_x86_64.whl", hash = "sha256:db2b80ea58eab4f86b2beec3cc8b39e8ff9276ac20e96b7cce43c8ae84cd6b5a", size = 54654, upload-time = "2026-05-08T21:0
0:12.604Z" },
    { url = "https://files.pythonhosted.org/packages/2c/7d/49777a3e20b55863d4794384a38acd460c04157b0a00f8602b0d508b8431/propcache-0.5.2-cp312-cp312-ma
cosx_11_0_arm64.whl", hash = "sha256:e5cbfac9f61484f7e9f3597775500cd3ebe8274e9b050c38f9525c77c97520bf", size = 55190, upload-time = "2026-05-08T21:00:
13.935Z" },
    { url = "https://files.pythonhosted.org/packages/44/c7/085d0cd63062e84044e3f05797749c3f8e3938ff3aeb0eb2f69d43fafc91/propcache-0.5.2-cp312-cp312-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:5dbc581d2814337da56222fab8dc5f161cd798a434e49bac27930aaef798e14
4", size = 59995, upload-time = "2026-05-08T21:00:15.526Z" },
    { url = "https://files.pythonhosted.org/packages/9c/42/32cf8e3009e92b2645cf1e944f701e8ea4e924dffde1ee26db860bcbf7e4/propcache-0.5.2-cp312-cp312-ma
nylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:857187f381f88c8e2fa2fe56ab94879d011b883d5a2ee5a1b60a8cd2a06846d
9", size = 63422, upload-time = "2026-05-08T21:00:16.824Z" },
    { url = "https://files.pythonhosted.org/packages/9e/1b/f112433f99fc979431b87a39ef169e3f8df070d99a72792c56d6937ac48b/propcache-0.5.2-cp312-cp312-ma
nylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:178b4a2cdaac1818e2bf1c5a99b94383fa73ea5382e032a48dec07dc5668dc42", si
ze = 64342, upload-time = "2026-05-08T21:00:18.362Z" },
    { url = "https://files.pythonhosted.org/packages/14/15/5574111ae50dd6e879456888c0eadd4c5a869959775854e18e18a6b345f3/propcache-0.5.2-cp312-cp312-ma
nylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:6f328175a2cde1f0ff2c4ed8ce968b9dcfb55f3a7153f39e2957ed994da13476",
 size = 61639, upload-time = "2026-05-08T21:00:19.692Z" },
    { url = "https://files.pythonhosted.org/packages/cc/da/4d775080b1490c0ae604acda868bd71aabe3a89ed16f2aa4339eb8a283e7/propcache-0.5.2-cp312-cp312-ma
nylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:5671d09a36b06d0fd4a3da0fccbcae360e9b1570924171a15e9e0997f0249fba", size = 61588, uplo
ad-time = "2026-05-08T21:00:21.155Z" },
    { url = "https://files.pythonhosted.org/packages/04/ac/f076982cbe2195ee9cf32de5a1e46951d9fb399fc207f390562dd0fd8fb2/propcache-0.5.2-cp312-cp312-mu
sllinux_1_2_aarch64.whl", hash = "sha256:80168e2ebe4d3ec6599d10ad8f520304ae1cad9b6c5a95372aef1b66b7bfb53a", size = 60029, upload-time = "2026-05-08T21
:00:22.713Z" },
    { url = "https://files.pythonhosted.org/packages/70/60/189be62e0dd898dce3b331e1b8c7a543cd3a405ac0c81fe8ee8a9d5d77e1/propcache-0.5.2-cp312-cp312-mu
sllinux_1_2_armv7l.whl", hash = "sha256:45f11346f884bc47444f6e6647131055844134c3175b629f84952e2b5cd62b64", size = 56774, upload-time = "2026-05-08T21:
00:24.001Z" },
    { url = "https://files.pythonhosted.org/packages/ea/9e/93377b9c7939c1ffae98f878dee955efadfd638078bc86dbc21f9d52f651/propcache-0.5.2-cp312-cp312-mu
sllinux_1_2_ppc64le.whl", hash = "sha256:8e778ebd44ef4f66ed60a0416b06b489687db264a9c0b3620362f26489492913", size = 63532, upload-time = "2026-05-08T21
:00:25.545Z" },
    { url = "https://files.pythonhosted.org/packages/14/f9/590ef6cfb9b8028d516d287812ece32bb0bc5f11fbb9c8bf6b2e6313fec8/propcache-0.5.2-cp312-cp312-mu
sllinux_1_2_riscv64.whl", hash = "sha256:c0cb9ed24c8964e172768d455a38254c2dd8a552905729ce006cad3d3dda59b1", size = 61592, upload-time = "2026-05-08T21
:00:27.186Z" },
    { url = "https://files.pythonhosted.org/packages/b4/5e/70958b3034c297a630bba2f17ca7abc2d5f39a803ad7e370ab79d1ecd022/propcache-0.5.2-cp312-cp312-mu
sllinux_1_2_s390x.whl", hash = "sha256:1d1ad32d9d4355e2be65574fd0bfd3677e7066b009cd5b9b2dee8aa6a6393b33", size = 64788, upload-time = "2026-05-08T21:0
0:28.8Z" },
    { url = "https://files.pythonhosted.org/packages/12/fd/77fe5936d8c3086ca9048f7f415f122ed82e53884a9ec193646b42deef06/propcache-0.5.2-cp312-cp312-mu
sllinux_1_2_x86_64.whl", hash = "sha256:c80f4ba3e8f00189165999a742ee526ebeccedf6c3f7beb0c7df821e9772435a", size = 62514, upload-time = "2026-05-08T21:
00:30.098Z" },
    { url = "https://files.pythonhosted.org/packages/cf/74/66bd798b5b3be70aa1b391f5cc9d6a0a5532d7fd3b19ec0b213e72e6ad9d/propcache-0.5.2-cp312-cp312-wi
n32.whl", hash = "sha256:8c7972d8f193740d9175f0998ab38717e6cd322d5935c5b0fef8c0d323fd9031", size = 39018, upload-time = "2026-05-08T21:00:31.622Z" },
    { url = "https://files.pythonhosted.org/packages/61/7c/5c0d34aa3024694d6dcb9271cdbdd08c4e47c1c0ad95ec7e7bc74cdea145/propcache-0.5.2-cp312-cp312-wi
n_amd64.whl", hash = "sha256:d9ee8826a7d47863a08ac44e1a5f611a462eefc3a194b492da242128bec75b42", size = 42322, upload-time = "2026-05-08T21:00:32.918Z"
 },
    { url = "https://files.pythonhosted.org/packages/4d/91/875812f1a3feb20ceba818ef39fbe4d92f1081e04ac815c822496d0d038b/propcache-0.5.2-cp312-cp312-wi
n_arm64.whl", hash = "sha256:2800a4a8ead6b28cccd1ec54b59346f0def7922ee1c7598e8499c733cfbb7c84", size = 38172, upload-time = "2026-05-08T21:00:35.124Z"
 },
    { url = "https://files.pythonhosted.org/packages/c5/09/f049e45385503fe67db75a6b6186a7b9f0c3930366dc960522c312a825b1/propcache-0.5.2-cp313-cp313-ma
cosx_10_13_universal2.whl", hash = "sha256:099aaf4b4d1a02265b92a977edf00b5c4f63b3b17ac6de39b0d637c9cac0188a", size = 94457, upload-time = "2026-05-08T
21:00:36.355Z" },
    { url = "https://files.pythonhosted.org/packages/6b/65/83d1d05655baf63113731bd5a1008435e14f8d1e5a06cbe4ec5b23ad7a31/propcache-0.5.2-cp313-cp313-ma
cosx_10_13_x86_64.whl", hash = "sha256:68ce1c44c7a813a7f71ea04315a8c7b330b63db99d059a797a4651bb6f69f117", size = 53835, upload-time = "2026-05-08T21:0
0:38.072Z" },
    { url = "https://files.pythonhosted.org/packages/a9/12/a6ba6482bb5ea3260c000c9b20881c95fa11c6b30173715668259f844ed7/propcache-0.5.2-cp313-cp313-ma
cosx_11_0_arm64.whl", hash = "sha256:fc299c129490f55f254cd90be0deca4764e36e9a7c08b4aa588479a3bbed3098", size = 54545, upload-time = "2026-05-08T21:00:
39.319Z" },
    { url = "https://files.pythonhosted.org/packages/a9/19/7fa086f5764c59ec8a8e157cd93aa8497acc00aba9dcdec56bfffb32602d/propcache-0.5.2-cp313-cp313-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:a6ae2198be502c10f09b2516e7b5d019816924bc3183a43ce792a7bd6625e6f
4", size = 59886, upload-time = "2026-05-08T21:00:40.621Z" },
    { url = "https://files.pythonhosted.org/packages/a1/e4/5d7663dc8235956c8f5281698a3af1d351d8820341ddd890f59d9a9127f2/propcache-0.5.2-cp313-cp313-ma
nylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:6041d31504dc1779d700e1edcfb08eea334b357620b06681a4eabb57a74e574
e", size = 63261, upload-time = "2026-05-08T21:00:41.775Z" },
    { url = "https://files.pythonhosted.org/packages/4a/4a/15a03adee24d6350da4292caeac44c34c033d2afe5e87eb370f38854560f/propcache-0.5.2-cp313-cp313-ma
nylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:f7eabc04151c78a9f4d5bbb5f1faf571e4defeb4b585e0fe95b60ff2dbe4d3d7", si
ze = 64184, upload-time = "2026-05-08T21:00:43.018Z" },
    { url = "https://files.pythonhosted.org/packages/8b/c6/979176efdaa3d239e36d503d5af63a0a773b36662ed8f52e5b6a6d9fd40e/propcache-0.5.2-cp313-cp313-ma
nylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:4db0ba63d693afd40d249bd93f842b5f144f8fcbb83de05660373bcf30517b1d",
```

```
 size = 61534, upload-time = "2026-05-08T21:00:44.507Z" },
    { url = "https://files.pythonhosted.org/packages/c8/22/63e8cd1bae4c2d2be6493b6b7d10566ddafad88137cfbc99964a1119853c/propcache-0.5.2-cp313-cp313-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:1dbcf7675229b35d31abb6547d8ebc8c27a830ac3f9a794edff6254873ec7c0a", size = 61500, upload-time = "2026-05-08T21:00:45.796Z" },
    { url = "https://files.pythonhosted.org/packages/60/5a/28e5d9acbac1cc9ccb67045e8c1b943aa8d79fdf39c93bd73cacd68008ea/propcache-0.5.2-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:d310c013aad2c72f1c3f2f8dd3279d460a858c551f97aeb8c63e4693cca7b4d2", size = 59994, upload-time = "2026-05-08T21:00:47.093Z" },
    { url = "https://files.pythonhosted.org/packages/f3/40/db650677f554a95b9c01a7c9d93d629e93a15562f5deb4573c9ee136fed2/propcache-0.5.2-cp313-cp313-musllinux_1_2_armv7l.whl", hash = "sha256:06187263ddad280d05b4d8a8b3bb7d164cbebd469236544a42e6d9b28ac6a4fa", size = 56884, upload-time = "2026-05-08T21:00:48.376Z" },
    { url = "https://files.pythonhosted.org/packages/80/45/70b39b89516ff8b96bf732fa6fded8cef20f293cb1508690101c3c07ec51/propcache-0.5.2-cp313-cp313-musllinux_1_2_ppc64le.whl", hash = "sha256:3115559b8effafd63b142ea5ed53d63a16ea6469cbc63dce4ee194b42db5d853", size = 63464, upload-time = "2026-05-08T21:00:49.954Z" },
    { url = "https://files.pythonhosted.org/packages/f9/e2/fa59d3a89eac5534293124af4f1d0d0ada091ce4a0ab4610ce03fd2bdd8d/propcache-0.5.2-cp313-cp313-musllinux_1_2_riscv64.whl", hash = "sha256:c60462af8e6dc30c35407c7237ea908d777b22862bbee27bc4699c0d8bcdc45a", size = 61588, upload-time = "2026-05-08T21:00:51.281Z" },
    { url = "https://files.pythonhosted.org/packages/0b/97/efb547a55c4bc7381cfb202d6a2239ac621045277bc1ea5dfd3a7f0516c0/propcache-0.5.2-cp313-cp313-musllinux_1_2_s390x.whl", hash = "sha256:40314bca9ac559716fe374094fc81c11dcc34b64fd6c585360f577569050570 4", size = 64667, upload-time = "2026-05-08T21:00:52.602Z" },
    { url = "https://files.pythonhosted.org/packages/92/56/f5c7d9b4b7595d5127da38974d791b2153f3d1eae6c674af3583ace92ad3/propcache-0.5.2-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:cfa21e036ce1e1db2be04ba3b85d2df1bb1702fa01932d984c5464c665228ff4", size = 62463, upload-time = "2026-05-08T21:00:54.303Z" },
    { url = "https://files.pythonhosted.org/packages/bd/3b/484a3a65fc9f9f60c41dcd17b428bace5389544e2c680994534a20755066/propcache-0.5.2-cp313-cp313-win32.whl", hash = "sha256:f156a3529f38063b6dbaf356e15602a7f95f8055b1295a438433a6386f10463d", size = 38621, upload-time = "2026-05-08T21:00:55.808Z" },
    { url = "https://files.pythonhosted.org/packages/1c/fd/3f0f10dba4dabad3bf53102be007abf55481067952bde0fdddff439e7c61/propcache-0.5.2-cp313-cp313-win_amd64.whl", hash = "sha256:dfed59d0a5aeb01e242e66ff0300bc4a265a7c05f612d30016f0b60b1017d757", size = 41649, upload-time = "2026-05-08T21:00:57.061Z" },
    { url = "https://files.pythonhosted.org/packages/90/ec/6ce619cc32bb500a482f811f9cd509368b4e58e638d13f2c68f370d6b475/propcache-0.5.2-cp313-cp313-win_arm64.whl", hash = "sha256:ba338430e87ceb9c8f0cf754de38a9860560261e56c00376debd628698a7364f", size = 37636, upload-time = "2026-05-08T21:00:58.646Z" },
    { url = "https://files.pythonhosted.org/packages/1b/82/c1d268bbbf2ef981c5bf0fbbe746db617c66e3bcefe431a1aa8943fbe23a/propcache-0.5.2-cp313-cp313t-macosx_10_13_universal2.whl", hash = "sha256:a592f5f3da71c8691c788c13cb6734b6d17663d2e1cb8caddf0673d01ef8847d", size = 98872, upload-time = "2026-05-08T21:00:59.889Z" },
    { url = "https://files.pythonhosted.org/packages/f4/d4/52c871e73e864e6b34c0e2d58ac1ec5ccd149497ddc7ad2137ae98323a35/propcache-0.5.2-cp313-cp313t-macosx_10_13_x86_64.whl", hash = "sha256:6a997d0489e9668a384fcfd5061b857aa5361de73191cac204d04b889cfbbafa", size = 56257, upload-time = "2026-05-08T21:01:01.195Z" },
    { url = "https://files.pythonhosted.org/packages/67/f0/9b90ca2a210b3d09bcfcd96ecd0f55545c091535abce2a45de2775cfd357/propcache-0.5.2-cp313-cp313t-macosx_11_0_arm64.whl", hash = "sha256:10734b5484ea113152ee25a91dccedf81631791805d2c9ccb054958e51842c94", size = 56696, upload-time = "2026-05-08T21:01:02.941Z" },
    { url = "https://files.pythonhosted.org/packages/9d/0e/6e9d4ba07c8e56e21ddec1e75f12148142b21ca83a51871babce095334f4/propcache-0.5.2-cp313-cp313t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:cafca7e56c12bb02ae16d283742bef25a61122e9dab2b5b3f2ccbe589ce32164", size = 62378, upload-time = "2026-05-08T21:01:04.475Z" },
    { url = "https://files.pythonhosted.org/packages/65/19/c10badaa463dde8a27ce884f8ee2ec37e6035b7c9f5ff0c8f74f06f08dac/propcache-0.5.2-cp313-cp313t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:f064f8d2b59177878b7615df1735cd8fe3462ed6be8c7b217d17a276489c2b7f", size = 65283, upload-time = "2026-05-08T21:01:05.959Z" },
    { url = "https://files.pythonhosted.org/packages/b0/b6/93bea99ca80e19cef6512a8580e5b7857bbe09422d9daa7fd4ef5723306c/propcache-0.5.2-cp313-cp313t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:f78abfa8dfc32376fd1aacf597b2f2fbbe0ea751419aee718af5d4f82537ef8c", size = 66616, upload-time = "2026-05-08T21:01:07.228Z" },
    { url = "https://files.pythonhosted.org/packages/83/e4/5c7462e50625f051f37fb38b8224f7639f667184bbd34424ec83819bb1b7/propcache-0.5.2-cp313-cp313t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:f7467da8a9822bf1a55336f877340c5bcbd3c482afc43a99771169f74a26dedc", size = 63773, upload-time = "2026-05-08T21:01:08.514Z" },
    { url = "https://files.pythonhosted.org/packages/ca/b6/99238894047b13c823be25027e736626cd414a52a5e30d2c3347c2733529/propcache-0.5.2-cp313-cp313t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:a6ddc6ac9e25de626c1f129c1b467d7ecd33ce2237d3fd0c4e429feef0a7ee1f", size = 63664, upload-time = "2026-05-08T21:01:09.874Z" },
    { url = "https://files.pythonhosted.org/packages/85/1e/a3a1a63116a2b8edb415a8bb9a6f0c34bd03830b1e18e8ce2904e1dc1cf4/propcache-0.5.2-cp313-cp313t-musllinux_1_2_aarch64.whl", hash = "sha256:2f22cbbac9e26a8e864c0985ff1268d5d939d53d9d9411a9824279097e03a2cb", size = 62643, upload-time = "2026-05-08T21:01:11.132Z" },
    { url = "https://files.pythonhosted.org/packages/e4/03/893cf147de2fc6543c5eaa07ad833170e7e2a2385725bbebe8c0503723bb/propcache-0.5.2-cp313-cp313t-musllinux_1_2_armv7l.whl", hash = "sha256:fc76378c62a0f04d0cd82fbb1a2cd2d7e28fcb40d5873f28a6c44e388aaa2751", size = 59595, upload-time = "2026-05-08T21:01:12.387Z" },
    { url = "https://files.pythonhosted.org/packages/86/3b/04c1a2e12c57766568ba75ba72b3bf2042818d4c1425fab6fc07155c7cff/propcache-0.5.2-cp313-cp313t-musllinux_1_2_ppc64le.whl", hash = "sha256:acd2c8edba48e31e58a363b8cf4e5c7db3b04b3f9e371f601df30d9b0d244836", size = 65711, upload-time = "2026-05-08T21:01:13.676Z" },
    { url = "https://files.pythonhosted.org/packages/1c/34/80f8d0099f8d6bacc4de1624c85672681c8cd1149ca2da0e38fd120b817f/propcache-0.5.2-cp313-cp313t-musllinux_1_2_riscv64.whl", hash = "sha256:452b5065457eb9991ec5eb38ff41d6cd4c991c9ac7c531c4d5849ae473a9a13f", size = 64247, upload-time = "2026-05-08T21:01:14.936Z" },
    { url = "https://files.pythonhosted.org/packages/f3/1a/8b08f3a5f1037e9e370c55883ceeeee0f6dd0416fb2d2d67b8bfc91f2a79/propcache-0.5.2-cp313-cp313t-musllinux_1_2_s390x.whl", hash = "sha256:3430bb2bfe1331885c427745a751e774ee679fd4344f80b97bf879815fe8fa55", size = 67102, upload-time = "2026-05-08T21:01:16.281Z" },
    { url = "https://files.pythonhosted.org/packages/34/68/8bdb7bb7756d76e005490649d10e4a8369e610c74d619f71e1aedf889e9c/propcache-0.5.2-cp313-cp313t-musllinux_1_2_x86_64.whl", hash = "sha256:cef6cea3922890dd6c9654971001fa797b526c16ab5e1e46c05fd6f877be7568", size = 64964, upload-time = "2026-05-08T21:01:17.57Z" },
    { url = "https://files.pythonhosted.org/packages/0a/aa/50fb0b5d3968b61a510926ff8b8465f1d6e976b3ab74496d7a4b9fc42515/propcache-0.5.2-cp313-cp313t-win32.whl", hash = "sha256:72d61e16dd78228b58c5d47be830ff3da7e5f139abdf0aef9d86cde1c5cf2191", size = 42546, upload-time = "2026-05-08T21:01:18.946Z" },
    { url = "https://files.pythonhosted.org/packages/ae/4c/0ddbae64321bd4a95bcbfc19307238016b5b1fee645c84626c8d539e5b74/propcache-0.5.2-cp313-cp313t-win_amd64.whl", hash = "sha256:0958834041a0166d343b8d2cedcd8bcbaeb4fdbe0cf08320c5379f143c3be6e7", size = 46330, upload-time = "2026-05-08T21:01:20.162Z" },
    { url = "https://files.pythonhosted.org/packages/00/d9/9cddc8efb78d8af264c5ec9f6d10b62f57c515feda8d321595f56010fb23/propcache-0.5.2-cp313-cp313t-win_arm64.whl", hash = "sha256:6de8bd93ddde9b992cf2b2e0d796d501a19026b5b9fd87356d7d0779531a8d96", size = 40521, upload-time = "2026-05-08T21:01:21.399Z" },
    { url = "https://files.pythonhosted.org/packages/e2/ea/23ee535d90ce8bcc465a3028eb3cc0ce3bd1005f4bb27710b30587de798d/propcache-0.5.2-cp314-cp314-macosx_10_15_universal2.whl", hash = "sha256:46088abff4cba581dea21ae0467a480526cb25aa5f3c269e909f800328bc3999", size = 94662, upload-time = "2026-05-08T21:01:22.683Z" },
    { url = "https://files.pythonhosted.org/packages/b5/06/c5a52f419b5d8972f8d46a7577476090d8e3263ff589ce40b5ca4968d5be/propcache-0.5.2-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:fc88b26f08d634f7bc819a7852e5214f5802641ab8d9fd5326892292eee1993e", size = 53928, upload-time = "2026-05-08T21:01:23.986Z" },
    { url = "https://files.pythonhosted.org/packages/63/b1/4260d67d6bd85e58a66b72d54ce15d5de789b6f3870cc6bedf8ff9667401/propcache-0.5.2-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:97797ebb098e670a2f92dd66f32897e30d7615b14e7f59711de23e30a9072539", size = 54650, upload-time = "2026-05-08T21:01:25.305Z" },
    { url = "https://files.pythonhosted.org/packages/70/06/2f46c318e3307cd7a6a7481def374ce838c0fe20084b39dd54b0879d0e99/propcache-0.5.2-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:ba57fffe4ac99c5d30076161b5866336d97600769bad35cc68f7774b15298a4e", size = 59912, upload-time = "2026-05-08T21:01:26.545Z" },
    { url = "https://files.pythonhosted.org/packages/4c/29/fe1aebec2ce57ab985a9c382bded1124431f85078113aa222c5d278430d4/propcache-0.5.2-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:583c19759d9eec1e5b69e2fbef36a7d9c326041be9746cb822d335c8cedc2979", size = 63300, upload-time = "2026-05-08T21:01:27.937Z" },
    { url = "https://files.pythonhosted.org/packages/b4/18/2334b26768b6c82be8c69e83671b767d5ef426aa09b0cba6c2ea47816774/propcache-0.5.2-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:d0326e2e5e1f3163fa306c834e48e8d490e5fae607a097a40c0648109b47ba80", size = 64208, upload-time = "2026-05-08T21:01:29.484Z" },
    { url = "https://files.pythonhosted.org/packages/2b/76/7f1bfd6afff4c5e38e36a3c6d68eb5f4b7311ea80baf693db78d95b603c4/propcache-0.5.2-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:e00820e192c8dbebcafb383ebbf99030895f09905e7a0eb2e0340a0bcc2bc825", size = 61633, upload-time = "2026-05-08T21:01:31.068Z" },
    { url = "https://files.pythonhosted.org/packages/c4/46/b3ff8aba2b4953a3e50de2cf72f1b5748b8eca93b15f3dc2c84339084c09/propcache-0.5.2-cp314-cp314-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:c66afea89b1e43725731d2004732a046fe6fe955d51f952c3e95a7314a284a39", size = 61724, upload-time = "2026-05-08T21:01:32.374Z" },
    { url = "https://files.pythonhosted.org/packages/c5/01/814cfcafbcff954f94c01cf30e097ddc88a076b5440fbcf4570753437d40/propcache-0.5.2-cp314-cp314-mu
```

```
sllinux_1_2_aarch64.whl", hash = "sha256:d4dc37dec6c6cdad0b57881a5658fd14fbf53e333b1a86cf86559f190e1d9ec4", size = 60069, upload-time = "2026-05-08T21
:01:33.67Z" },
    { url = "https://files.pythonhosted.org/packages/da/68/5c6f7622d510cc666a300687e06fd060c1a43361c0c9b20d284f06d8096a/propcache-0.5.2-cp314-cp314-mu
sllinux_1_2_armv7l.whl", hash = "sha256:5570dbcc97571c15f68068e529c92715a12f8d54030e272d264b377e22bd17a5", size = 57099, upload-time = "2026-05-08T21:
01:34.915Z" },
    { url = "https://files.pythonhosted.org/packages/55/27/9cb0b4c679124085327957d42521c99dba04c88c90c3e55a6f0b633ebccc/propcache-0.5.2-cp314-cp314-mu
sllinux_1_2_ppc64le.whl", hash = "sha256:f814362777a9f841adddb200ecdf8f5cb1e5a3c4b7a86378edbd6ccb26edd702", size = 63391, upload-time = "2026-05-08T21
:01:36.231Z" },
    { url = "https://files.pythonhosted.org/packages/f0/9d/7258aaa5bdf60fc6f27591eef6fe52768cb0beda7140be477c8b12c9794a/propcache-0.5.2-cp314-cp314-mu
sllinux_1_2_riscv64.whl", hash = "sha256:196913dea116aeb5a2ba95af4ddcb7ea85559ae07d8eee8751688310d09168c3", size = 61626, upload-time = "2026-05-08T21
:01:37.545Z" },
    { url = "https://files.pythonhosted.org/packages/8e/0d/41c602003e8a9b16fe1e7eadf62c7bfba9d5474370b24200bf48b315f45f/propcache-0.5.2-cp314-cp314-mu
sllinux_1_2_s390x.whl", hash = "sha256:6e7b8719005dd1175be4ab1cd25e9b98659a5e0347331506ec6760d2773a7fb5", size = 64781, upload-time = "2026-05-08T21:0
1:38.83Z" },
    { url = "https://files.pythonhosted.org/packages/8b/f3/38e66b1856e9bd079deea015bc4a55f7767c0e4db2f7dcf69e7e680ba4ce/propcache-0.5.2-cp314-cp314-mu
sllinux_1_2_x86_64.whl", hash = "sha256:51f96d685ab16e88cab128cd37a52c5da540809c8b879fa047731bfcb4ad35a4", size = 62570, upload-time = "2026-05-08T21:
01:40.415Z" },
    { url = "https://files.pythonhosted.org/packages/95/ca/bbfe9b910ce57dde8bb4876b4520fc02a4e89497c10de26be936758a3aaa/propcache-0.5.2-cp314-cp314-wi
n32.whl", hash = "sha256:cc6fc3cc62e8501d3ed62894425040d2728ecddb1ed072737a5c70bd537aa9f0", size = 39436, upload-time = "2026-05-08T21:01:41.654Z" },
    { url = "https://files.pythonhosted.org/packages/61/d2/45c9defbaa1ea297035d9d4cce9e8f80daafbf19319c6007f157c6256ea9/propcache-0.5.2-cp314-cp314-wi
n_amd64.whl", hash = "sha256:81e3a30b0bb60caa22033dd0f8a3618d1d67356212514f62c57db75cb0ef410c", size = 42373, upload-time = "2026-05-08T21:01:43.041Z"
 },
    { url = "https://files.pythonhosted.org/packages/44/68/9ea5103f41d5217d7d6ec24db90018e23aebec070c3f9a6e54d12b841fd8/propcache-0.5.2-cp314-cp314-wi
n_arm64.whl", hash = "sha256:0d2c9bf8528f135dbb805ce027567e09164f7efa51a2be07458a2c0420f292d0", size = 38554, upload-time = "2026-05-08T21:01:44.336Z"
 },
    { url = "https://files.pythonhosted.org/packages/8a/81/fadf555f42d3b762eea8a53950b0489fdc0aa9da5f8ed9e10ce0a4e01b48/propcache-0.5.2-cp314-cp314t-m
acosx_10_15_universal2.whl", hash = "sha256:4bc8ff1feffc6a61c7002ffe84634c41b822e104990ae009f44a0834430070bb", size = 99395, upload-time = "2026-05-08
T21:01:45.883Z" },
    { url = "https://files.pythonhosted.org/packages/f5/c9/c61e134a686949cf7971af3a390148b1156f7be81c73bc0cd12c873e2d48/propcache-0.5.2-cp314-cp314t-m
acosx_10_15_x86_64.whl", hash = "sha256:79aa3ff0a9b566633b642fa9caf7e21ed1c13d6feca718187873f199e1514078", size = 56653, upload-time = "2026-05-08T21:
01:47.307Z" },
    { url = "https://files.pythonhosted.org/packages/cb/73/daf935ea7048ddd7ec8eec5345b4a40b619d2d178b3c0a0900796bc3c794/propcache-0.5.2-cp314-cp314t-m
acosx_11_0_arm64.whl", hash = "sha256:1b31822f4474c4036bae62de9402710051d431a606d6a0f907fec79935a071aa", size = 56914, upload-time = "2026-05-08T21:01
:48.573Z" },
    { url = "https://files.pythonhosted.org/packages/79/9f/aba959b435ea18617edd7cf0a7ad0b9c574b8fc7e3d2cd55fb59cb255d33/propcache-0.5.2-cp314-cp314t-m
anylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:13fef48778b5a2a756523fdb781326b028ca75e32858b04f2cdd19f3945649
17", size = 62567, upload-time = "2026-05-08T21:01:49.903Z" },
    { url = "https://files.pythonhosted.org/packages/6c/a1/859942de9a791ff42f6141736f5b37749b8f53e65edfa49638c67dd67e6a/propcache-0.5.2-cp314-cp314t-m
anylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:8b73ab70f1a3351fbc71f663b3e645af6dd0329100c353081cf69c37433fc6
fe", size = 65542, upload-time = "2026-05-08T21:01:51.204Z" },
    { url = "https://files.pythonhosted.org/packages/b5/61/315bc0fd6c0fc7f80a528b8afd209e5fc4a875ea79571b91b8f50f442907/propcache-0.5.2-cp314-cp314t-m
anylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:5538d2c13d93e4698af7e092b57bc7298fd35d1d58e656ae18f23ee0d0378e03", s
ize = 66845, upload-time = "2026-05-08T21:01:52.539Z" },
    { url = "https://files.pythonhosted.org/packages/47/f7/9f8122e3132e8e354ac41975ef8f1099be7d5a16bc7ae562734e993665c0/propcache-0.5.2-cp314-cp314t-m
anylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:cd645f03898405cabe694fb8bc35241e3a9c332ec85627584fe3de201452b335"
, size = 63985, upload-time = "2026-05-08T21:01:53.847Z" },
    { url = "https://files.pythonhosted.org/packages/c8/54/c317819ec157cbf6f35df9df9657a6f82daf34d5faf15948b2f639c2192e/propcache-0.5.2-cp314-cp314t-m
anylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:a473b3440261e0c60706e732b2ed2f517857344fc21bf48fdfe211e2d98eb285", size = 63999, upl
oad-time = "2026-05-08T21:01:55.179Z" },
    { url = "https://files.pythonhosted.org/packages/5a/56/387e3f7dfce0a9233df41fb888aa1c30222cb4bbbf09537c02dd9bd85fe2/propcache-0.5.2-cp314-cp314t-m
usllinux_1_2_aarch64.whl", hash = "sha256:7afa37062e6650640e932e4cc9297d81f9f42d9944029cc386b8247dea4da837", size = 62779, upload-time = "2026-05-08T2
1:01:57.489Z" },
    { url = "https://files.pythonhosted.org/packages/a1/9c/596784cb5824ed61ee960d3f8655a3f0993e107c6e98ab6c818b7fb92ccb/propcache-0.5.2-cp314-cp314t-m
usllinux_1_2_armv7l.whl", hash = "sha256:8a90efd5777e996e42d568db9ac740b944d691e565cbfd31b2f7832f9184b2b8", size = 59796, upload-time = "2026-05-08T21
:01:58.736Z" },
    { url = "https://files.pythonhosted.org/packages/c2/3d/1a6cfa1726a48542c1e8784a0761421476a5b68e09b7f36bf95eb954aaba/propcache-0.5.2-cp314-cp314t-m
usllinux_1_2_ppc64le.whl", hash = "sha256:f19bb891234d72535764d703bfed1153cc34f4214d5bd7150aee1eec9e8f4366", size = 66023, upload-time = "2026-05-08T2
1:02:00.228Z" },
    { url = "https://files.pythonhosted.org/packages/e4/0e/05fd6990369477076e4e280bcb970de760fddf0161a46e988bc95f7940ec/propcache-0.5.2-cp314-cp314t-m
usllinux_1_2_riscv64.whl", hash = "sha256:32775082acd2d807ee3db715c7770d38767b817870acfa08c29e057f3c4d5b56", size = 64448, upload-time = "2026-05-08T2
1:02:01.888Z" },
    { url = "https://files.pythonhosted.org/packages/cd/86/5f8da315a4309c62c10c0b2516b17492d5d3bbe1bb862b96604db67e2a37/propcache-0.5.2-cp314-cp314t-m
usllinux_1_2_s390x.whl", hash = "sha256:9282fb1a3bccd038da9f768b927b24a0c753e466c086b7c4f3c6982851eefb2d", size = 67329, upload-time = "2026-05-08T21:
02:03.484Z" },
    { url = "https://files.pythonhosted.org/packages/da/d3/3368efe79ab21f0cdf86ef49895811c9cc933131d4cde1f28a624e22e712/propcache-0.5.2-cp314-cp314t-m
usllinux_1_2_x86_64.whl", hash = "sha256:cc49723e2f60d6b32a0f0b08a3fd6d13203c07f1cd9566cfce0f12a917c967a2", size = 65172, upload-time = "2026-05-08T21
:02:04.745Z" },
    { url = "https://files.pythonhosted.org/packages/d5/07/127e8b0bacfb325396196f9d976a22453049b89b9b2b08477cc3145faa44/propcache-0.5.2-cp314-cp314t-w
in32.whl", hash = "sha256:2d7aa89ebca5acc98cba9d1472d976e394782f587bad6661003602a619fd1821", size = 43813, upload-time = "2026-05-08T21:02:06.025Z" },
    { url = "https://files.pythonhosted.org/packages/88/fb/46dad6c0ae49ed230ab1b16c890c2b6314e2403e6c412976f4a72d64a527/propcache-0.5.2-cp314-cp314t-w
in_amd64.whl", hash = "sha256:d447bb0b3054be5818458fbb171208b1d9ff11eba14e18ca18b90cbb45767370", size = 47764, upload-time = "2026-05-08T21:02:07.353Z
" },
    { url = "https://files.pythonhosted.org/packages/e7/c4/a47d0a63aa309d10d59ede6e9d4cff03a344a79d1f0f4cd0cd74997b53e0/propcache-0.5.2-cp314-cp314t-w
in_arm64.whl", hash = "sha256:fe67a3d11cd9b4efabfa45c3d00ffba2b26811442a73a581a94b67c2b5faccf6", size = 41140, upload-time = "2026-05-08T21:02:09.065Z
" },
    { url = "https://files.pythonhosted.org/packages/3a/ed/1cdcab6ba3d6ab7feca11fc14f0eeea80755bb53ef4e892079f31b10a25f/propcache-0.5.2-py3-none-any.w
hl", hash = "sha256:be1ddfcbb376e3de5d2e2db1d58d6d67463e6b4f9f040c000de8e300295465fe", size = 14036, upload-time = "2026-05-08T21:02:10.673Z" },
]

[[package]]
name = "protobuf"
version = "7.35.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/da/01/9ef0afd7999eb9badb3a768b4aedd78c86d4c65cfaf1958ab276199e76b4/protobuf-7.35.1.tar.gz", h
ash = "sha256:ce115a26fe0c39a2c29973d914d327e516a6455464489fe3cd1e51a1b354f81a", size = 458717, upload-time = "2026-06-11T21:55:40.257Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/10/03/8aeeb7458d22546bf64b5250ca1daeb5ff757d900e8e4a7476c6f0db843e/protobuf-7.35.1-cp310-abi3-mac
osx_10_9_universal2.whl", hash = "sha256:24f857477359a85c0c235261b8ba905fd51b2562f4a64ca1df5473f29850cbf6", size = 433226, upload-time = "2026-06-11T2
1:55:31.719Z" },
    { url = "https://files.pythonhosted.org/packages/37/4b/dfb89eb0e652a1ff073c39a59fb5e3a83cfe9b57a2c83fa6d78270101767/protobuf-7.35.1-cp310-abi3-man
ylinux2014_aarch64.whl", hash = "sha256:11d6b0ec246892d85215b0a13ca6e0233cf5284b68f0ac02646427f4ff88a799", size = 328847, upload-time = "2026-06-11T21
:55:34.035Z" },
    { url = "https://files.pythonhosted.org/packages/0f/58/dc12f2cd484951524af6e3382c785869b9b3fb5e52ee95ae23add53ee8f9/protobuf-7.35.1-cp310-abi3-man
ylinux2014_s390x.whl", hash = "sha256:b73f9489a4b8b1c9cb1f8ed951c736392592edb24b9d6819f36d2e10b171d5b4", size = 344030, upload-time = "2026-06-11T21:5
5:34.941Z" },
    { url = "https://files.pythonhosted.org/packages/e4/be/5b3cfe508bfab6761414ff944e3366eb13be4fd71efcd69450f89ba39f43/protobuf-7.35.1-cp310-abi3-man
ylinux2014_x86_64.whl", hash = "sha256:74758715c53d7158fb76caf4f0cfdacc5329a4b1bb994f865d6cf302d413a1c4", size = 327130, upload-time = "2026-06-11T21:
55:35.921Z" },
    { url = "https://files.pythonhosted.org/packages/d8/bc/6d6c7ba8709c85f8f2c390b2b118d6fb08a783676a572271851bf45a7d22/protobuf-7.35.1-cp310-abi3-win
32.whl", hash = "sha256:353652e4efd0bca5b5fc2656abf8307ef351f0cf938c9eba09f0e09c20a25c30", size = 428945, upload-time = "2026-06-11T21:55:37.034Z" },
    { url = "https://files.pythonhosted.org/packages/0a/19/8d0cb6f20a1ef7b18f1c8986ad5783f22f84cce39c6ce9a6e645ea55192e/protobuf-7.35.1-cp310-abi3-win
_amd64.whl", hash = "sha256:230a75ddfc2de4806e56696ce9640c1cdfdb6543b7cfce98d42a4c0a0e7bdb87", size = 439996, upload-time = "2026-06-11T21:55:38.123Z"
 },
    { url = "https://files.pythonhosted.org/packages/19/c7/5f7c636ec43e0c545e28d1f1db71990108306f7bdcb89f069ba97e428e7f/protobuf-7.35.1-py3-none-any.w
hl", hash = "sha256:4bc97768d8fe4ad6743c8a19403e314511ed9f6d13205b687e52421c023ac1b9", size = 171659, upload-time = "2026-06-11T21:55:39.155Z" },
```

```
]

[[package]]
name = "pyasn1"
version = "0.6.4"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a4/9a/23310166d960def5897e91fe20e5b724601b02a22e84ba1f94232c0b7f67/pyasn1-0.6.4.tar.gz", hash
 = "sha256:9c447d8431c947fe4c8febc4ed9e760bc29011a5b01e5c74b67025bd9fb8ce81", size = 151262, upload-time = "2026-07-09T01:12:33.988Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/9a/3b/6163796d69c3977d1e4287bea4a6979161cbbdd170ebb430511e8e1999ce/pyasn1-0.6.4-py3-none-any.whl"
, hash = "sha256:deda9277cfd454080ec40b207fb6df82206a3a2688735233cdcd8d3d565f088b", size = 84410, upload-time = "2026-07-09T01:12:32.92Z" },
]

[[package]]
name = "pycparser"
version = "3.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/1b/7d/92392ff7815c21062bea51aa7b87d45576f649f16458d78b7cf94b9ab2e6/pycparser-3.0.tar.gz", has
h = "sha256:600f49d217304a5902ac3c37e1281c9fe94e4d0489de643a9504c5cdfdfc6b29", size = 103492, upload-time = "2026-01-21T14:26:51.89Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/0c/c3/44f3fbbfa403ea2a7c779186dc20772604442dde72947e7d01069cbe98e3/pycparser-3.0-py3-none-any.whl
", hash = "sha256:b727414169a36b7d524c1c3e31839a521725078d7b2ff038656844266160a992", size = 48172, upload-time = "2026-01-21T14:26:50.693Z" },
]

[[package]]
name = "pydantic"
version = "2.13.4"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "annotated-types" },
    { name = "pydantic-core" },
    { name = "typing-extensions" },
    { name = "typing-inspection" },
]
sdist = { url = "https://files.pythonhosted.org/packages/18/a5/b60d21ac674192f8ab0ba4e9fd860690f9b4a6e51ca5df118733b487d8d6/pydantic-2.13.4.tar.gz", h
ash = "sha256:c40756b57adaa8b1efeeced5c196f3f3b7c435f90e84ea7f443901bec8099ef6", size = 844775, upload-time = "2026-05-06T13:43:05.343Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/fd/7b/122376b1fd3c62c1ed9dc80c931ace4844b3c55407b6fb2d199377c9736f/pydantic-2.13.4-py3-none-any.w
hl", hash = "sha256:45a282cde31d808236fd7ea9d919b128653c8b38b393d1c4ab335c62924d9aba", size = 472262, upload-time = "2026-05-06T13:43:02.641Z" },
]

[[package]]
name = "pydantic-core"
version = "2.46.4"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/9d/56/921726b776ace8d8f5db44c4ef961006580d91dc52b803c489fafd1aa249/pydantic_core-2.46.4.tar.g
z", hash = "sha256:62f875393d7f270851f20523dd2e29f082bcc82292d66db2b64ea71f64b6e1c1", size = 471464, upload-time = "2026-05-06T13:37:06.98Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/ce/8c/af022f0af448d7747c5154288d46b5f2bc5f17366eaa0e23e9aa04d59f3b/pydantic_core-2.46.4-cp312-cp3
12-macosx_10_12_x86_64.whl", hash = "sha256:3245406455a5d98187ec35530fd772b1d799b26667980872c8d4614991e2c4a2", size = 2106158, upload-time = "2026-05-
06T13:38:57.215Z" },
    { url = "https://files.pythonhosted.org/packages/19/95/6195171e385007300f0f5574592e467c568becce2d937a0b6804f218bc49/pydantic_core-2.46.4-cp312-cp3
12-macosx_11_0_arm64.whl", hash = "sha256:962ccbab7b642487b1d8b7df90ef677e03134cf1fd8880bf698649b22a69371f", size = 1951724, upload-time = "2026-05-06
T13:37:02.697Z" },
    { url = "https://files.pythonhosted.org/packages/8e/bc/f47d1ff9cbb1620e1b5b697eef06010035735f07820180e74178226b27b3/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:8233f2947cf85404441fd7e0085f53b10c93e0ee78611099b5c7237e36aacbf7", size = 1975742
, upload-time = "2026-05-06T13:37:09.448Z" },
    { url = "https://files.pythonhosted.org/packages/5b/11/9b9a5b0306345664a2da6410877af6e8082481b5884b3ddd78d47c6013ce/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:3a233125ac121aa3ffba9a2b59edfc4a985a76092dc8279586ab4b71390875e7", size = 2052418,
upload-time = "2026-05-06T13:37:38.234Z" },
    { url = "https://files.pythonhosted.org/packages/f1/b7/a65fec226f5d78fc39f4a13c4cc0c768c22b113438f60c14adc9d2865038/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:5b712b53160b79a5850310b912a5ef8e57e56947c8ad690c227f5c9d7e561712", size = 2232274
, upload-time = "2026-05-06T13:38:27.753Z" },
    { url = "https://files.pythonhosted.org/packages/68/f0/92039db98b907ef49269a8271f67db9cb78ae2fc68062ef7e4e77adb5f61/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:9401557acd873c3a7f3eb9383edef8ac4968f9510e340f4808d427e75667e7b4", size = 2309940, up
load-time = "2026-05-06T13:38:05.353Z" },
    { url = "https://files.pythonhosted.org/packages/5f/97/2aab507d3d00ca626e8e57c1eac6a79e4e5fbcc63eb99733ff55d1717f65/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:926c9541b14b12b1681dca8a0b75feb510b06c6341b70a8e500c2fdcff837cce", size = 2094516,
upload-time = "2026-05-06T13:39:10.577Z" },
    { url = "https://files.pythonhosted.org/packages/22/37/a8aca44d40d737dde2bc05b3c6c07dff0de07ce6f82e9f3167aeaf4d5dea/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_31_riscv64.whl", hash = "sha256:56cb4851bcaf3d117eddcef4fe66afd750a50274b0da8e22be256d10e5611987", size = 2136854, upload-time = "2026-
05-06T13:40:22.59Z" },
    { url = "https://files.pythonhosted.org/packages/24/99/fcef1b79238c06a8cbec70819ac722ba76e02bc8ada9b0fd66eba40da01b/pydantic_core-2.46.4-cp312-cp3
12-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:c68fcd102d71ea85c5b2dfac3f4f8476eff42a9e078fd5faefff6d145063536b", size = 2180306, upload-t
ime = "2026-05-06T13:40:10.666Z" },
    { url = "https://files.pythonhosted.org/packages/ae/6c/fc44000918855b42779d007ae63b0532794739027b2f417321cddbc44f6a/pydantic_core-2.46.4-cp312-cp3
12-musllinux_1_1_aarch64.whl", hash = "sha256:b2f69dec1725e79a012d920df1707de5caf7ed5e08f3be4435e25803efc47458", size = 2190044, upload-time = "2026-0
5-06T13:40:43.231Z" },
    { url = "https://files.pythonhosted.org/packages/6b/65/d9cadc9f1920d7a127ad2edba16c1db7916e59719285cd6c94600b0080ba/pydantic_core-2.46.4-cp312-cp3
12-musllinux_1_1_armv7l.whl", hash = "sha256:8d0820e8192167f80d88d64038e609c31452eeca865b4e1d9950a27a4609b00b", size = 2329133, upload-time = "2026-05
-06T13:39:57.365Z" },
    { url = "https://files.pythonhosted.org/packages/d0/cf/c873d91679f3a30bcf5e7ac280ce5573483e72295307685120d0d5ad3416/pydantic_core-2.46.4-cp312-cp3
12-musllinux_1_1_x86_64.whl", hash = "sha256:fbdb89b3e1c94a30cc5edfce477c6e6a5dc4d8f84665b455c27582f211a1c72c", size = 2374464, upload-time = "2026-05
-06T13:38:06.976Z" },
    { url = "https://files.pythonhosted.org/packages/47/bd/6f2fc8188f31bf10590f1e98e7b306336161fac930a8c514cd7bd828c7dc/pydantic_core-2.46.4-cp312-cp3
12-win32.whl", hash = "sha256:9aa768456404a8bf48a4406685ac2bec8e72b62c69313734fa3b73cf33b3a894", size = 1974823, upload-time = "2026-05-06T13:40:47.98
5Z" },
    { url = "https://files.pythonhosted.org/packages/40/8c/985c1d41ea1107c2534abd9870e4ed5c8e7669b5c308297835c001e7a1c4/pydantic_core-2.46.4-cp312-cp3
12-win_amd64.whl", hash = "sha256:e9c26f834c65f5752f3f06cb08cb86a913ceb7274d0db6e267808a708b46bc89", size = 2072919, upload-time = "2026-05-06T13:39:2
1.153Z" },
    { url = "https://files.pythonhosted.org/packages/c4/ba/f463d006e0c47373ca7ec5e1a261c59dc01ef4d62b2657af925fb0deee3a/pydantic_core-2.46.4-cp312-cp3
12-win_arm64.whl", hash = "sha256:4fc73cb559bdb54b1134a706a2802a4cddd27a0633f5abb7e53056268751ac6a", size = 2027604, upload-time = "2026-05-06T13:39:0
3.753Z" },
    { url = "https://files.pythonhosted.org/packages/51/a2/5d30b469c5267a17b39dec53208222f76a8d351dfac4af661888c5aee77d/pydantic_core-2.46.4-cp313-cp3
13-macosx_10_12_x86_64.whl", hash = "sha256:5d5902252db0d3cedf8d4a1bc68f70eeb430f7e4c7104c8c476753519b423008", size = 2106306, upload-time = "2026-05-
06T13:37:48.029Z" },
    { url = "https://files.pythonhosted.org/packages/c1/81/4fa520eaffa8bd7d1525e644cd6d39e7d60b1592bc5b516693c7340b50f1/pydantic_core-2.46.4-cp313-cp3
13-macosx_11_0_arm64.whl", hash = "sha256:c94f0688e7b8d0a67abf40e57a7eaaecd17cc9586706a31b76c031f63df052b4", size = 1951906, upload-time = "2026-05-06
T13:37:17.012Z" },
    { url = "https://files.pythonhosted.org/packages/03/d5/fd02da45b659668b05923b17ba3a0100a0a3d5541e3bd8fcc4ecb711309e/pydantic_core-2.46.4-cp313-cp3
13-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:f027324c56cd5406ca49c124b0db10e56c69064fec039acc571c29020cc87c76", size = 1976802
, upload-time = "2026-05-06T13:37:35.113Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/21/f2/95727e1368be3d3ed485eaab7adbd7dda408f33f7a36e8b48e0144002b91/pydantic_core-2.46.4-cp313-cp313-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:e739fee756ba1010f8bcccb534252e85a35fe45ae92c295a06059ce58b74ccd3", size = 2052446, upload-time = "2026-05-06T13:37:12.313Z" },
    { url = "https://files.pythonhosted.org/packages/9c/86/5d99feea3f77c7234b8718075b23db11532773c1a0dbd9b9490215dc2eeb/pydantic_core-2.46.4-cp313-cp313-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:9d56801be94b86a9da183e5f3766e6310752b99ff647e38b09a9500d88e46e76", size = 2232757, upload-time = "2026-05-06T13:39:01.149Z" },
    { url = "https://files.pythonhosted.org/packages/d2/3a/508ac615935ef7588cf6d9e9b91309fdc2da751af865e02a9098de88258c/pydantic_core-2.46.4-cp313-cp313-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:2412e734dcb48da14d4e4006b82b46b74f2518b8a26ee7e58c6844a6cd6d03c4", size = 2309275, upload-time = "2026-05-06T13:37:41.406Z" },
    { url = "https://files.pythonhosted.org/packages/07/f8/41db9de19d7987d6b04715a02b3b40aea467000275d9d758ffaa31af7d50/pydantic_core-2.46.4-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:9551187363ffc0de2a00b2e47c25aeaeb1020b69b668762966df15fc5659dd5a", size = 2094467, upload-time = "2026-05-06T13:39:18.847Z" },
    { url = "https://files.pythonhosted.org/packages/2c/e2/f35033184cb11d0052daf4416e8e10a502ea2ac006fc4f459aee872727d1/pydantic_core-2.46.4-cp313-cp313-manylinux_2_31_riscv64.whl", hash = "sha256:0186750b482eefa11d7f435892b09c5c606193ef3375bcf94aa00ae6bfb66262", size = 2134417, upload-time = "2026-05-06T13:40:17.944Z" },
    { url = "https://files.pythonhosted.org/packages/7e/7b/6ceeb1cc90e193862f444ebe373d8fdf613f0a82572dde03fb10734c6c71/pydantic_core-2.46.4-cp313-cp313-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:5855698a4856556d86e8e6cd8434bc3ac0314ee8e12089ae0e143f64c6256e4e", size = 2179782, upload-time = "2026-05-06T13:40:32.618Z" },
    { url = "https://files.pythonhosted.org/packages/5a/f2/c8d7773ede6af08036423a00ae0ceffce266c3c52a096c435d68c896083f/pydantic_core-2.46.4-cp313-cp313-musllinux_1_1_aarch64.whl", hash = "sha256:cbaf13819775b7f769bf4a1f066cb6df7a28d4480081a589828ef190226881cd", size = 2188782, upload-time = "2026-05-06T13:36:51.018Z" },
    { url = "https://files.pythonhosted.org/packages/59/31/0c864784e31f09f05cdd87606f08923b9c9e7f6e51dd27f20f62f975ce9f/pydantic_core-2.46.4-cp313-cp313-musllinux_1_1_armv7l.whl", hash = "sha256:633147d34cf4550417f12e2b1a0383973bdf5cdfde212cb09e9a581cf10820be", size = 2328334, upload-time = "2026-05-06T13:40:37.764Z" },
    { url = "https://files.pythonhosted.org/packages/c2/eb/4f6c8a41efa30baa755590f4141abf3a8c370fab610915733e74134a7270/pydantic_core-2.46.4-cp313-cp313-musllinux_1_1_x86_64.whl", hash = "sha256:82cf5301172168103724d49a1444d3378cb20cdee30b116a1bd6031236298a5d", size = 2372986, upload-time = "2026-05-06T13:39:34.152Z" },
    { url = "https://files.pythonhosted.org/packages/5b/24/b375a480d53113860c299764bfe9f349a3dc9108b3adc0d7f0d786492ebf/pydantic_core-2.46.4-cp313-cp313-win32.whl", hash = "sha256:9fa8ae11da9e2b3126c6426f147e0fba88d96d65921799bb30c6abd1cb2c97fb", size = 1973693, upload-time = "2026-05-06T13:37:55.072Z" },
    { url = "https://files.pythonhosted.org/packages/7e/e8/cff247591966f2d22ec8c003cd7587e27b7ba7b81ab2fb888e3ab75dc285/pydantic_core-2.46.4-cp313-cp313-win_amd64.whl", hash = "sha256:6b3ace8194b0e5204818c92802dcdca7fc6d88aabbb799d7c795540d9cd6d292", size = 2071819, upload-time = "2026-05-06T13:38:49.139Z" },
    { url = "https://files.pythonhosted.org/packages/c6/1a/f4aee670d5670e9e148e0c82c7db98d780be566c6e6a97ee8035528ca0b3/pydantic_core-2.46.4-cp313-cp313-win_arm64.whl", hash = "sha256:184c081504d17f1c1066e430e117142b2c77d9448a97f7b65c6ac9fd9aee238d", size = 2027411, upload-time = "2026-05-06T13:40:45.796Z" },
    { url = "https://files.pythonhosted.org/packages/8d/74/228a26ddad29c6672b805d9fd78e8d251cd04004fa7eed0e622096cd0250/pydantic_core-2.46.4-cp314-cp314-macosx_10_12_x86_64.whl", hash = "sha256:428e04521a40150c85216fc8b85e8d39fece235a9cf5e383761238c7fa9b96fb", size = 2102079, upload-time = "2026-05-06T13:38:41.019Z" },
    { url = "https://files.pythonhosted.org/packages/ad/1f/8970b150a4b4365623ae00fc88603491f763c627311ae8031e3111356d6e/pydantic_core-2.46.4-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:23ace664830ee0bfe014a0c7bc248b1f7f25ed7ad103852c317624a1083af462", size = 1952179, upload-time = "2026-05-06T13:36:59.812Z" },
    { url = "https://files.pythonhosted.org/packages/95/30/5211a831ae054928054b2f79731661087a2bc5c01e825c672b3a4a8f1b3e/pydantic_core-2.46.4-cp314-cp314-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:ce5c1d2a8b27468f433ca974829c44060b8097eedc39933e3c206a90ee49c4a9", size = 1978926, upload-time = "2026-05-06T13:37:39.933Z" },
    { url = "https://files.pythonhosted.org/packages/57/e9/689668733b1eb67adeef047db3c2e8788fcf65a7fd9c9e2b46b7744fe245/pydantic_core-2.46.4-cp314-cp314-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:7283d57845ecf5a163403eb0702dfc220cc4fbdd18919cb5ccea4f95ee1cdab4", size = 2046785, upload-time = "2026-05-06T13:38:01.995Z" },
    { url = "https://files.pythonhosted.org/packages/60/d9/6715260422ff50a2109878fd24d948a6c3446bb2664f34ee78cd972b3acd/pydantic_core-2.46.4-cp314-cp314-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:8daafc69c93ee8a0204506a3b6b30f586ef54028f52aeeeb5c4cfc5184fd5914", size = 2228733, upload-time = "2026-05-06T13:40:50.371Z" },
    { url = "https://files.pythonhosted.org/packages/18/ae/fdb2f64316afca925640f8e70bb1a564b0ec2721c1389e25b8eb4bf9a299/pydantic_core-2.46.4-cp314-cp314-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:cd2213145bcc2ba85884d0ac63d222fece9209678f77b9b4d76f054c561adb28", size = 2307534, upload-time = "2026-05-06T13:37:21.531Z" },
    { url = "https://files.pythonhosted.org/packages/89/1d/8eff589b45bb8190a9d12c49cfad0f176a5cbd1534908a6b5125e2886239/pydantic_core-2.46.4-cp314-cp314-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:7a5f930472650a82629163023e630d160863fce524c616f4e5186e5de9d9a49b", size = 2099732, upload-time = "2026-05-06T13:39:31.942Z" },
    { url = "https://files.pythonhosted.org/packages/06/d5/ee5a3366637fee41dee51a1fc91562dcf12ddbc68fda34e6b253da2324bb/pydantic_core-2.46.4-cp314-cp314-manylinux_2_31_riscv64.whl", hash = "sha256:c1b3f518abeca3aa13c712fd202306e145abf59a18b094a6bafb2d2bbf59192c", size = 2129627, upload-time = "2026-05-06T13:37:25.033Z" },
    { url = "https://files.pythonhosted.org/packages/94/33/2414be571d2c6a6c4d08be21f9292b6d3fdb08949a97b6dfe985017821db/pydantic_core-2.46.4-cp314-cp314-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:1a7dd0b3ee80d90150e3495a3a13ac34dbcbfd4f012996a6a1d8900e91b5c0fb", size = 2179141, upload-time = "2026-05-06T13:37:14.046Z" },
    { url = "https://files.pythonhosted.org/packages/7b/79/7daa95be995be0eecc4cf75064cb33f9bbbfe3fe0158caf2f0d4a996a5c7/pydantic_core-2.46.4-cp314-cp314-musllinux_1_1_aarch64.whl", hash = "sha256:3fb702cd90b0446a3a1c5e470bfa0dd23c0233b676a9099ddcc964fa6ca13898", size = 2184325, upload-time = "2026-05-06T13:36:53.615Z" },
    { url = "https://files.pythonhosted.org/packages/9f/cb/d0a382f5c0de8a222dc61c65348e0ce831b1f68e0a018450d31c2cace3a5/pydantic_core-2.46.4-cp314-cp314-musllinux_1_1_armv7l.whl", hash = "sha256:b8458003118a712e66286df6a707db01c52c0f52f7db8e4a38f0da1d3b94fc4e", size = 2323990, upload-time = "2026-05-06T13:40:29.971Z" },
    { url = "https://files.pythonhosted.org/packages/05/db/d9ba624cc4a5aced1598e88c04fdbd8310c8a69b9d38b9a3d39ce3a61ed7/pydantic_core-2.46.4-cp314-cp314-musllinux_1_1_x86_64.whl", hash = "sha256:372429a130e469c9cd698925ce5fc50940b7a1336b0d82038e63d5bbc4edc519", size = 2369978, upload-time = "2026-05-06T13:37:23.027Z" },
    { url = "https://files.pythonhosted.org/packages/f2/20/d15df15ba918c423461905802bfd2981c3af0bfa0e40d05e13edbfa48bc3/pydantic_core-2.46.4-cp314-cp314-win32.whl", hash = "sha256:85bb3611ff1802f3ee7fdd7dbff26b56f343fb432d57a4728fdd49b6ef35e2f4", size = 1966354, upload-time = "2026-05-06T13:38:03.499Z" },
    { url = "https://files.pythonhosted.org/packages/fc/b6/6b8de4c0a7d7ab3004c439c80c5c1e0a3e8d78bbae19379b01960383d9e5/pydantic_core-2.46.4-cp314-cp314-win_amd64.whl", hash = "sha256:811ff8e9c313ab425368bcbb36e5c4ebd7108c2bbf4e4089cfbb0b01eff63fac", size = 2072238, upload-time = "2026-05-06T13:39:40.807Z" },
    { url = "https://files.pythonhosted.org/packages/32/36/51eb763beec1f4cf59b1db243a7dcc39cbb41230f050a09b9d69faaf0a48/pydantic_core-2.46.4-cp314-cp314-win_arm64.whl", hash = "sha256:bfec22eab3c8cc2ceec0248aec886624116dc079afa027ecc8ad4a7e62010f8a", size = 2018251, upload-time = "2026-05-06T13:37:26.72Z" },
    { url = "https://files.pythonhosted.org/packages/e8/91/855af51d625b23aa987116a19e231d2aaef9c4a415273ddc189b79a45fee/pydantic_core-2.46.4-cp314-cp314t-macosx_10_12_x86_64.whl", hash = "sha256:af8244b2bef6aaad6d92cda81372de7f8c8d36c9f0c3ea36e827c60e7d9467a0", size = 2099593, upload-time = "2026-05-06T13:39:47.682Z" },
    { url = "https://files.pythonhosted.org/packages/fb/1b/8784a54c65edb5f49f0a14d6977cf1b209bba85a4c77445b255c2de58ab3/pydantic_core-2.46.4-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:5a4330cdbc57162e4b3aa303f588ba752257694c9c9be3e7ebb11b4aca659b5d", size = 1935226, upload-time = "2026-05-06T13:40:40.428Z" },
    { url = "https://files.pythonhosted.org/packages/e8/e7/1955d28d1afc56dd4b3ad7cc0cf39df1b9852964cf16e5d13912756d6d6b/pydantic_core-2.46.4-cp314-cp314t-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:29c61fc04a3d840155ff08e475a04809278972fe6aef51e2720554e96367e34b", size = 1974605, upload-time = "2026-05-06T13:37:32.029Z" },
    { url = "https://files.pythonhosted.org/packages/93/e2/3fedbf0ba7a22850e6e9fd78117f1c0f10f950182344d8a6c535d468fdd8/pydantic_core-2.46.4-cp314-cp314t-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:c50f2528cf200c5eed56faf3f4e22fcd5f38c157a8b78576e6ba3168ec35f000", size = 2030777, upload-time = "2026-05-06T13:38:55.239Z" },
    { url = "https://files.pythonhosted.org/packages/f8/61/46be275fcaaba0b4f5b9669dd852267ce1ff616592dccf7a7845588df091/pydantic_core-2.46.4-cp314-cp314t-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:0cbe8b01f948de4286c74cdd6c667aceb38f5c1e26f0693b3983d9d74887c65e", size = 2236641, upload-time = "2026-05-06T13:37:08.096Z" },
    { url = "https://files.pythonhosted.org/packages/60/db/12e93e46a8bac9988be3c016860f83293daea8c716c029c9ace279036f2f/pydantic_core-2.46.4-cp314-cp314t-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:617d7e2ca7dcb8c5cf6bcb8c59b8832c94b36196bbf1cbd1bfb56ed341905edd", size = 2286404, upload-time = "2026-05-06T13:40:20.221Z" },
    { url = "https://files.pythonhosted.org/packages/e2/4a/4d8b19008f38d31c53b8219cfedc2e3d5de5fe99d90076b7e767de29274f/pydantic_core-2.46.4-cp314-cp314t-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:7027560ee92211647d0d34e3f7cd6f50da56399d26a9c8ad0da286d3869a53f3", size = 2109219, upload-time = "2026-05-06T13:38:12.153Z" },
    { url = "https://files.pythonhosted.org/packages/88/70/3cbc40978fefb7bb09c6708d40d4ad1a5d70fd7213c3d17f971de868ec1f/pydantic_core-2.46.4-cp314-cp314t-manylinux_2_31_riscv64.whl", hash = "sha256:f99626688942fb746e545232e7726926f3be91b5975f8b55327665fafda991c7", size = 2110594, upload-time = "2026-05-06T13:40:02.971Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/9d/20/b8d36736216e29491125531685b2f9e61aa5b4b2599893f8268551da3338/pydantic_core-2.46.4-cp314-cp314t-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:fc3e9034a63de20e15e8ade85358bc6efc614008cab72898b4b4952bea0509ff", size = 2159542, upload-time = "2026-05-06T13:39:27.506Z" },
    { url = "https://files.pythonhosted.org/packages/1d/a2/367df868eb584dacf6bf82a389272406d7178e301c4ac82545ab98bc2dd9/pydantic_core-2.46.4-cp314-cp314t-musllinux_1_1_aarch64.whl", hash = "sha256:97e7cf2be5c77b7d1a9713a05605d49460d02c6078d38d8bef3cbe323c548424", size = 2168146, upload-time = "2026-05-06T13:38:31.93Z" },
    { url = "https://files.pythonhosted.org/packages/c1/b8/4460f77f7e201893f649a29ab355dddd3beee8a97bcb1a320db414f9a06e/pydantic_core-2.46.4-cp314-cp314t-musllinux_1_1_armv7l.whl", hash = "sha256:3bf92c5d0e00fefaab325a4d27828fe6b6e2a21848686b5b60d2d9eeb09d76c6", size = 2306309, upload-time = "2026-05-06T13:37:44.717Z" },
    { url = "https://files.pythonhosted.org/packages/64/c4/be2639293acd87dc8ddbcec41a73cee9b2ebf996fe6d892a1a74e88ad3f7/pydantic_core-2.46.4-cp314-cp314t-musllinux_1_1_x86_64.whl", hash = "sha256:3ecbc122d18468d06ca279dc26a8c2e2d5acb10943bb35e36ae92096dc3b5565", size = 2369736, upload-time = "2026-05-06T13:37:05.645Z" },
    { url = "https://files.pythonhosted.org/packages/30/a6/9f9f380dbb301f67023bf8f707aaa75daadf84f7152d95c410fd7e81d994/pydantic_core-2.46.4-cp314-cp314t-win32.whl", hash = "sha256:e846ae7835bf0703ae43f534ab79a867146dadd59dc9ca5c8b53d5c8f7c9ef02", size = 1955575, upload-time = "2026-05-06T13:38:51.116Z" },
    { url = "https://files.pythonhosted.org/packages/40/1f/f1eb9eb350e795d1af8586289746f5c5677d16043040d63710e22abc43c9/pydantic_core-2.46.4-cp314-cp314t-win_amd64.whl", hash = "sha256:2108ba5c1c1eca18030634489dc544844144ee36357f2f9f780b93e7ddbb44b5", size = 2051624, upload-time = "2026-05-06T13:38:21.672Z" },
    { url = "https://files.pythonhosted.org/packages/f6/d2/42dd53d0a85c27606f316d3aa5d2869c4e8470a5ed6dec30e4a1abe19192/pydantic_core-2.46.4-cp314-cp314t-win_arm64.whl", hash = "sha256:4fcbe087dbc2068af7eda3aa87634eba216dbda64d1ae73c8684b621d33f6596", size = 2017325, upload-time = "2026-05-06T13:40:52.723Z" },
    { url = "https://files.pythonhosted.org/packages/9d/1d/8987ad40f65ae1432753072f214fb5c74fe47ffbd0698bb9cbbb585664f8/pydantic_core-2.46.4-graalpy312-graalpy250_312_native-macosx_10_12_x86_64.whl", hash = "sha256:1d8ba486450b14f3b1d63bc521d410ec7565e52f887b9fb671791886436a42f7", size = 2095527, upload-time = "2026-05-06T13:39:52.283Z" },
    { url = "https://files.pythonhosted.org/packages/64/d3/84c282a7eee1d3ac4c0377546ef5a1ea436ce26840d9ac3b7ed54a377507/pydantic_core-2.46.4-graalpy312-graalpy250_312_native-macosx_11_0_arm64.whl", hash = "sha256:3009f12e4e90b7f88b4f9adb1b0c4a3d58fe7820f3238c190047209d148026df", size = 1936024, upload-time = "2026-05-06T13:40:15.671Z" },
    { url = "https://files.pythonhosted.org/packages/d7/ca/eac61596cdeb4d7e174d3dc0bd8a6238f14f75f97a24e7b7db4c7e7340a0/pydantic_core-2.46.4-graalpy312-graalpy250_312_native-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:ad785e92e6dc634c21555edc8bd6b64957ab844541bcb96a1366c202951ae526", size = 1990696, upload-time = "2026-05-06T13:38:34.717Z" },
    { url = "https://files.pythonhosted.org/packages/fa/c3/7c8b240552251faf6b3a957db200fcfbbcec36763c050428b601e0c9b83b/pydantic_core-2.46.4-graalpy312-graalpy250_312_native-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:00c603d540afdd6b80eb39f078f33ebd46211f02f33e34a32d9f053bba711de0", size = 2147590, upload-time = "2026-05-06T13:39:29.883Z" },
]

[[package]]
name = "pydantic-settings"
version = "2.14.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "pydantic" },
    { name = "python-dotenv" },
    { name = "typing-inspection" },
]
sdist = { url = "https://files.pythonhosted.org/packages/5c/b5/8f48e906c3e0205276e8bd8cb7512217a87b2685304d64be27cad5b3019f/pydantic_settings-2.14.2.tar.gz", hash = "sha256:c19dd64b19097f1de80184f0cc7b0272a13ae6e170cbf240a3e27e381ed14a5f", size = 237700, upload-time = "2026-06-19T13:44:56.324Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/77/c1/6e422f34e569cf8e18df68d1939c81c099d2b61e4f7d9621c8a77560799c/pydantic_settings-2.14.2-py3-none-any.whl", hash = "sha256:a20c97b37910b6550d5ea50fbcc2d4187defe58cd57070b73863d069419c9440", size = 61715, upload-time = "2026-06-19T13:44:55.02Z" },
]

[[package]]
name = "pygments"
version = "2.20.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/c3/b2/bc9c9196916376152d655522fdcebac55e66de6603a76a02bca1b6414f6c/pygments-2.20.0.tar.gz", hash = "sha256:6757cd03768053ff99f3039c1a36d6c0aa0b263438fcab17520b30a303a82b5f", size = 4955991, upload-time = "2026-03-29T13:29:33.898Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/f4/7e/a72dd26f3b0f4f2bf1dd8923c85f7ceb43172af56d63c7383eb62b332364/pygments-2.20.0-py3-none-any.whl", hash = "sha256:81a9e26dd42fd28a23a2d169d86d7ac03b46e2f8b59ed4698fb4785f946d0176", size = 1231151, upload-time = "2026-03-29T13:29:30.038Z" },
]

[[package]]
name = "pyjwt"
version = "2.13.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/3b/81/58d0ac84e1ef3a3843791d6954d94c0b33d526c75eeb1efbce9d0a4c4077/pyjwt-2.13.0.tar.gz", hash = "sha256:41571c89ca91598c79e8ef18a2d07367d4810fbbd6f637794879baf1b7703423", size = 107515, upload-time = "2026-05-21T19:54:36.618Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a3/5e/ecf12fdb62546d64385c158514e9b2b671f7832108ef2ecd2020ce0af2d1/pyjwt-2.13.0-py3-none-any.whl", hash = "sha256:66adcc2aff09b3f1bbd95fc1e1577df8ac8723c978552fd43304c8a290ac5728", size = 31274, upload-time = "2026-05-21T19:54:35.362Z" },
]

[package.optional-dependencies]
crypto = [
    { name = "cryptography" },
]

[[package]]
name = "pytest"
version = "9.1.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "colorama", marker = "sys_platform == 'win32'" },
    { name = "iniconfig" },
    { name = "packaging" },
    { name = "pluggy" },
    { name = "pygments" },
]
sdist = { url = "https://files.pythonhosted.org/packages/e4/47/b9efed96c114afcfa3c9d3fe98a76a1d14c74a9e266d397cf6eb64be5e01/pytest-9.1.1.tar.gz", hash = "sha256:1088fbde8f2b49d95a549a195707afa7a76a3ce9bcadc26b6d71f0ffda5fe313", size = 1636369, upload-time = "2026-06-19T10:58:32.857Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/24/25/1de2678b631f5a49215c6c96fff41ba892b0a34df68d6d80292b1b48aa7f/pytest-9.1.1-py3-none-any.whl", hash = "sha256:37a86b45efb9a47a61a36449063e8e18d0cab3161329fc099eb21783169c4f0c", size = 386536, upload-time = "2026-06-19T10:58:31.347Z" },
]

[[package]]
name = "pytest-asyncio"
version = "1.4.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "pytest" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
]
```

```
sdist = { url = "https://files.pythonhosted.org/packages/43/7c/d36d04db312ecf4298932ef77e6e4a9e8ad017906e24e34f0b0c361a2473/pytest_asyncio-1.4.0.tar.gz", hash = "sha256:c6c0d2259945122819f171a32ecea2c349ead889ee28176caaf492143424be42", size = 58514, upload-time = "2026-05-26T09:56:04.083Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/03/e2/08a497ef684b88559c9cc5f4ad53a37e7b99e727094a86d6ea32536d5d3c/pytest_asyncio-1.4.0-py3-none-any.whl", hash = "sha256:933ca923a23075a87fb7070c0ec272a6848489824d887c85c81267093283 5aa1", size = 16930, upload-time = "2026-05-26T09:56:02.576Z" },
]

[[package]]
name = "pytest-cov"
version = "7.1.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "coverage" },
    { name = "pluggy" },
    { name = "pytest" },
]
sdist = { url = "https://files.pythonhosted.org/packages/b1/51/a849f96e117386044471c8ec2bd6cfebacda285da9525c9106aeb28da671/pytest_cov-7.1.0.tar.gz", hash = "sha256:30674f2b5f6351aa09702a9c8c364f6a01c27aae0c1366ae8016160d1efc56b2", size = 55592, upload-time = "2026-03-21T20:11:16.284Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/9d/7a/d968e294073affff457b041c2be9868a40c1c71f4a35fcc1e45e5493067b/pytest_cov-7.1.0-py3-none-any.whl", hash = "sha256:a0461110b7865f9a271aa1b51e516c9a95de9d696734a2f71e3e78f46e1d4678", size = 22876, upload-time = "2026-03-21T20:11:14.438Z" },
]

[[package]]
name = "python-dateutil"
version = "2.9.0.post0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "six" },
]
sdist = { url = "https://files.pythonhosted.org/packages/66/c0/0c8b6ad9f17a802ee498c46e004a0eb49bc148f2fd230864601a86dcf6db/python-dateutil-2.9.0.post0.tar.gz", hash = "sha256:37dd54208da7e1cd875388217d5e00ebd4179249f90fb72437e91a35459a0ad3", size = 342432, upload-time = "2024-03-01T18:36:20.211Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/ec/57/56b9bcc3c9c6a792fcbaf139543cee77261f3651ca9da0c93f5c1221264b/python_dateutil-2.9.0.post0-py2.py3-none-any.whl", hash = "sha256:a8b2bc7bffae282281c8140a97d3aa9c14da0b136dfe83f850eea9a5f7470427", size = 229892, upload-time = "2024-03-01T18:36:18.57Z" },
]

[[package]]
name = "python-discovery"
version = "1.5.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "filelock" },
]
sdist = { url = "https://files.pythonhosted.org/packages/04/b7/1581a8103855c43567776aa34135e5ec3c597346c23bfd10c7eb5e0b10a4/python_discovery-1.5.1.tar.gz", hash = "sha256:e2ea8b884cd1701f386eda8cf327b87743f1dc21b7f784470799537d95635384", size = 77200, upload-time = "2026-07-31T22:06:02.48Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/6a/07/a89b539750a159d5101c4eb9fc84e2961f65cefbd5e0b7440b284471c0b0/python_discovery-1.5.1-py3-none-any.whl", hash = "sha256:ac07f44cade589d954e9d6a1e1468539fdddd2cf676beb51da73e0f156b7c932", size = 35752, upload-time = "2026-07-31T22:06:01.116Z" },
]

[[package]]
name = "python-dotenv"
version = "1.2.2"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/82/ed/0301aeeac3e5353ef3d94b6ec08bbcabd04a72018415dcb29e588514bba8/python_dotenv-1.2.2.tar.gz", hash = "sha256:2c371a91fbd7ba082c2c1dc1f8bf89ca22564a087c2c287cd9b662adde799cf3", size = 50135, upload-time = "2026-03-01T16:00:26.196Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/0b/d7/1959b9648791274998a9c3526f6d0ec8fd2233e4d4acce81bbae76b44b2a/python_dotenv-1.2.2-py3-none-any.whl", hash = "sha256:1d8214789a24de455a8b8bd8ae6fe3c6b69a5e3d64aa8a8e5d68e694bbcb285a", size = 22101, upload-time = "2026-03-01T16:00:25.09Z" },
]

[[package]]
name = "python-jose"
version = "3.5.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "ecdsa" },
    { name = "pyasn1" },
    { name = "rsa" },
]
sdist = { url = "https://files.pythonhosted.org/packages/c6/77/3a1c9039db7124eb039772b935f2244fbb73fc8ee65b9acf2375da1c07bf/python_jose-3.5.0.tar.gz", hash = "sha256:fb4eaa44dbeb1c26dcc69e4bd7ec54a1cb8dd64d3b4d81ef08d90ff453f2b01b", size = 92726, upload-time = "2025-05-28T17:31:54.288Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d9/c3/0bd11992072e6a1c513b16500a5d07f91a24017c5909b02c72c62d7ad024/python_jose-3.5.0-py2.py3-none-any.whl", hash = "sha256:abd1202f23d34dfad2c3d28cb8617b90acf34132c7afd60abd0b0b7d3cb55771", size = 34624, upload-time = "2025-05-28T17:31:52.802Z" },
]

[package.optional-dependencies]
cryptography = [
    { name = "cryptography" },
]

[[package]]
name = "python-multipart"
version = "0.0.32"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/5b/42/55c32bb9b12693c092ad250a0e82edb5b31ddeda6eb772de5f308b3804ad/python_multipart-0.0.32.tar.gz", hash = "sha256:be54b7f3fa167bb83e4fcd936b887b708f4e57fe75911c02aebf53efaf8d938e", size = 46881, upload-time = "2026-06-04T16:18:58.647Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e1/04/e8135ebd1ad02c56ec633277529b2602ff99ff634be76cdba5744cf554fd/python_multipart-0.0.32-py3-none-any.whl", hash = "sha256:ff6d3f776f16878c894e52e107296ffc890e913c611b1a4ec6c44e2821fe2e23", size = 30042, upload-time = "2026-06-04T16:18:57.319Z" },
]

[[package]]
name = "pywin32"
version = "312"
source = { registry = "https://pypi.org/simple" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/83/ff/32aa7d2ed0ab12b323aaa64f9b75e6ad4f8fd09f9ccfc28c79414d46838d/pywin32-312-cp312-cp312-win32.whl", hash = "sha256:dab4f65ac9c4e48400a2a0530c46c3c579cd5905ecd11b80692373915269208b", size = 6371877, upload-time = "2026-06-04T07:49:28.836Z" },
    { url = "https://files.pythonhosted.org/packages/03/d9/77040d3b43df3f3be32ea289433d660d2727f5ba327bc73be835127d9d60/pywin32-312-cp312-cp312-win_am
```

```
d64.whl", hash = "sha256:b457f6d628a47e8a7346ce22acb7e1a46a4a78b52e1d17e1af56871bd19a93bc", size = 6914841, upload-time = "2026-06-04T07:49:31.85Z" },
    { url = "https://files.pythonhosted.org/packages/e3/cc/7b1ec671775756020a0ee7f4feeaf3c568f0ab86bd3900088cf986937a92/pywin32-312-cp312-cp312-win_ar
m64.whl", hash = "sha256:6017c58e12f6809fbb0555b75df144c2922a9ffd18e4b9b5afa863b6c1a9d950", size = 6727901, upload-time = "2026-06-04T07:49:34.244Z" }
,
    { url = "https://files.pythonhosted.org/packages/2d/41/12fbfd7f36ed2146d8bc9de96c2741296bf0d490b98508496cff322e274c/pywin32-312-cp313-cp313-win32.
whl", hash = "sha256:7a27df850933d16a8eabfbaeb73d52b273e2da667f80d70b01a89d1f6828d02c", size = 6370184, upload-time = "2026-06-04T07:49:36.253Z" },
    { url = "https://files.pythonhosted.org/packages/ba/db/36a78e3403099d31d9746d13fdcde5accc43c1155f375a34d15983a479a7/pywin32-312-cp313-cp313-win_am
d64.whl", hash = "sha256:c53e878d15a1c44788082bfe712a905433473aa38f86375b7cf8b45e3acbaaf9", size = 6914298, upload-time = "2026-06-04T07:49:38.876Z" }
,
    { url = "https://files.pythonhosted.org/packages/84/37/c1697194092b76de9ed47ca124323f02c57ffc8a45c06f88a3d5acaf01eb/pywin32-312-cp313-cp313-win_ar
m64.whl", hash = "sha256:59aba5d5940842075343a5ddc6b11f1cdf0d1567fe745290359dfbcc7c2eb831", size = 6727640, upload-time = "2026-06-04T07:49:41.083Z" }
,
    { url = "https://files.pythonhosted.org/packages/fc/2b/1f3cded5822fd49c02f40544cbb5f58c7cfd6b1694869fd476cb6170ee97/pywin32-312-cp314-cp314-win32.
whl", hash = "sha256:a77a90fbb6881238d2ca9c6fd797b25817f3768fe78d214a90137ff055a75f5b", size = 6468928, upload-time = "2026-06-04T07:49:43.188Z" },
    { url = "https://files.pythonhosted.org/packages/21/82/3bf86d2e280890201313 2e1ce905a7da0da53790f3836c64bf44d55e24f3/pywin32-312-cp314-cp314-win_am
d64.whl", hash = "sha256:a4dd3a848290ef724347b19f301045831d8e802fa4464f491b98b1e0a081432e", size = 7024157, upload-time = "2026-06-04T07:49:45.34Z" },
    { url = "https://files.pythonhosted.org/packages/a4/0e/73f6d6800b4f27655abd9e9f6aaeaefcddb2b946e4674efa2bab184a7f7b/pywin32-312-cp314-cp314-win_ar
m64.whl", hash = "sha256:9fce94568364e0155e6dfb781ac5d95903be8baf28670632beab1b523f300daa", size = 6839598, upload-time = "2026-06-04T07:49:47.613Z" }
,
    { url = "https://files.pythonhosted.org/packages/eb/61/caa39686032d2ebdd04ff0ab5cbe163126c0066d98e00c9018646e42393b/pywin32-312-cp315-cp315-win32.
whl", hash = "sha256:5c1fbe4a937a73ae9297384a3da38518cbc694c68ad8a809b2e19acd350f03ed", size = 6471159, upload-time = "2026-06-04T07:49:50.035Z" },
    { url = "https://files.pythonhosted.org/packages/0f/cd/7e1de64a4a6f69c04214169657ccab0d93a670ea50e35eb8f489d7378249/pywin32-312-cp315-cp315-win_am
d64.whl", hash = "sha256:c2f03a0f73f804a13c2735b99392b0cd426bb4f2c4d0178e5ac966a0f21618d5", size = 7025293, upload-time = "2026-06-04T07:49:54.857Z" }
,
    { url = "https://files.pythonhosted.org/packages/23/ed/4532e9388e65fa16b46776ef47ad631a64eda1631884488af707666350ed/pywin32-312-cp315-cp315-win_ar
m64.whl", hash = "sha256:a8597d28f267b39074aef51fa593530082b39cbe5a074226096857b1fed2dfb9", size = 6840337, upload-time = "2026-06-04T07:49:57.531Z" }
,
]

[[package]]
name = "pyyaml"
version = "6.0.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/05/8e/961c0007c59b8dd7729d542c61a4d537767a59645b82a0b521206e1e25c2/pyyaml-6.0.3.tar.gz", hash
 = "sha256:d76623373421df22fb4cf8817020cbb7ef15c725b9d5e45f17e189bfc384190f", size = 130960, upload-time = "2025-09-25T21:33:16.546Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d1/33/422b98d2195232ca1826284a76852ad5a86fe23e31b009c9886b2d0fb8b2/pyyaml-6.0.3-cp312-cp312-macos
x_10_13_x86_64.whl", hash = "sha256:7f047e29dcae44602496db43be01ad42fc6f1cc0d8cd6c83d342306c32270196", size = 182063, upload-time = "2025-09-25T21:32:
11.445Z" },
    { url = "https://files.pythonhosted.org/packages/89/a0/6cf41a19a1f2f3feab0e9c0b74134aa2ce6849093d5517a0c550fe37a648/pyyaml-6.0.3-cp312-cp312-macos
x_11_0_arm64.whl", hash = "sha256:fc09d0aa354569bc501d4e787133afc08552722d3ab34836a80547331bb5d4a0", size = 173973, upload-time = "2025-09-25T21:32:12
.492Z" },
    { url = "https://files.pythonhosted.org/packages/ed/23/7a778b6bd0b9a8039df8b1b1d80e2e2ad78aa04171592c8a5c43a56a6af4/pyyaml-6.0.3-cp312-cp312-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:9149cad251584d5fb4981be1ecde53a1ca46c891a79788c0df828d2f166bda28",
 size = 775116, upload-time = "2025-09-25T21:32:13.652Z" },
    { url = "https://files.pythonhosted.org/packages/65/30/d7353c338e12baef4ecc1b09e877c1970bd3382789c159b4f89d6a70dc09/pyyaml-6.0.3-cp312-cp312-manyl
inux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:5fdec68f91a0c6739b380c83b951e2c72ac0197ace422360e6d5a959d8d97b2c", size
= 844011, upload-time = "2025-09-25T21:32:15.21Z" },
    { url = "https://files.pythonhosted.org/packages/8b/9d/b3589d3877982d4f2329302ef98a8026e7f4443c765c46cfecc8858c6b4b/pyyaml-6.0.3-cp312-cp312-manyl
inux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:ba1cc08a7ccde2d2ec775841541641e4548226580ab850948cbfda66a1befcdc", si
ze = 807870, upload-time = "2025-09-25T21:32:16.431Z" },
    { url = "https://files.pythonhosted.org/packages/05/c0/b3be26a015601b822b97d9149ff8cb5ead58c66f981e04fedf4e762f4bd4/pyyaml-6.0.3-cp312-cp312-musll
inux_1_2_aarch64.whl", hash = "sha256:8dc52c23056b9ddd46818a57b78404882310fb473d63f17b07d5c40421e47f8e", size = 761089, upload-time = "2025-09-25T21:3
2:17.56Z" },
    { url = "https://files.pythonhosted.org/packages/be/8e/98435a21d1d4b46590d5459a22d88128103f8da4c2d4cb8f14f2a96504e1/pyyaml-6.0.3-cp312-cp312-musll
inux_1_2_x86_64.whl", hash = "sha256:41715c910c881bc081f1e8872880d3c650acf13dfa8214bad49ed4cede7c34ea", size = 790181, upload-time = "2025-09-25T21:32
:18.834Z" },
    { url = "https://files.pythonhosted.org/packages/74/93/7baea19427dcfbe1e5a372d81473250b379f04b1bd3c4c5ff825e2327202/pyyaml-6.0.3-cp312-cp312-win32
.whl", hash = "sha256:96b533f0e99f6579b3d4d4995707cf36df9100d67e0c8303a0c55b27b5f99bc5", size = 137658, upload-time = "2025-09-25T21:32:20.209Z" },
    { url = "https://files.pythonhosted.org/packages/86/bf/899e81e4cce32febab4fb42bb97dcdf66bc135272882d1987881a4b519e9/pyyaml-6.0.3-cp312-cp312-win_a
md64.whl", hash = "sha256:5fcd34e47f6e0b794d17de1b4ff496c00986e1c83f7ab2fb8fcfe9616ff7477b", size = 154003, upload-time = "2025-09-25T21:32:21.167Z" }
,
    { url = "https://files.pythonhosted.org/packages/1a/08/67bd04656199bbb51dbed1439b7f27601dfb576fb864099c7ef0c3e55531/pyyaml-6.0.3-cp312-cp312-win_a
rm64.whl", hash = "sha256:64386e5e707d03a7e172c0701abfb7e10f0fb753ee1d773128192742712a98fd", size = 140344, upload-time = "2025-09-25T21:32:22.617Z" }
,
    { url = "https://files.pythonhosted.org/packages/d1/11/0fd08f8192109f7169db964b5707a2f1e8b745d4e239b784a5a1dd80d1db/pyyaml-6.0.3-cp313-cp313-macos
x_10_13_x86_64.whl", hash = "sha256:8da9669d359f02c0b91ccc01cac4a67f16afec0dac22c2ad09f46bee0697eba8", size = 181669, upload-time = "2025-09-25T21:32:
23.673Z" },
    { url = "https://files.pythonhosted.org/packages/b1/16/95309993f1d3748cd644e02e38b75d50cbc0d9561d21f390a76242ce073f/pyyaml-6.0.3-cp313-cp313-macos
x_11_0_arm64.whl", hash = "sha256:2283a07e2c21a2aa78d9c4442724ec1eb15f5e42a723b99cb3d822d48f5f7ad1", size = 173252, upload-time = "2025-09-25T21:32:25
.149Z" },
    { url = "https://files.pythonhosted.org/packages/50/31/b20f376d3f810b9b2371e72ef5adb33879b25edb7a6d072cb7ca0c486398/pyyaml-6.0.3-cp313-cp313-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:ee2922902c45ae8ccada2c5b501ab86c36525b883eff4255313a253a3160861c",
 size = 767081, upload-time = "2025-09-25T21:32:26.575Z" },
    { url = "https://files.pythonhosted.org/packages/49/1e/a55ca81e949270d5d4432fbbd19dfea5321eda7c41a849d443dc92fd1ff7/pyyaml-6.0.3-cp313-cp313-manyl
inux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:a33284e20b78bd4a18c8c2282d549d10bc8408a2a7ff57653c0cf0b9be0afce5", size
= 841159, upload-time = "2025-09-25T21:32:27.727Z" },
    { url = "https://files.pythonhosted.org/packages/74/27/e5b8f34d02d9995b80abcef563ea1f8b56d20134d8f4e5e81733b1feceb2/pyyaml-6.0.3-cp313-cp313-manyl
inux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:0f29edc409a6392443abf94b9cf89ce99889a1dd5376d94316ae5145dfedd5d6", si
ze = 801626, upload-time = "2025-09-25T21:32:28.878Z" },
    { url = "https://files.pythonhosted.org/packages/f9/11/ba845c23988798f40e52ba45f34849aa8a1f2d4af4b798588010792ebad6/pyyaml-6.0.3-cp313-cp313-musll
inux_1_2_aarch64.whl", hash = "sha256:f7057c9a337546edc7973c0d3ba84ddcdf0daa14533c2065749c9075001090e6", size = 753613, upload-time = "2025-09-25T21:3
2:30.178Z" },
    { url = "https://files.pythonhosted.org/packages/3d/e0/7966e1a7bfc0a45bf0a7fb6b98ea03fc9b8d84fa7f2229e9659680b69ee3/pyyaml-6.0.3-cp313-cp313-musll
inux_1_2_x86_64.whl", hash = "sha256:eda16858a3cab07b80edaf74336ece1f986ba330fdb8ee0d6c0d68fe82bc96be", size = 794115, upload-time = "2025-09-25T21:32
:31.353Z" },
    { url = "https://files.pythonhosted.org/packages/de/94/980b50a6531b3019e45ddeada0626d45fa85cbe22300844a7983285bed3b/pyyaml-6.0.3-cp313-cp313-win32
.whl", hash = "sha256:d0eae10f8159e8fdad514efdc92d74fd8d682c933a6dd088030f3834bc8e6b26", size = 137427, upload-time = "2025-09-25T21:32:32.58Z" },
    { url = "https://files.pythonhosted.org/packages/97/c9/39d5b874e8b28845e4ec2202b5da735d0199dbe5b8fb85f91398814a9a46/pyyaml-6.0.3-cp313-cp313-win_a
md64.whl", hash = "sha256:79005a0d97d5ddabfeeea4cf676af11e647e41d81c9a7722a193022accdb6b7c", size = 154090, upload-time = "2025-09-25T21:32:33.659Z" }
,
    { url = "https://files.pythonhosted.org/packages/73/e8/2bdf3ca2090f68bb3d75b44da7bbc71843b19c9f2b9cb9b0f4ab7a5a4329/pyyaml-6.0.3-cp313-cp313-win_a
rm64.whl", hash = "sha256:5498cd1645aa724a7c71c8f378eb29ebe23da2fc0d7a08071d89469bf1d2defb", size = 140246, upload-time = "2025-09-25T21:32:34.663Z" }
,
    { url = "https://files.pythonhosted.org/packages/9d/8c/f4bd7f6465179953d3ac9bc44ac1a8a3e6122cf8ada906b4f96c60172d43/pyyaml-6.0.3-cp314-cp314-macos
x_10_13_x86_64.whl", hash = "sha256:8d1fab6bb153a416f9aeb4b8763bc0f22a5586065f86f7664fc23339fc1c1fac", size = 181814, upload-time = "2025-09-25T21:32:
35.712Z" },
    { url = "https://files.pythonhosted.org/packages/bd/9c/4d95bb87eb2063d20db7b60faa3840c1b18025517ae857371c4dd55a6b3a/pyyaml-6.0.3-cp314-cp314-macos
x_11_0_arm64.whl", hash = "sha256:34d5fcd24b8445fadc33f9cf348c1047101756fd760b4dacb5c3e99755703310", size = 173809, upload-time = "2025-09-25T21:32:36
.789Z" },
    { url = "https://files.pythonhosted.org/packages/92/b5/47e807c2623074914e29dabd16cbbdd4bf5e9b2db9f8090fa64411fc5382/pyyaml-6.0.3-cp314-cp314-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:501a031947e3a9025ed4405a168e6ef5ae3126c59f90ce0cd6f2bfc477be31b7",
 size = 766454, upload-time = "2025-09-25T21:32:37.966Z" },
    { url = "https://files.pythonhosted.org/packages/02/9e/e5e9b168be58564121efb3de6859c452fccde0ab093d8438905899a3a483/pyyaml-6.0.3-cp314-cp314-manyl
inux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:b3bc83488de33889877a0f2543ade9f70c67d66d9ebb4ac959502e12de895788", size
```

```
= 836355, upload-time = "2025-09-25T21:32:39.178Z" },
    { url = "https://files.pythonhosted.org/packages/88/f9/16491d7ed2a919954993e48aa941b200f38040928474c9e85ea9e64222c3/pyyaml-6.0.3-cp314-cp314-manyl
inux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:c458b6d084f9b935061bc36216e8a69a7e293a2f1e68bf956dcd9e6cbcd143f5", si
ze = 794175, upload-time = "2025-09-25T21:32:40.865Z" },
    { url = "https://files.pythonhosted.org/packages/dd/3f/5989debef34dc6397317802b527dbbafb2b4760878a53d4166579111411e/pyyaml-6.0.3-cp314-cp314-musll
inux_1_2_aarch64.whl", hash = "sha256:7c6610def4f163542a622a73fb39f534f8c101d690126992300bf3207eab9764", size = 755228, upload-time = "2025-09-25T21:3
2:42.084Z" },
    { url = "https://files.pythonhosted.org/packages/d7/ce/af88a49043cd2e265be63d083fc75b27b6ed062f5f9fd6cdc223ad62f03e/pyyaml-6.0.3-cp314-cp314-musll
inux_1_2_x86_64.whl", hash = "sha256:5190d403f121660ce8d1d2c1bb2ef1bd05b5f68533fc5c2ea899bd15f4399b35", size = 789194, upload-time = "2025-09-25T21:32
:43.362Z" },
    { url = "https://files.pythonhosted.org/packages/23/20/bb6982b26a40bb43951265ba29d4c246ef0ff59c9fdcdf0ed04e0687de4d/pyyaml-6.0.3-cp314-cp314-win_a
md64.whl", hash = "sha256:4a2e8cebe2ff6ab7d1050ecd59c25d4c8bd7e6f400f5f82b96557ac0abafd0ac", size = 156429, upload-time = "2025-09-25T21:32:57.844Z" }
,
    { url = "https://files.pythonhosted.org/packages/f4/f4/a4541072bb9422c8a883ab55255f918fa378ecf083f5b85e87fc2b4eda1b/pyyaml-6.0.3-cp314-cp314-win_a
rm64.whl", hash = "sha256:93dda82c9c22deb0a405ea4dc5f2d0cda384168e466364dec6255b293923b2f3", size = 143912, upload-time = "2025-09-25T21:32:59.247Z" }
,
    { url = "https://files.pythonhosted.org/packages/7c/f9/07dd09ae774e4616edf6cda684ee78f97777bdd15847253637a6f052a62f/pyyaml-6.0.3-cp314-cp314t-maco
sx_10_13_x86_64.whl", hash = "sha256:02893d100e99e03eda1c8fd5c441d8c60103fd175728e23e431db1b589cf5ab3", size = 189108, upload-time = "2025-09-25T21:32
:44.377Z" },
    { url = "https://files.pythonhosted.org/packages/4e/78/8d08c9fb7ce09ad8c38ad533c1191cf27f7ae1effe5bb9400a46d9437fcf/pyyaml-6.0.3-cp314-cp314t-maco
sx_11_0_arm64.whl", hash = "sha256:c1ff362665ae507275af2853520967820d9124984e0f7466736aea23d8611fba", size = 183641, upload-time = "2025-09-25T21:32:4
5.407Z" },
    { url = "https://files.pythonhosted.org/packages/7b/5b/3babb19104a46945cf816d047db2788bcaf8c94527a805610b0289a01c6b/pyyaml-6.0.3-cp314-cp314t-many
linux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:6adc77889b628398debc7b65c073bcb99c4a0237b248cacaf3fe8a557563ef6c"
, size = 831901, upload-time = "2025-09-25T21:32:48.83Z" },
    { url = "https://files.pythonhosted.org/packages/8b/cc/dff0684d8dc44da4d22a13f35f073d558c268780ce3c6ba1b87055bb0b87/pyyaml-6.0.3-cp314-cp314t-many
linux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:a80cb027f6b349846a3bf6d73b5e95e782175e52f22108cfa17876aaeff93702", size
 = 861132, upload-time = "2025-09-25T21:32:50.149Z" },
    { url = "https://files.pythonhosted.org/packages/b1/5e/f77dc6b9036943e285ba76b49e118d9ea929885becb0a29ba8a7c75e29fe/pyyaml-6.0.3-cp314-cp314t-many
linux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:00c4bdeba853cc34e7dd471f16b4114f4162dc03e6b7afcc2128711f0eca823c", s
ize = 839261, upload-time = "2025-09-25T21:32:51.808Z" },
    { url = "https://files.pythonhosted.org/packages/ce/88/a9db1376aa2a228197c58b37302f284b5617f56a5d959fd1763fb1675ce6/pyyaml-6.0.3-cp314-cp314t-musl
linux_1_2_aarch64.whl", hash = "sha256:66e1674c3ef6f541c35191caae2d429b967b99e02040f5ba928632d9a7f0f065", size = 805272, upload-time = "2025-09-25T21:
32:52.941Z" },
    { url = "https://files.pythonhosted.org/packages/da/92/1446574745d74df0c92e6aa4a7b0b3130706a4142b2d1a5869f2eaa423c6/pyyaml-6.0.3-cp314-cp314t-musl
linux_1_2_x86_64.whl", hash = "sha256:16249ee61e95f858e83976573de0f5b2893b3677ba71c9dd36b9cf8be9ac6d65", size = 829923, upload-time = "2025-09-25T21:3
2:54.537Z" },
    { url = "https://files.pythonhosted.org/packages/f0/7a/1c7270340330e575b92f397352af856a8c06f230aa3e76f86b39d01b416a/pyyaml-6.0.3-cp314-cp314t-win_
amd64.whl", hash = "sha256:4ad1906908f2f5ae4e5a8ddfce73c320c2a1429ec52eafd27138b7f1cbe341c9", size = 174062, upload-time = "2025-09-25T21:32:55.767Z"
},
    { url = "https://files.pythonhosted.org/packages/f1/12/de94a39c2ef588c7e6455cfbe7343d3b2dc9d6b6b2f40c4c6565744c873d/pyyaml-6.0.3-cp314-cp314t-win_
arm64.whl", hash = "sha256:ebc55a14a21cb14062aa4162f906cd962b28e2e9ea38f9b4391244cd8de4ae0b", size = 149341, upload-time = "2025-09-25T21:32:56.828Z"
},
]

[[package]]
name = "redis"
version = "8.1.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a8/99/604f0b666d4c616d891cf77ebb9db6bb21601344c051aebf1b72b9ff915f/redis-8.1.0.tar.gz", hash
= "sha256:6e1a19beef9225c83efd689c7e6b7da2d5215b1f42cd13b7fc3714d0a09c7b25", size = 5254356, upload-time = "2026-07-30T08:51:00.269Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/66/9d/c5731f6e3608663d4d3656fd8d3aecee8b509c3082818f5a13eae925baea/redis-8.1.0-py3-none-any.whl",
 hash = "sha256:a4fe1aac3d3b3cc791d4b3d5931c5a956045dc951ee74d1c913ee3ac4d2ee9fb", size = 560618, upload-time = "2026-07-30T08:50:58.497Z" },
]

[[package]]
name = "referencing"
version = "0.37.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "attrs" },
    { name = "rpds-py" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/22/f5/df4e9027acead3ecc63e50fe1e36aca1523e1719559c499951bb4b53188f/referencing-0.37.0.tar.gz"
, hash = "sha256:44aefc3142c5b842538163acb373e24cce6632bd54bdb01b21ad5863489f50d8", size = 78036, upload-time = "2025-10-13T15:30:48.871Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/2c/58/ca301544e1fa93ed4f80d724bf5b194f6e4b945841c5bfd555878eea9fcb/referencing-0.37.0-py3-none-an
y.whl", hash = "sha256:381329a9f99628c9069361716891d34ad94af76e461dcb0335825aecc7692231", size = 26766, upload-time = "2025-10-13T15:30:47.625Z" },
]

[[package]]
name = "regex"
version = "2026.7.19"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/20/98/04b13f1ddfb63158025291c02e03eb42fbb7acb51d091d541050eb4e35e8/regex-2026.7.19.tar.gz", h
ash = "sha256:7e77b324909c1617cbb4c668677e2c6ae13f44d7c1de0d4f15f2e3c10f3315b5", size = 416440, upload-time = "2026-07-19T00:19:48.923Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/3b/b9/d11d7e501ac8fd7d617684423ebb9561e0b998481c1e4cbc0cb212c5d74a/regex-2026.7.19-cp312-cp312-ma
cosx_10_13_universal2.whl", hash = "sha256:2cc3460cedf7579948486eab03bc9ad7089df4d7281c0f47f4afe03e8d13f02d", size = 496778, upload-time = "2026-07-19
T00:17:05.677Z" },
    { url = "https://files.pythonhosted.org/packages/3f/a9/a5ab6f312f24318019170dc485d5421fe4f89e43a98640da50d95a8a7041/regex-2026.7.19-cp312-cp312-ma
cosx_10_13_x86_64.whl", hash = "sha256:0e9554c8785eac5cffe6300f69a91f58ba72bc88a5f8d661235ad7c6aa5b8ccd", size = 297122, upload-time = "2026-07-19T00:
17:07.59Z" },
    { url = "https://files.pythonhosted.org/packages/b3/63/4cab4d7f2d384a144d420b763d97674cb70619c878ea6fcd7640d0e62143/regex-2026.7.19-cp312-cp312-ma
cosx_11_0_arm64.whl", hash = "sha256:d7da47a0f248977f08e2cb659ff3c17ddc13a4d39b3a7baa0a81bf5b415430f6", size = 292009, upload-time = "2026-07-19T00:17
:09.648Z" },
    { url = "https://files.pythonhosted.org/packages/22/85/102a81b218298957d4ea7d2f084fae537a71add9d6ff93c8e67284c5f45e/regex-2026.7.19-cp312-cp312-ma
nylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:93db40c8de0815baab96a06e08a984bac71f989d13bab789e382158c5d42679
7", size = 796708, upload-time = "2026-07-19T00:17:11.542Z" },
    { url = "https://files.pythonhosted.org/packages/78/b5/dc136af5629938a037cd2b304c12240e132ec92f38be8ff9cc89af2a1f2d/regex-2026.7.19-cp312-cp312-ma
nylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:66bd62c59a5427746e8c44becae1d9b99d22fb13f30f492083dfb9ad7c45cc1
8", size = 865651, upload-time = "2026-07-19T00:17:13.312Z" },
    { url = "https://files.pythonhosted.org/packages/e0/75/67402ae3cd9c8c988a4c805d15ee3eef015e7ca4cb112cf3e640fc1f4153/regex-2026.7.19-cp312-cp312-ma
nylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:1649eb39fcc9ea80c4d2f110fde2b8ab2aef3877b98f02ab9b14e961f418c511", si
ze = 911756, upload-time = "2026-07-19T00:17:15.015Z" },
    { url = "https://files.pythonhosted.org/packages/2a/8e/096d00c7c480ef2ff4265349b14e2261d4ab787ba1f74e2e80d1c58079c3/regex-2026.7.19-cp312-cp312-ma
nylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9dce8ec9695f531a1b8a6f314fd4b393adcccf2ea861db480cdf97a301d01a68",
 size = 801798, upload-time = "2026-07-19T00:17:17.208Z" },
    { url = "https://files.pythonhosted.org/packages/f0/41/e7ecac6edb5722417f85cc67eaf386322fbe8acf6918ec2fdc37c20dd9d0/regex-2026.7.19-cp312-cp312-ma
nylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:3080a7fd38ef049bd489e01c970c97dd84ff446a885b0f1f6b26d9b1ad13ce11", size = 776933, upl
oad-time = "2026-07-19T00:17:19.347Z" },
    { url = "https://files.pythonhosted.org/packages/6f/69/03c9b3f058d66403e0ca2c938696e81d51cd4c6d47ec5265f02f96948d9a/regex-2026.7.19-cp312-cp312-mu
sllinux_1_2_aarch64.whl", hash = "sha256:1d793a7988e04fcb1e2e135567443d82173225d657419ec09414a9b5a145b986", size = 784338, upload-time = "2026-07-19T0
```

0:17:21.057Z" },
    { url = "https://files.pythonhosted.org/packages/f6/f7/b38ab3d43f284afbb618fcd15d0e77eb786ae461ce1f6bc7494619ddc0f2/regex-2.6.7.19-cp312-cp312-musllinux_1_2_ppc64le.whl", hash = "sha256:e8b0abe7d870f53ca5143895fef7d1041a0c831a140d3dc2c760dd7ba25d4a8b", size = 860452, upload-time = "2026-07-19T00:17:23.119Z" },
    { url = "https://files.pythonhosted.org/packages/15/5c/ff60ef0571121714f3cf9920bc183071e384a10b556d042e0fdb06cc07a5/regex-2.6.7.19-cp312-cp312-musllinux_1_2_riscv64.whl", hash = "sha256:4e5413bd5f13d3a4e3539ca98f70f75e7fca92518dd7f117f030ebedd10b60cb", size = 765958, upload-time = "2026-07-19T00:17:24.81Z" },
    { url = "https://files.pythonhosted.org/packages/aa/0f/bd34021162c0ab47f9a315bd56cd5642e920c8e5668a75ef6c6a6fca590d/regex-2.6.7.19-cp312-cp312-musllinux_1_2_s390x.whl", hash = "sha256:73b133a9e6fb512858e7f065e96f1180aa46646bc74a83aea62f1d314f3dd035", size = 851765, upload-time = "2026-07-19T00:17:26.993Z" },
    { url = "https://files.pythonhosted.org/packages/2a/20/a2ca43edade0595cccfdc98636739f536d9e26898e7dbddc2b9e98898953/regex-2.6.7.19-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:dbe6493fbd27321b1d1f2dd4f5c7e5bd4d8b1d7cab7f32fd67db3d0b2ed8248a", size = 789714, upload-time = "2026-07-19T00:17:28.699Z" },
    { url = "https://files.pythonhosted.org/packages/5d/47/e02db4015d424fc83c00ea0ac8c5e5ec14397943de9abf909d5ce3a25931/regex-2.6.7.19-cp312-cp312-win32.whl", hash = "sha256:ddd67571c10869f65a5d7dde536d1e066e306cc90de57d7de4d5f34802428bb5", size = 267157, upload-time = "2026-07-19T00:17:31.051Z" },
    { url = "https://files.pythonhosted.org/packages/08/8e/c780c131f79b42ed22d1bd7da4096c2c35f813e835acd02ef0f018bd892c/regex-2.6.7.19-cp312-cp312-win_amd64.whl", hash = "sha256:e30d40268a28d54ce043703175049700 4c22602b8e3ab891f759b795a003b312", size = 277777, upload-time = "2026-07-19T00:17:32.848Z" },
    { url = "https://files.pythonhosted.org/packages/3e/4c/e4d7e086449bdf379d89774bf1f89dc4a41943f3c5a6125a03905b34b5fb/regex-2.6.7.19-cp312-cp312-win_arm64.whl", hash = "sha256:de9208bb427130c82a5dbfd104f92c8876fc9559278c880b3002755bbbe9c83d", size = 277136, upload-time = "2026-07-19T00:17:34.803Z" },
    { url = "https://files.pythonhosted.org/packages/5d/3d/84165e4299ff76f3a40fe1f2abf939e976f693383a08d2beea6af62bd2c1/regex-2.6.7.19-cp313-cp313-macosx_10_13_universal2.whl", hash = "sha256:f035d9dc1d25eff9d361456572231c7d27b5ccd473ca7dc0adfce732bd006d40", size = 496552, upload-time = "2026-07-19T00:17:36.808Z" },
    { url = "https://files.pythonhosted.org/packages/02/a2/a65293e6e4cf28eb7ee1be5335a5386c40d6742e9f47fafc8fec785e16c7/regex-2.6.7.19-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:c42572142ed0b9d5d261ba727157c426510da78e20828b66bbb855098b8a4e38", size = 296983, upload-time = "2026-07-19T00:17:38.816Z" },
    { url = "https://files.pythonhosted.org/packages/95/47/2d0564e93d87bc48618360ddca232a2ca612bbdf53ce8465d45ca5ce14ee/regex-2.6.7.19-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:40b34dd88658e4fedd2fddbf0275ac970d00614b731357f425722a3ed1983d11", size = 291832, upload-time = "2026-07-19T00:17:40.726Z" },
    { url = "https://files.pythonhosted.org/packages/07/cd/42dfbabff3dfc9603c501c0e2e2c5adbb09d127b267bf5348de0af338c15/regex-2.6.7.19-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:0c41c63992bf1874cebb6e7f56fd7d3c007924659a604ae3d90e427d40d4fd13", size = 796775, upload-time = "2026-07-19T00:17:42.382Z" },
    { url = "https://files.pythonhosted.org/packages/df/5d/f6a4839f2b934e3eed5973fd07f5929ee97d4c98939fb275ea23c274ee16/regex-2.6.7.19-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:1d3372064506b94dd2c67c845f2db8062e9e9ba84d04e33cb96d7d33c11fe1ae", size = 865687, upload-time = "2026-07-19T00:17:44.185Z" },
    { url = "https://files.pythonhosted.org/packages/14/b0/b47d6c36049bc59806a50bd4c86ced70bbe058d787f80281b1d7a9b0e024/regex-2.6.7.19-cp313-cp313-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:fce7760bf283405b2c7999cab3da4e72f7deca6396013115e3f7a955db9760da", size = 911962, upload-time = "2026-07-19T00:17:46.442Z" },
    { url = "https://files.pythonhosted.org/packages/2a/be/ff61f28f9273658cfe23acbbac5217221f6519960ed401e61dfdab12bc35/regex-2.6.7.19-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:c0d702548d89d572b2929879bc883bb7a4c4709efafe4512cadee56c55c9bd15", size = 801817, upload-time = "2026-07-19T00:17:48.25Z" },
    { url = "https://files.pythonhosted.org/packages/c3/bb/8b4f7f26b333f9f79e1b453613c39bb4776f51d38ae66dd0ba31d6b354ca/regex-2.6.7.19-cp313-cp313-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:d446c6ac40bb6e05025ccee55b84d80fe9bf8e93010ffc4bb9484f13d498835f", size = 776908, upload-time = "2026-07-19T00:17:50.183Z" },
    { url = "https://files.pythonhosted.org/packages/09/13/610110fc5921d380516d03c26b652555f08aa0d23ea78a771231873c3638/regex-2.6.7.19-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:4c3501bfa814ab07b5580741f9bf78dfdfe146a04057f82df9e2402d2a975939", size = 784426, upload-time = "2026-07-19T00:17:52.454Z" },
    { url = "https://files.pythonhosted.org/packages/ca/f5/1ef9e2a83a5947c57ebff0b377cb5727c3d5ec1992317a320d035cd0dbb6/regex-2.6.7.19-cp313-cp313-musllinux_1_2_ppc64le.whl", hash = "sha256:c4585c3e64b4f9e583b4d2683f18f5d5d872b3d71dcf24594b74ecc23602fa96", size = 860600, upload-time = "2026-07-19T00:17:54.229Z" },
    { url = "https://files.pythonhosted.org/packages/a0/02/073af33a3ec149241d11c80acea91e722aa0adbf05addd50f251c4fe89c3/regex-2.6.7.19-cp313-cp313-musllinux_1_2_riscv64.whl", hash = "sha256:571fde9741eb0ccde23dd4e0c1d50fbae910e901fa7e629faf39b2dda740d220", size = 765950, upload-time = "2026-07-19T00:17:56.041Z" },
    { url = "https://files.pythonhosted.org/packages/81/a9/d1e9f819dc394a568ef370cd56cf25394e957a2235f8370f23b576e5a475/regex-2.6.7.19-cp313-cp313-musllinux_1_2_s390x.whl", hash = "sha256:15b364b9b98d6d2fe1a85034c23a3180ff913f46caddc3895f6fd65186255ccc", size = 851794, upload-time = "2026-07-19T00:17:57.897Z" },
    { url = "https://files.pythonhosted.org/packages/03/3a/8ae83eda7579feacdf984e71fb9e70635fb6f832eeddca58427ec4fca926/regex-2.6.7.19-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:ffd8893ccc1c2fce6e0d6ca402d716fe1b29db70c7132609a05955e31b2aa8f2", size = 789845, upload-time = "2026-07-19T00:17:59.97Z" },
    { url = "https://files.pythonhosted.org/packages/4b/23/c195cbfe5a75fdec64d8f6554fd15237b837919d2c61bdc141d7c807b08b/regex-2.6.7.19-cp313-cp313-win32.whl", hash = "sha256:f0fa4fa9c3632d708742baf2282f2055c11d888a790362670a403cbf48a2c404", size = 267135, upload-time = "2026-07-19T00:18:01.958Z" },
    { url = "https://files.pythonhosted.org/packages/b2/80/a11de8404b7272b70acb45c1c05987cce60b45d5693da2e176f0e390d564/regex-2.6.7.19-cp313-cp313-win_amd64.whl", hash = "sha256:d51ffd3427640fa2da6ade574ceba932f210ad095f65fcc450a2b0a0d454868e", size = 277747, upload-time = "2026-07-19T00:18:04.121Z" },
    { url = "https://files.pythonhosted.org/packages/d1/29/0f5c8eff1b4f1f3d83276d365fccecf666afcc7d947420943bf394d07adb/regex-2.6.7.19-cp313-cp313-win_arm64.whl", hash = "sha256:c670fe7be5b6020b76bc6e8d2196074657e1327595bca93a389e1a76ab130ad8", size = 277129, upload-time = "2026-07-19T00:18:05.821Z" },
    { url = "https://files.pythonhosted.org/packages/dc/4c/44b74742052cedda40f9ae469532a037112f7311a36669a891fba8984bb0/regex-2.6.7.19-cp313-cp313t-macosx_10_13_universal2.whl", hash = "sha256:db47b561c9afd884baa1f96f797c9ca369872c4b65912bc691cfa99e68340af2", size = 501134, upload-time = "2026-07-19T00:18:07.567Z" },
    { url = "https://files.pythonhosted.org/packages/f0/45/bbd038b5e39ee5613a5a689290145b40058cc152c41de9cc23639d2b9734/regex-2.6.7.19-cp313-cp313t-macosx_10_13_x86_64.whl", hash = "sha256:65dcd28d3eba2ab7c2fd906485cc301392b47cc2234790d27d4e4814e02cdfda", size = 299418, upload-time = "2026-07-19T00:18:09.38Z" },
    { url = "https://files.pythonhosted.org/packages/65/38/c5bde94b4cedfd5850d64c3f08222d8e1600e84f6ee71d9b44b4b8163f74/regex-2.6.7.19-cp313-cp313t-macosx_11_0_arm64.whl", hash = "sha256:f2e7f8e2ab6c2922be02c7ec45185aa5bd771e2e57b95455ee343a44d8130dff", size = 294486, upload-time = "2026-07-19T00:18:11.188Z" },
    { url = "https://files.pythonhosted.org/packages/d7/6a/2f5e107cb26c960b781967178899daf2787a7ab151844ed3c01d6fc95474/regex-2.6.7.19-cp313-cp313t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:fe31f28c94402043161876a258a9c6f757cb485905c7614ce8d6cd40e6b7bdc1", size = 811643, upload-time = "2026-07-19T00:18:12.975Z" },
    { url = "https://files.pythonhosted.org/packages/37/d4/a2f963406d7d73a62eed84ba05a258afb6cad1b21aa4517443ce40506b78/regex-2.6.7.19-cp313-cp313t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:f8f6fa298bb4f7f58a33334406218ba74716e68feddf5e4e54cd5d8082705abf", size = 871081, upload-time = "2026-07-19T00:18:14.733Z" },
    { url = "https://files.pythonhosted.org/packages/45/a3/44be546340bedb15f13063f5e7fe16793ea4d9ea2e805d09bd174ac27724/regex-2.6.7.19-cp313-cp313t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:cc1b2440423a851fad781309dd87843868f4f66a6bcd1ddb9225cf4ec2c84732", size = 917372, upload-time = "2026-07-19T00:18:16.724Z" },
    { url = "https://files.pythonhosted.org/packages/f8/f6/e0870b0fd2a40dba0074e4b76e514b21313d37946c9248453e34ec43923e/regex-2.6.7.19-cp313-cp313t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:8ac59a0900474a52b7c04af8196affc22bd9842acb0950df12f7b813e983609a", size = 816089, upload-time = "2026-07-19T00:18:18.617Z" },
    { url = "https://files.pythonhosted.org/packages/ae/27/957e8e22690ad6634572b39b71f130a6105f4d0718bb16849eac00fff147/regex-2.6.7.19-cp313-cp313t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:4896db1f4ce0576765b8272aa922df324e0f5b9bb2c3d03044ff32a7234a9aba", size = 785206, upload-time = "2026-07-19T00:18:20.464Z" },
    { url = "https://files.pythonhosted.org/packages/76/a4/186e410941e731037c01166069ab86da9f65e8f8110c18009ccf4bd623ee/regex-2.6.7.19-cp313-cp313t-musllinux_1_2_aarch64.whl", hash = "sha256:4e6883a021db30511d9fb8cfb0f222ce1f2c369f7d4d8b0448f449a93ba0bdfc", size = 800431, upload-time = "2026-07-19T00:18:22.716Z" },
    { url = "https://files.pythonhosted.org/packages/73/9f/e4e10e023d291d64a33e246610b724493bf1ce98e0e59c9b7c837e5acfb7/regex-2.6.7.19-cp313-cp313t-musllinux_1_2_ppc64le.whl", hash = "sha256:09523a592938aa9f587fb74467c63ff0cf88fc3df14c82ab0f0517dcf76aaa62", size = 864906, upload-time = "2026-07-19T00:18:24.772Z" },
    { url = "https://files.pythonhosted.org/packages/24/57/ccb20b6be5f1f52a053d1ba2a8f7a077edb9d918248b8490d7506c6832b3/regex-2.6.7.19-cp313-cp313t-musllinux_1_2_riscv64.whl", hash = "sha256:1ebac3474b8589fce2f9b225b650afd61448f7c73a5d0255a10cc6366471aed1", size = 773559, upload-time = "2026-07-19T00:18:27.008Z" },
    { url = "https://files.pythonhosted.org/packages/a3/82/f3b263cf8fad927dc102891da8502e718b7ff9d19af7a2a07c03865d7188/regex-2.6.7.19-cp313-cp313t-musllinux_1_2_s390x.whl", hash = "sha256:4a0530bb1b8c1c985e7e2122e2b4d3aedd8a3c21c6bfddae6767c4405668b56e", size = 857739, upload-time = "2026-07-19T00:18:29.107Z" },
    { url = "https://files.pythonhosted.org/packages/47/2e/1687bd1b6c2aed5e672ccf845fc11557821fe7366d921b50889ea5ce57bf/regex-2.6.7.19-cp313-cp313t-m

usllinux_1_2_x86_64.whl", hash = "sha256:2ef7eeb108c47ce7bcc9513e51bcb1bf57e8f483d52fce68a8642e3527141ae0", size = 804522, upload-time = "2026-07-19T00:18:31.362Z" },
    { url = "https://files.pythonhosted.org/packages/76/7c/cc4e7655181b2d9235b704f2c5e19d8eff002bbc437bae59baee0e381aca/regex-2026.7.19-cp313-cp313t-win32.whl", hash = "sha256:64b6ca7391a1395c2638dd5c7456d67bea44fc6c5e8e92c5dc8aa6a8f23292b4", size = 269141, upload-time = "2026-07-19T00:18:33.479Z" },
    { url = "https://files.pythonhosted.org/packages/bb/14/961b4c7b05a2391c32dbc85e27773076671ef8f97f36cec70fe414734c02/regex-2026.7.19-cp313-cp313t-win_amd64.whl", hash = "sha256:f04b9f56b0e0614c0126be12c2c2d9f8850c1e57af302bd0a63bed379d4af974", size = 280036, upload-time = "2026-07-19T00:18:35.419Z" },
    { url = "https://files.pythonhosted.org/packages/ce/67/795644550d788ddbb6dc458c95895f8009978ea6d6ea76b005eb3f45e8c9/regex-2026.7.19-cp313-cp313t-win_arm64.whl", hash = "sha256:fcee38cd8e5089d6d4f048ba1233b3ad76e5954f545382180889112ff5cb712d", size = 279394, upload-time = "2026-07-19T00:18:37.454Z" },
    { url = "https://files.pythonhosted.org/packages/d2/25/0c4c452f8ef3efe456745b2f33195f5904b573fb4c2ff3f0cb9ec188461e/regex-2026.7.19-cp314-cp314-macosx_10_13_universal2.whl", hash = "sha256:a81758ed242b861b72e778ba34d41366441a2e10b16b472784c88da2dea7e2dd", size = 496750, upload-time = "2026-07-19T00:18:39.633Z" },
    { url = "https://files.pythonhosted.org/packages/24/9e/b70ca6c1704f6c7cd32a9e143c86cc5968d10981eca284bad670c245ea7d/regex-2026.7.19-cp314-cp314-macosx_10_13_x86_64.whl", hash = "sha256:4aa5435cdb3eb6f55fe98a171b05e3fbcd95fadaa4aa32acf62afd9b0cfdbcac", size = 297093, upload-time = "2026-07-19T00:18:41.583Z" },
    { url = "https://files.pythonhosted.org/packages/87/74/0b692da2520d51fbff19c88b83d97e4c702909dd02386c585998b7e2dbed/regex-2026.7.19-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:60be8693a1dadc210bbcbc0db3e26da5f7d01d1d5a3da594e99b4fa42df404f5", size = 292043, upload-time = "2026-07-19T00:18:43.347Z" },
    { url = "https://files.pythonhosted.org/packages/e3/a7/1d478e614016045a33feae57446215f9fd65b665a5ceb2f891fb3183bc52/regex-2026.7.19-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:d19662dbedbe783d323196312d38f5ba53cf562963782521719 85da6899887d3", size = 797214, upload-time = "2026-07-19T00:18:45.362Z" },
    { url = "https://files.pythonhosted.org/packages/aa/ae/11b9c9411d92c30e3d2db32df5a31133e4a99a8fc397a604fd08f6c4bffb/regex-2026.7.19-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:d15df07081d91b76ff20d43f94592ee110330152d617b730fdbe5ef9fb68005 3", size = 866433, upload-time = "2026-07-19T00:18:47.315Z" },
    { url = "https://files.pythonhosted.org/packages/b1/62/2b2efc4992f91d6d204b24c647c9f9412e85379d92b7c0ab9fdae622327e/regex-2026.7.19-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:56ad4d9f77df871a99e25c37091052a02528ec0eb059de928ee33956b854b45b", size = 911360, upload-time = "2026-07-19T00:18:49.588Z" },
    { url = "https://files.pythonhosted.org/packages/14/71/986ceea9aa3da548bf1357cad89b63915ec6d21ec957c8113b29ece567df/regex-2026.7.19-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:7322ec6cc9fba9d49ab888bb82d67ac5625627aa168f0165139b17018df3fb8a", size = 801275, upload-time = "2026-07-19T00:18:51.767Z" },
    { url = "https://files.pythonhosted.org/packages/15/be/ce9d9534b2cda96eab32c548261224b9b4e220a4126f098f60f42ae7b4cd/regex-2026.7.19-cp314-cp314-manylinux2014_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:9c7472192ebfad53a6be7c4a8bfb2d64b81c0e93a1fc8c57e1dd0b638297b5d1", size = 777131, upload-time = "2026-07-19T00:18:54.053Z" },
    { url = "https://files.pythonhosted.org/packages/61/2b/58b5c710f2c3929515a25f3a1ca0dad0dcd4518d4fff3cf23bc7adb8dcd2/regex-2026.7.19-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:c10b82c2634df08dfb13b1f04e38fe310d086ee092f4f69c0c8da234251e556e", size = 785020, upload-time = "2026-07-19T00:18:56.579Z" },
    { url = "https://files.pythonhosted.org/packages/84/03/5fe091935b74f15fe0f97998c215cae418d1c0413f6258c7d4d2e83aa37f/regex-2026.7.19-cp314-cp314-musllinux_1_2_ppc64le.whl", hash = "sha256:17ed5692f6acc4183e98331101a5f9e4f64d72fe58b753da4d444a2c77d05b12", size = 861263, upload-time = "2026-07-19T00:18:58.64Z" },
    { url = "https://files.pythonhosted.org/packages/d8/fa/d60bf82e10841eef62a9e32aac401468f05fddfbcb2942e342b1ba3d2433/regex-2026.7.19-cp314-cp314-musllinux_1_2_riscv64.whl", hash = "sha256:22a992de9a0d91bda927bf02b94351d737a0302905432c88a53de7c4b9ce62e2", size = 766199, upload-time = "2026-07-19T00:19:00.705Z" },
    { url = "https://files.pythonhosted.org/packages/bf/5d/11e64d151b0662b81d6bf644c74dc118d461df85bdf2577fadbbf751788a/regex-2026.7.19-cp314-cp314-musllinux_1_2_s390x.whl", hash = "sha256:618a0aed532be87294c4477b0481f3aa0f1520f4014a4374dd4cf789b4cd2c97", size = 851317, upload-time = "2026-07-19T00:19:03.015Z" },
    { url = "https://files.pythonhosted.org/packages/7c/34/532efb87488d90807bae6a443d357ee5e2728a478c597619c8aaa17cc0bd/regex-2026.7.19-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:2ce9e679f776649746729b6c86382da519ef649c8e34cc41df0d2e5e0f6c36d4", size = 789557, upload-time = "2026-07-19T00:19:05.338Z" },
    { url = "https://files.pythonhosted.org/packages/d6/90/3a8d5ca977171ec3ae21a71207d2228b2663bde14d7f7ef0e6363ecf9290/regex-2026.7.19-cp314-cp314-win32.whl", hash = "sha256:73f272fba87b6ccfe70a137d02a54af386f6d27aa509fbffdd978f5947aae1aa", size = 272531, upload-time = "2026-07-19T00:19:07.487Z" },
    { url = "https://files.pythonhosted.org/packages/96/e1/8862885e70409de70e8c005f57fb2e7be8d9ef0317250d60f4c9660a300d/regex-2026.7.19-cp314-cp314-win_amd64.whl", hash = "sha256:d721e53758b2cca74990185eb0671dd466d7a388a1a45d0c6f4c13cef41a68ac", size = 280831, upload-time = "2026-07-19T00:19:09.46Z" },
    { url = "https://files.pythonhosted.org/packages/08/82/2693e53e29f9104d9de95d37ce4dd826bd32d5f9c0085d3aa6ac042675c4/regex-2026.7.19-cp314-cp314-win_arm64.whl", hash = "sha256:65fa6cb38ed5e9c3637e68e544f598b39c3b86b808ed0627a67b68320384b459", size = 281099, upload-time = "2026-07-19T00:19:11.398Z" },
    { url = "https://files.pythonhosted.org/packages/92/b7/9a01aa16461a18cde9d7b9c3ab21e501db2ce33725f53014342b91df2b0a/regex-2026.7.19-cp314-cp314t-macosx_10_13_universal2.whl", hash = "sha256:5a2721c8720e2cb3c209925dfb9200199b4b07361c9e01d321719404b21458b3", size = 501121, upload-time = "2026-07-19T00:19:13.425Z" },
    { url = "https://files.pythonhosted.org/packages/f3/5e/bbaeca815dc9191c424c94a4fdc5c87c75748a64a6271821212ebdd4e1a3/regex-2026.7.19-cp314-cp314t-macosx_10_13_x86_64.whl", hash = "sha256:199535629f25caf89698039af3d1ad5fcae7f933e2112c73f1cdf49165c99518", size = 299415, upload-time = "2026-07-19T00:19:15.43Z" },
    { url = "https://files.pythonhosted.org/packages/cd/d6/0dd1a321afaab95eb7ff44aa0f637301786f1dc71c6b797b9ed236ed8890/regex-2026.7.19-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:9b60d7814174f059e5de4ab98271cc5ba9259cfea55273a81544dceea32dc8d9", size = 294483, upload-time = "2026-07-19T00:19:17.879Z" },
    { url = "https://files.pythonhosted.org/packages/92/5f/40bacf91d0904f812e13bbbab3864604c463eced8afdc54aeaa50492ea95/regex-2026.7.19-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:dbece16025afda5e3031af0c4059207e61dcf73ef13af844964f57f387d1c435", size = 811833, upload-time = "2026-07-19T00:19:20.102Z" },
    { url = "https://files.pythonhosted.org/packages/94/7c/4902744261f775aeede8b5627314b38482da29cf49a57b66a6fb753246c5/regex-2026.7.19-cp314-cp314t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:d24ecb4f5e009ea0bd275ee37ad9953b32005e2e5e60f8bbae16da0dbbf0d3a0", size = 871270, upload-time = "2026-07-19T00:19:22.365Z" },
    { url = "https://files.pythonhosted.org/packages/16/70/6980c9be6bf21c0a60ed3e0aea39cf419ecf3b08d1d9947bc56e196ef186/regex-2026.7.19-cp314-cp314t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:8cae6fd77a5b72dae505084b1a2ee0360139faf72fedbab667cd7cc65aae7a6a", size = 917534, upload-time = "2026-07-19T00:19:24.529Z" },
    { url = "https://files.pythonhosted.org/packages/52/92/8b2bd872782ce8c42691e39acb38eb8efe014e5ddb78ad7d943d6f197ce9/regex-2026.7.19-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:9724e6cb5e478cd7d8cabf027826178739cb18cf0e117d0e32814d479fa02276", size = 816135, upload-time = "2026-07-19T00:19:26.919Z" },
    { url = "https://files.pythonhosted.org/packages/de/2d/33a602f657bdc4041f17d79f92ab18261d255d91a06117a6e29df023e5e2/regex-2026.7.19-cp314-cp314t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:572fc57b0009c735ee56c175ea021b637a15551a312f56734277f923d6fd0f6c", size = 785492, upload-time = "2026-07-19T00:19:29.192Z" },
    { url = "https://files.pythonhosted.org/packages/9e/36/0987cf4cb271680064a70d24a475873775a151d0b7058698a006cb0cae4a/regex-2026.7.19-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:20568e182eb82d39a6bf7cff3fd58566f14c75c6f74b2c8c96537eecf9010e3a", size = 800658, upload-time = "2026-07-19T00:19:31.392Z" },
    { url = "https://files.pythonhosted.org/packages/a8/24/c14f31c135e1ba55fa4f9a58ca98d0842512bf6188230763c31c8f449e3b/regex-2026.7.19-cp314-cp314t-musllinux_1_2_ppc64le.whl", hash = "sha256:1d58561843f0ff7dc78b4c28b5e2dc388f3eff94ebc8a232a3adba961fc00009", size = 865073, upload-time = "2026-07-19T00:19:33.485Z" },
    { url = "https://files.pythonhosted.org/packages/14/85/181a12211f22469f24d2de1ebddfe397d2396e2c29013b9a58134a91069a/regex-2026.7.19-cp314-cp314t-musllinux_1_2_riscv64.whl", hash = "sha256:61bb1bd45520aacd56dd80943bd34991fb5350afdd1f36f2282230fd5154a218", size = 773684, upload-time = "2026-07-19T00:19:35.599Z" },
    { url = "https://files.pythonhosted.org/packages/23/58/bd1a0c1a62251366f8d21f41b1ea3c76994962071b8b6ea42f72d505c0f0/regex-2026.7.19-cp314-cp314t-musllinux_1_2_s390x.whl", hash = "sha256:cd3584591ea4429026cdb931b054342c2bcf189b44ff367f8d5c15bc092a2966", size = 857769, upload-time = "2026-07-19T00:19:37.738Z" },
    { url = "https://files.pythonhosted.org/packages/e4/4f/f7e2dad6756b2fe1fe75dd90a628c3b45f249d39f948dd90cd2476325417/regex-2026.7.19-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:5cc26a66e212fa5d6c6170c3a40d99d888db3020c6fdab1523250d4341382e44", size = 804546, upload-time = "2026-07-19T00:19:40.229Z" },
    { url = "https://files.pythonhosted.org/packages/2b/d7/01d31d5bdb09bc026fab77f59a371fdf8f9b292e4810546c56182ca70498/regex-2026.7.19-cp314-cp314t-win32.whl", hash = "sha256:2c4e61e2e1be56f63ec3cc618aa9e0de81ef6f43d177205451840022e24f5b78", size = 274526, upload-time = "2026-07-19T00:19:42.398Z" },
    { url = "https://files.pythonhosted.org/packages/52/0e/cea4ce73bc0a8247a0748228ae6669984c7e1f8134b6fa66e59c0572e0ea/regex-2026.7.19-cp314-cp314t-win_amd64.whl", hash = "sha256:c639ea314df70a7b2811e8020448c75af8c9445f5a60f8a4ced81c306a9380c2", size = 283763, upload-time = "2026-07-19T00:19:44.644Z" },
    { url = "https://files.pythonhosted.org/packages/6f/b6/26e41975febae63b7a6e3e02f32cff6cff2e4f10d19c929082f56aebf7c6/regex-2026.7.19-cp314-cp314t-win_arm64.whl", hash = "sha256:9a15e785f244f3e07847b984ce8773fc3da10a9f3c131cc49a4c5b4d672b4547", size = 283451, upload-time = "2026-07-19T00:19:46.639

```
Z" },
]

[[package]]
name = "requests"
version = "2.34.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "certifi" },
    { name = "charset-normalizer" },
    { name = "idna" },
    { name = "urllib3" },
]
sdist = { url = "https://files.pythonhosted.org/packages/ac/c3/e2a2b89f2d3e2179abd6d00ebd70bff6273f37fb3e0cc209f48b39d00cbf/requests-2.34.2.tar.gz", h
ash = "sha256:f288924cae4e29463698d6d60bc6a4da69c89185ad1e0bcc4104f584e960b9ed", size = 142856, upload-time = "2026-05-14T19:25:27.735Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/a0/f4/c67b0b3f1b9245e8d266f0f112c500d50e5b4e83cb6f3b71b6528104182a/requests-2.34.2-py3-none-any.w
hl", hash = "sha256:2a0d60c172f83ac6ab31e4554906c0f3b3588d37b5cb939b1c061f4907e278e0", size = 73075, upload-time = "2026-05-14T19:25:26.443Z" },
]

[[package]]
name = "requests-toolbelt"
version = "1.0.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "requests" },
]
sdist = { url = "https://files.pythonhosted.org/packages/f3/61/d7545dafb7ac2230c70d38d31cbfe4cc64f7144dc41f6e4e4b78ecd9f5bb/requests-toolbelt-1.0.0.ta
r.gz", hash = "sha256:7681a0a3d047012b5bdc0ee37d7f8f07ebe76ab08caeccfc3921ce23c88d5bc6", size = 206888, upload-time = "2023-05-01T04:11:33.229Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/3f/51/d4db610ef29373b879047326cbf6fa98b6c1969d6f6dc423279de2b1be2c/requests_toolbelt-1.0.0-py2.py
3-none-any.whl", hash = "sha256:cccfdd665f0a24fcf4726e690f65639d272bb0637b9b92dfd91a5568ccf6bd06", size = 54481, upload-time = "2023-05-01T04:11:28.42
7Z" },
]

[[package]]
name = "rpds-py"
version = "2026.6.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/aa/2a/9618a122aeb2a169a28b03889a2995fe297588964333d4a7d67bdf46e147/rpds_py-2026.6.3.tar.gz",
hash = "sha256:1cebd1337c242e4ec2293e541f712b2da849b29f48f0c293684b71c0632625d4", size = 64051, upload-time = "2026-06-30T07:17:53.009Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/5c/be/2e8974163072e7bab7df1a5acd54c4498e75e35d6d18b864d3a9d5dadc92/rpds_py-2026.6.3-cp312-cp312-m
acosx_10_12_x86_64.whl", hash = "sha256:a0811d33247c3d6128a3001d763f2aa056bb3425204335400ac54f89eec3a0d0", size = 343691, upload-time = "2026-06-30T07
:15:14.96Z" },
    { url = "https://files.pythonhosted.org/packages/a4/73/319dfa745dd668efe89309141ded489126461fcecd2b8f3a3cda185129b6/rpds_py-2026.6.3-cp312-cp312-m
acosx_11_0_arm64.whl", hash = "sha256:538949e262e46caa31ac01bdb3c1e8f642622922cacbabbae6a8445d9dc33eaf", size = 338542, upload-time = "2026-06-30T07:1
5:16.267Z" },
    { url = "https://files.pythonhosted.org/packages/21/63/4239893be1c4d09b709b1a8f6be4188f0870084ff547f46606b8a75f1b03/rpds_py-2026.6.3-cp312-cp312-m
anylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:55927d532399c2c646100ff7feb48eaa940ad70f42cd68e1328f3ded9f81ca24", size = 368180, upl
oad-time = "2026-06-30T07:15:17.62Z" },
    { url = "https://files.pythonhosted.org/packages/1c/ca/9c5de382225234ceb37b1844ebdb140db12b2a278bb9efe2fcd19f6c82ce/rpds_py-2026.6.3-cp312-cp312-m
anylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:f56f1695bc5c0871cbc33dc0130fcf503aab0c57dcc5a6700a4f49eba4f2652e", size = 375067, uploa
d-time = "2026-06-30T07:15:18.952Z" },
    { url = "https://files.pythonhosted.org/packages/87/dc/863f69d1bf04ade34b7fe0d59b9fdf6f0135fe2d7cbca74f1d665589559d/rpds_py-2026.6.3-cp312-cp312-m
anylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:270b293dae9058fc9fcedab50f13cebf46fb8ed1d1d54e0521a9da5d6b211975", size = 490509, upl
oad-time = "2026-06-30T07:15:20.434Z" },
    { url = "https://files.pythonhosted.org/packages/ce/ef/eac16a12048b45ec7c7fa94f2be3438a5f26bf9cc8580b18a1cfd609b7f6/rpds_py-2026.6.3-cp312-cp312-m
anylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:127565fead0a10943b282957bd5447804ff3160ad79f2ad2635e6d249e380680", size = 382754, upload-
time = "2026-06-30T07:15:21.831Z" },
    { url = "https://files.pythonhosted.org/packages/04/8f/d2f3f532616be4d06c316ef119683e832bd3d41e112bf3a88f4151c95b17/rpds_py-2026.6.3-cp312-cp312-m
anylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:ecabd69db66de867690f9797f2f8fa27ba501bbc24540cbdbdc649cd15888ba6", size = 366189, uploa
d-time = "2026-06-30T07:15:23.371Z" },
    { url = "https://files.pythonhosted.org/packages/e3/29/41a7b0e98a4b44cd676ab7598419623373eb43b20be68c084935c1a8cf88/rpds_py-2026.6.3-cp312-cp312-m
anylinux_2_31_riscv64.whl", hash = "sha256:58eadac9cd119677b60e1cf8ac4052f35949d71b8a9e5556efccbe82533cf22a", size = 377750, upload-time = "2026-06-30
T07:15:24.659Z" },
    { url = "https://files.pythonhosted.org/packages/2e/05/ecda0bec46f9a1565090bcdc941d023f6a25aff85fda28f89f8d19878152/rpds_py-2026.6.3-cp312-cp312-m
anylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:7491ee23305ac3eb59e492b6945881f5cd77a6f731061a3f25b77fd40f9e99a4", size = 395576, upload-time =
 "2026-06-30T07:15:25.987Z" },
    { url = "https://files.pythonhosted.org/packages/68/a8/6ed52f03ee6cb854ce78785cc9a9a672eb880e83fd7224d471f667d151f1/rpds_py-2026.6.3-cp312-cp312-m
usllinux_1_2_aarch64.whl", hash = "sha256:2c99f7e8ccb3dd6e3e4bfeac657a7b208c9bac8075f4b078c02d7404c34107fa", size = 543807, upload-time = "2026-06-30T
07:15:27.356Z" },
    { url = "https://files.pythonhosted.org/packages/8f/d6/156c0d3eea27ba09b92562ba2364ba124c0a061b199e17eac637cd25a5e2/rpds_py-2026.6.3-cp312-cp312-m
usllinux_1_2_i686.whl", hash = "sha256:62698275682bf121181861295c9181e789030a2d516071f5b8f3c23c170cd0fc", size = 611187, upload-time = "2026-06-30T07:
15:28.931Z" },
    { url = "https://files.pythonhosted.org/packages/f1/31/774212ed989c62f7f310220089f9b0a3fb8f40f5443d1727abd5d9f52bc9/rpds_py-2026.6.3-cp312-cp312-m
usllinux_1_2_x86_64.whl", hash = "sha256:a214c993455f99a89aaeadc9b21241900037adc9d97203e374d75513c5911822", size = 573030, upload-time = "2026-06-30T0
7:15:30.553Z" },
    { url = "https://files.pythonhosted.org/packages/c9/50/22f73127a41f1ce4f87fe39aadfb9a126345801c274aa93ae88456249327/rpds_py-2026.6.3-cp312-cp312-w
in32.whl", hash = "sha256:501f9f04a588d6a09179368c5707130144519176 7c64e4b52a6aa9871f1ef5ed", size = 202185, upload-time = "2026-06-30T07:15:32.027Z" }
,
    { url = "https://files.pythonhosted.org/packages/04/3a/f0ee4d4dde9d3b69dedf1b5f74e7a40017046d55052d173e418c6a94f960/rpds_py-2026.6.3-cp312-cp312-w
in_amd64.whl", hash = "sha256:2c958bf94822e9290a40aaf2a822d4bc5c88099093e3948ad6c571eca9272e5f", size = 220394, upload-time = "2026-06-30T07:15:33.359
Z" },
    { url = "https://files.pythonhosted.org/packages/f3/83/3382fe37f809b59f02aac04dbc4e765b480b46ee0227ed516e3bdc4d3dfc/rpds_py-2026.6.3-cp312-cp312-w
in_arm64.whl", hash = "sha256:22bffe6042b9bcb0822bcd1955ec00e245daf17b4344e4ed8e9551b976b63e96", size = 215753, upload-time = "2026-06-30T07:15:34.778
Z" },
    { url = "https://files.pythonhosted.org/packages/a4/9e/b818ee580026ec578138e961027a68820c40afeb1ec8f6819b54fb99e196/rpds_py-2026.6.3-cp313-cp313-m
acosx_10_12_x86_64.whl", hash = "sha256:3cfe765c1da0072636ca06628261e0ea05688e160d5c8a03e0217c3854037223", size = 343012, upload-time = "2026-06-30T07
:15:36.005Z" },
    { url = "https://files.pythonhosted.org/packages/f3/6b/686d9dc4359a8f163cfbbf89ee0b4e586431de22fe8248edb63a8cf50d49/rpds_py-2026.6.3-cp313-cp313-m
acosx_11_0_arm64.whl", hash = "sha256:f4d78253f6996be4901669ad25319f842f740eccf4d58e3c7f3dd39e6dde1d8f", size = 338203, upload-time = "2026-06-30T07:1
5:37.462Z" },
    { url = "https://files.pythonhosted.org/packages/9e/9b/069aa329940f8207615e091f5eedbbd40e1e15eac68a0790fd05ccdf796c/rpds_py-2026.6.3-cp313-cp313-m
anylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:54f45a148e28767bf343d33a684693c70e451c6f4c0e9904709a723fafbdfc1f", size = 367984, upl
oad-time = "2026-06-30T07:15:39.008Z" },
    { url = "https://files.pythonhosted.org/packages/14/db/34c203e4becff3703e4d3bc121842c00b8689197f398161203a880052f4e/rpds_py-2026.6.3-cp313-cp313-m
anylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:842e7b070435622248c7a2c44ae53fa1440e073cc3023bc919fed570884097a7", size = 374815, uploa
d-time = "2026-06-30T07:15:40.253Z" },
    { url = "https://files.pythonhosted.org/packages/ee/7d/8071067d2cc453d916ad836e828c943f575e8a44612537759002a1e07381/rpds_py-2026.6.3-cp313-cp313-m
anylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:8020133a74bd81b4572dd8e4be028a6b1ebcd70e6726edc3918008c08bee6ee6", size = 490545, upl
oad-time = "2026-06-30T07:15:41.729Z" },
    { url = "https://files.pythonhosted.org/packages/a3/42/da06c5aa8f0484ff07f270787434204d9f4535e2f8c3b51ed402267e63c3/rpds_py-2026.6.3-cp313-cp313-m
anylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:cdc7e35386f3847df728fbcb5e887e2d79c19e2fa1eba9e51b6621d23e3243af", size = 382828, upload-
time = "2026-06-30T07:15:43.327Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/57/d7/fe978efc2ae50abe48eb7464668ea99f53c010c60aeebb7b35ad27f23661/rpds_py-2026.6.3-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:acac386b453c2516111b50985d60ce46e7fadb5ea71ae7b25f4c946935bf27cf", size = 365678, uploaded-time = "2026-06-30T07:15:44.992Z" },
    { url = "https://files.pythonhosted.org/packages/69/9d/1d8922e1990b2a6eb532b6ff53d3e73d2b3bbffc84116c75826bee73dfc6/rpds_py-2026.6.3-cp313-cp313-manylinux_2_31_riscv64.whl", hash = "sha256:425560c6fa0415f27261727bb20bd097568485e5eb0c121f1949417d1c516885", size = 377811, upload-time = "2026-06-30T07:15:46.523Z" },
    { url = "https://files.pythonhosted.org/packages/b1/3d/198dceafb4fb034a6a47347e1b0735d34e0bd4a50be4e898d408ee66cb14/rpds_py-2026.6.3-cp313-cp313-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:a550fb4950a06dde3beb4721f5ad4b25bf4513784665b0a8522c792e2bd822a4", size = 395382, upload-time = "2026-06-30T07:15:47.955Z" },
    { url = "https://files.pythonhosted.org/packages/1f/f1/13968e49655d40b6b19d8b9140296bbc6f1d86b3f0f6c346cf9f1adddf4b/rpds_py-2026.6.3-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:4f4bca01b63096f606e095734dd56e74e175f94cfbf24ff3d63281cec61f7bb7", size = 543832, upload-time = "2026-06-30T07:15:49.33Z" },
    { url = "https://files.pythonhosted.org/packages/ac/ab/289bcb1b90bd3e40a2900c561fa0e2087345ecbb094f0b870f2345142b7c/rpds_py-2026.6.3-cp313-cp313-musllinux_1_2_i686.whl", hash = "sha256:ccffae9a092a00deb7efd545fe5e2c33c33b88e7c054337e9a74c179347d0b7d", size = 611011, upload-time = "2026-06-30T07:15:50.847Z" },
    { url = "https://files.pythonhosted.org/packages/1e/16/5043105e679436ccfbc8e5e0dd2d663ed18a8b8113515fd06a5e5d77c83e/rpds_py-2026.6.3-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:1cf01971c4f2c5553b772a542e4aaf191789cd331bc2cd4ff0e6e65ba49e1e97", size = 572431, upload-time = "2026-06-30T07:15:52.394Z" },
    { url = "https://files.pythonhosted.org/packages/85/ed/adab103321c0a6565d5ae1c2998349bc3ee175b82ccc5ae8fc04cc413075/rpds_py-2026.6.3-cp313-cp313-win32.whl", hash = "sha256:8c3d1e9c15b9d51ca0391e13da1a25a0a4df3c58a37c9dc368e0736cf7f69df0", size = 201710, upload-time = "2026-06-30T07:15:53.894Z" },
    { url = "https://files.pythonhosted.org/packages/7b/ed/a03b09668e74e5dabbf2e211f6468e1820c0552f7b0500082da31841bf7b/rpds_py-2026.6.3-cp313-cp313-win_amd64.whl", hash = "sha256:9250a9a0a6fd4648b3f868da8d91a4c52b5811a62df58e753d50ae4454a36f80", size = 219454, upload-time = "2026-06-30T07:15:55.25Z" },
    { url = "https://files.pythonhosted.org/packages/27/17/b8642c12930b71bc2b25831f6708ccf0f75abcd11883932ec9ce54ba3a78/rpds_py-2026.6.3-cp313-cp313-win_arm64.whl", hash = "sha256:900a67df3fd1660b035a4761c4ce73c382ea6b35f90f9863c36c6fd8bf8b09bb", size = 215063, upload-time = "2026-06-30T07:15:56.573Z" },
    { url = "https://files.pythonhosted.org/packages/b6/36/7fbe9dcdaf857fb3f63c2a2284b62492d95f5e8334e947e5fb6e7f68c9be/rpds_py-2026.6.3-cp314-cp314-macosx_10_12_x86_64.whl", hash = "sha256:931908d9fc855d8f74783377822be318edb6dcb19e47169dc038f9a1bf60b06e", size = 344510, upload-time = "2026-06-30T07:15:57.921Z" },
    { url = "https://files.pythonhosted.org/packages/ba/54/f785cc3d3f60839ca57a5af4927a9f347b07b2799c373fc20f7949f87c7e/rpds_py-2026.6.3-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:d7469697dce35be237db177d42e2a2ee26e6dcc5fc052078a6fefabd288c6edd", size = 339495, upload-time = "2026-06-30T07:15:59.238Z" },
    { url = "https://files.pythonhosted.org/packages/63/ef/d4cdaf309e6b095b43597103cf8c0b951d6cca2acce68c474f75ec12e0c7/rpds_py-2026.6.3-cp314-cp314-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:bcfbcf66006befb9fd2aeaa9e01feaf881b4dc330a02ba07d2322b1c11be7b5d", size = 369454, upload-time = "2026-06-30T07:16:01.021Z" },
    { url = "https://files.pythonhosted.org/packages/96/4a/9559a68b7ee15db09d7981212e8c2e219d2a1d6d4faa0391d813c3496a36/rpds_py-2026.6.3-cp314-cp314-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:847927daf4cffbd4e90e42bc890069897101edd015f956cb8721b3473372edda", size = 374583, uploaded-time = "2026-06-30T07:16:02.287Z" },
    { url = "https://files.pythonhosted.org/packages/ef/75/8964aa7d2c6e8ac43eba8eb6e6b0fdda1f46d39f2fc3e6aa9f2cb17f485d/rpds_py-2026.6.3-cp314-cp314-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:aca6c1ef08a82bfe327cc156da694660f599923e2e6665b6d81c9c2d0ac9ffc8", size = 492919, upload-time = "2026-06-30T07:16:03.723Z" },
    { url = "https://files.pythonhosted.org/packages/8f/97/6908094ac804115e65aedfd90f1b5fee4eebebd3f6c4cfc5419939267565/rpds_py-2026.6.3-cp314-cp314-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:ae50181a047c871561212bb97f7932a2d45fb53e947bd9b57ebad85b529cbc53", size = 383725, upload-time = "2026-06-30T07:16:05.305Z" },
    { url = "https://files.pythonhosted.org/packages/d1/9c/0d1fdc2e7aba23e290d603bc494e97bd205bae262ce33c6b32a69768ed5e/rpds_py-2026.6.3-cp314-cp314-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:dc319e5a1de4b6913aac94bf6a2f9e847371e0a140a43dd4991db1a09bc2d504", size = 367255, uploaded-time = "2026-06-30T07:16:07.086Z" },
    { url = "https://files.pythonhosted.org/packages/c4/fe/f0209ca4a9ed074bc8acb44dfd0e81c3122e94c9689f5645b7973a866719/rpds_py-2026.6.3-cp314-cp314-manylinux_2_31_riscv64.whl", hash = "sha256:e4316bf32babbed84e691e352faf967ce2f0f024174a8643c37c94a1080374fc", size = 379060, upload-time = "2026-06-30T07:16:08.525Z" },
    { url = "https://files.pythonhosted.org/packages/c6/8d/f1cc54c616b9d8897de8738aac148d20afca93f68187475fe194d09a71b9/rpds_py-2026.6.3-cp314-cp314-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:8c6e5a2f750cc71c3e3b11d71661f21d6f9bc6cebc6564b1466417a1ec03ec77", size = 395960, upload-time = "2026-06-30T07:16:09.989Z" },
    { url = "https://files.pythonhosted.org/packages/fb/04/aafff00f73aeca2945f734f1d483c64ab8f472d0864ab02377fd8e89c3b2/rpds_py-2026.6.3-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:4470ce197d4090875cf6affbf1f853338387428df97c4fb7b7106317b8214698", size = 545356, upload-time = "2026-06-30T07:16:11.816Z" },
    { url = "https://files.pythonhosted.org/packages/fd/cc/e229663b9e4ddac5a4acbe9085dd80a71af2a5d356b8b39d6bff233f24b0/rpds_py-2026.6.3-cp314-cp314-musllinux_1_2_i686.whl", hash = "sha256:ea964164cc9afa72d4d9b23cc28dafae93693c0a53e0b42acbff15b22c3f9ddd", size = 612319, upload-time = "2026-06-30T07:16:13.586Z" },
    { url = "https://files.pythonhosted.org/packages/e3/7a/8a0e6d3e6cd066af108b71b43122c3fe158dd9eb86acac626593a2582eb1/rpds_py-2026.6.3-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:639c8929aa0afe81be836b04de888460d6bed38b9c54cfc18da8f6bfabf5af5d", size = 573508, upload-time = "2026-06-30T07:16:15.23Z" },
    { url = "https://files.pythonhosted.org/packages/87/03/2a69ab618a789cf6cf85c86bb844c62d090e700ab1a2aa676b3741b6c516/rpds_py-2026.6.3-cp314-cp314-win32.whl", hash = "sha256:882076c00c0a608b131187055ddc5ae29f2e7eaf870d6168980420d58528a5c8", size = 202504, upload-time = "2026-06-30T07:16:16.893Z" },
    { url = "https://files.pythonhosted.org/packages/85/62/a3892ba945f4e24c78f352e5de3c7620d8479f73f211406a97263d13c7d2/rpds_py-2026.6.3-cp314-cp314-win_amd64.whl", hash = "sha256:0be972be84cfcaf46c8c6edf690ca0f154ac17babf1f6a955a51579b34ad2dc5", size = 220380, upload-time = "2026-06-30T07:16:18.108Z" },
    { url = "https://files.pythonhosted.org/packages/3d/e7/c2bd44dc831931815ad11ebb5f430b5a0a4d3caa9de837107876c30c3432/rpds_py-2026.6.3-cp314-cp314-win_arm64.whl", hash = "sha256:2a9c6f195058cb45335e8cc3802745c603d716eb96bc9625950c1aac71c0c703", size = 215976, upload-time = "2026-06-30T07:16:19.654Z" },
    { url = "https://files.pythonhosted.org/packages/79/9c/fff7b74bce9a091ec9a012a03f9ff5f69364eaf9451060dfc4486da2ffdd/rpds_py-2026.6.3-cp314-cp314t-macosx_10_12_x86_64.whl", hash = "sha256:f90938e92afda60266da758ee7d363447f7f0138c9559f9e1811629580582d90", size = 346840, upload-time = "2026-06-30T07:16:21.268Z" },
    { url = "https://files.pythonhosted.org/packages/e9/44/77bcb1168b33704908295533d27f10eb811e9e3e193e8993dc99572211d3/rpds_py-2026.6.3-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:ec829541c45bca16e61c7ae50c20501f213605beb75d1aba91a6ee37fbbb56a4", size = 340282, upload-time = "2026-06-30T07:16:22.875Z" },
    { url = "https://files.pythonhosted.org/packages/87/3c/7a9081c7c9e645b39efe19e4ffbeccd80add246327cd9b888aecffd72317/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:afd70d95892096cdb26f15a00c45907b17817577aa8d1c76b2dcc2788391f9e9", size = 370403, upload-time = "2026-06-30T07:16:24.415Z" },
    { url = "https://files.pythonhosted.org/packages/f7/69/af47021eb7dad6ff3396cb001c08f0f3c4d06c20253f75be6421a59fe6b7/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:29dfa0533a5d4c94d4dfa1b694fcb56c9c63aad8330ffdd816fd225d0a7a162f", size = 376055, upload-time = "2026-06-30T07:16:26.111Z" },
    { url = "https://files.pythonhosted.org/packages/81/fc/a3bcf517084396a6dd258c592567a3c011ba4557f2fde23dceaf26e74f2e/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:af05d726809bff6b141be124d4c7ce998f9c9c7f30edb1f46c07aa103d540b41", size = 494419, upload-time = "2026-06-30T07:16:27.596Z" },
    { url = "https://files.pythonhosted.org/packages/c9/eb/13d529d1788135425c7bf207f8463458ca5d92e43f3f701365b83e9dffc1/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:9826217f048f620d9a712672818bf231442c1b35d96b227a07eabd11b4bb6945", size = 384848, upload-time = "2026-06-30T07:16:29.183Z" },
    { url = "https://files.pythonhosted.org/packages/8e/f4/b7ac49f30013aba8f7b9566b1dd07e81de95e708c1374b7bacc5b9bc5c9c/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:536bceea4fa4acf7e1c61da2b5786304367c816c8895be71b8f537c480b0ea1f", size = 371369, upload-time = "2026-06-30T07:16:30.912Z" },
    { url = "https://files.pythonhosted.org/packages/31/86/6260bafa622f788b07ddec0e52d810305c8b9b0b8c27f58a2ab04bf62b4f/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_31_riscv64.whl", hash = "sha256:bc0011654b91cc4fb2ae701bec0a0ba1e552c0714247fa7af6c59e0ccfa3a4e1", size = 379673, upload-time = "2026-06-30T07:16:32.486Z" },
    { url = "https://files.pythonhosted.org/packages/19/c3/03f1ee79a047b48daeca157c89a18509cde22b6b951d642b9b0af1be660a/rpds_py-2026.6.3-cp314-cp314t-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:539d75de9e0d536c84ff18dfeb805398e58227001ce09231a26a08b9aed1ee0e", size = 397500, upload-time = "2026-06-30T07:16:34.471Z" },
    { url = "https://files.pythonhosted.org/packages/f0/95/8ed0cd8c377dca12aea498f119fe639fc474d1461545c39d2b5872eb1c0f/rpds_py-2026.6.3-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:166cf54d9f44fc6ceb53c7860258dde44a81406646de79f8ed3234fca3b6e538", size = 545978, upload-time = "2026-06-30T07:16:36.45Z" },
    { url = "https://files.pythonhosted.org/packages/d3/f2/0eb57f0eaa83f8fc152a7e03de968ab77e1f00732bebc892b190c6eebde7/rpds_py-2026.6.3-cp314-cp314t-musllinux_1_2_i686.whl", hash = "sha256:d34c20167764fbcf927194d532dd7e0c56772f0a5f943fa5ef9e9afbba8fb9db", size = 613350, upload-time = "2026-06-30T07:16:38.213Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/5b/de/e0674bdbc3ef7634989b3f854c3f34bc1f587d36e5bfdc5c378d57034619/rpds_py-2026.6.3-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:ea7bb13b7c9a29791f87a0387ba7d3ad3a6d783d827e4d3f27b40a0ff44495e2", size = 576486, upload-time = "2026-06-30T07:16:39.797Z" },
    { url = "https://files.pythonhosted.org/packages/f2/f6/21101359743cd136ada781e8210a85769578422ba460672eea0e29739200/rpds_py-2026.6.3-cp314-cp314t-win32.whl", hash = "sha256:6de4744d05bd1aa1be4ed7ea1189e397919680800811 3bbbf899a460966b925e", size = 201068, upload-time = "2026-06-30T07:16:41.316Z" },
    { url = "https://files.pythonhosted.org/packages/a6/b2/9574d4d44f7760c2aa32d92a0a4f41698e33f5b204a0bf5c9758f52c79d5/rpds_py-2026.6.3-cp314-cp314t-win_amd64.whl", hash = "sha256:c7b9a2f8f4d8e90af72571d3d495deebdd7e3c75451f5b41719aee166e940fc2", size = 220600, upload-time = "2026-06-30T07:16:43.091Z" },
    { url = "https://files.pythonhosted.org/packages/08/ae/f23a2697e6ee6340a578b0f136be6483657bef0c6f9497b752bb5c0964bb/rpds_py-2026.6.3-cp315-cp315-macosx_10_12_x86_64.whl", hash = "sha256:e059c5dde6452b44424bd1834557556c226b57781dee1227af23518459722b13", size = 344726, upload-time = "2026-06-30T07:16:44.5Z" },
    { url = "https://files.pythonhosted.org/packages/c3/63/e7b3a1a5358dd32c930a1062d8e15b67fd6e8922e81df9e91706d66ee5c8/rpds_py-2026.6.3-cp315-cp315-macosx_11_0_arm64.whl", hash = "sha256:2f7c26fbc5acd2522b95d4177fe4710ffd8e9b20529e703ffbf8db4d93903f05", size = 339587, upload-time = "2026-06-30T07:16:46.255Z" },
    { url = "https://files.pythonhosted.org/packages/ec/64/10a85681916ca55fffb91b0a211f84e34297c109243484dd6394660a8a7c/rpds_py-2026.6.3-cp315-cp315-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:a3086b538543802f84c843911242db20447de00d8752dd0efc936dbcf02218ba", size = 369585, upload-time = "2026-06-30T07:16:48.101Z" },
    { url = "https://files.pythonhosted.org/packages/76/c2/baf95c7c38823e12ba34407c5f5767a89e5cf2233895e56f608167ae9493/rpds_py-2026.6.3-cp315-cp315-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:8f2e5c5ee828d42cb11760761c0af6507927bec42d0ad5458f97c9203b054617", size = 375479, uploaded-time = "2026-06-30T07:16:49.93Z" },
    { url = "https://files.pythonhosted.org/packages/6a/94/0aad06c72d65101e11d33528d438cda99a39ce0da99466e156158f2541d3/rpds_py-2026.6.3-cp315-cp315-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:ed0c1e5d10cdc7135537988c74a0188da68e2f3c30813ba3744ab1e42e0480f9", size = 492418, upload-time = "2026-06-30T07:16:51.641Z" },
    { url = "https://files.pythonhosted.org/packages/b5/17/de3f5a479a1f056535d7489819639d8cd591ea6281d700390b43b1abd745/rpds_py-2026.6.3-cp315-cp315-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:8c2642a7603ec0b16ed77da4555db3b4b4723419048737883 27c0b0d7b95f1bb", size = 384123, upload-time = "2026-06-30T07:16:53.622Z" },
    { url = "https://files.pythonhosted.org/packages/46/7d/bf09bd1b145bb2671c03e1e6d1ab8651858d90d8c7dfeadd85a37a934fd8/rpds_py-2026.6.3-cp315-cp315-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:8e4320744c1ffdd95a603def63344bfab2d33edeab301c5007e7de9f9f5b3885", size = 367351, uploaded-time = "2026-06-30T07:16:55.241Z" },
    { url = "https://files.pythonhosted.org/packages/a3/ea/1bb734f314b8be319149ddee80b18bd41372bdcfbdf88d28131c0cd37719/rpds_py-2026.6.3-cp315-cp315-manylinux_2_31_riscv64.whl", hash = "sha256:a9f4645593036b81bbdb36b9c8e0ea0d1c3fee968c4d59db0344c14087ef143a", size = 378827, upload-time = "2026-06-30T07:16:56.841Z" },
    { url = "https://files.pythonhosted.org/packages/4b/93/d9611e5b25e26df9a3649813ed66193ace9347a7c7fc4ab7cf70e94851c0/rpds_py-2026.6.3-cp315-cp315-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:e55d236be29255554da47abe5c577637db7c24a02b8b46f0ca9524c855801868", size = 395966, upload-time = "2026-06-30T07:16:58.557Z" },
    { url = "https://files.pythonhosted.org/packages/c3/cb/99d77e16e5534ae1d90629bbe419ba6ee170833a6a85e3aa1cc41726fbbc/rpds_py-2026.6.3-cp315-cp315-musllinux_1_2_aarch64.whl", hash = "sha256:24e9c5386e16669b674a69c156c8eeefcb578f3b3397b713b08e6d60f3c7b187", size = 545680, upload-time = "2026-06-30T07:17:00.164Z" },
    { url = "https://files.pythonhosted.org/packages/59/15/11a29755f790cef7a2f755e8e14f4f0c33f39489e1893a632a2eee59672b/rpds_py-2026.6.3-cp315-cp315-musllinux_1_2_i686.whl", hash = "sha256:c60924535c75f1566b6eb75b5c31a48a43fef04fa2d0d201acbad8a9969c6107", size = 611853, upload-time = "2026-06-30T07:17:01.962Z" },
    { url = "https://files.pythonhosted.org/packages/68/86/0c27547e21644da938fb530f7e1a8148dd24d02db07e7a5f2567a17ce710/rpds_py-2026.6.3-cp315-cp315-musllinux_1_2_x86_64.whl", hash = "sha256:38a2fea2787428f811719ceb9114cb78964a3138838320c29ac39526c79c16ba", size = 573715, upload-time = "2026-06-30T07:17:03.693Z" },
    { url = "https://files.pythonhosted.org/packages/29/71/4d8fcf700931815594bce892255bbd973b94efaf0fc1932b0590df18d886/rpds_py-2026.6.3-cp315-cp315-win32.whl", hash = "sha256:d483fe17f01ad64b7bf7cc38fcefff1ca9fb83f8c2b2542b68f97ffe0611b369", size = 202864, upload-time = "2026-06-30T07:17:05.746Z" },
    { url = "https://files.pythonhosted.org/packages/eb/62/b577562de0edbb55b2be85ce5fd09c33e386b9b13eee09833af4240fd5c4/rpds_py-2026.6.3-cp315-cp315-win_amd64.whl", hash = "sha256:67e3a721ffc5d8d2210d3671872298c4a84e4b8035cfe42ffd7cde35d772b146", size = 220430, upload-time = "2026-06-30T07:17:07.471Z" },
    { url = "https://files.pythonhosted.org/packages/c8/95/d6d0b2509825141eef60669a5739eec88dbc6a48053d6c92993a5704defe/rpds_py-2026.6.3-cp315-cp315-win_arm64.whl", hash = "sha256:6e84adbcf4bf841aed8116a8264b9f50b4cb3e7bd89b516122e616ac56ca269e", size = 215877, upload-time = "2026-06-30T07:17:09.008Z" },
    { url = "https://files.pythonhosted.org/packages/b7/bf/f3ea278f0afd615c1d0f19cb69043a41526e2bb600c2b536eb192218eb27/rpds_py-2026.6.3-cp315-cp315t-macosx_10_12_x86_64.whl", hash = "sha256:ae6dd8f10bd17aad820876d24caec9efdafd80a318d16c0a48edb5e136902c6b", size = 346933, upload-time = "2026-06-30T07:17:10.762Z" },
    { url = "https://files.pythonhosted.org/packages/9d/29/9907bdf1c5346763cf10b7f6852aad86652168c259def904cbe0082c5864/rpds_py-2026.6.3-cp315-cp315t-macosx_11_0_arm64.whl", hash = "sha256:bdbd97738551fca3917c1bd7188bec1920bb520104f28e7e1007f9ceb17b7690", size = 340274, upload-time = "2026-06-30T07:17:12.266Z" },
    { url = "https://files.pythonhosted.org/packages/6f/2c/8e03767b5778ef25cebf74a7a91a2c3806f8eced4c92cb7406bbe060756d/rpds_py-2026.6.3-cp315-cp315t-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:8b95977e7211527ab0ba576e286d023389fbeeb32a6b7b771665d333c60e5342", size = 370763, upload-time = "2026-06-30T07:17:14.107Z" },
    { url = "https://files.pythonhosted.org/packages/2e/e1/df2a7e1ba2efd796af26194250b8d42c821b46592311595162af9ef0528d/rpds_py-2026.6.3-cp315-cp315t-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:d15fde0e6fb0d88a60d221204873743e5d9f0b7d29165e62cd86d0413ad74ba6", size = 376467, upload-time = "2026-06-30T07:17:15.76Z" },
    { url = "https://files.pythonhosted.org/packages/6b/de/8a0814d1946af29cb068fb259aa8622f856df1d0bab58429448726b537f5/rpds_py-2026.6.3-cp315-cp315t-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:a136d453475ac0fcbda502ef1e6504bd28d6d904700915d278deeab0d00fe140", size = 496689, upload-time = "2026-06-30T07:17:17.308Z" },
    { url = "https://files.pythonhosted.org/packages/df/f3/f19e0c852ba13694f5a79f3b719331051573cb5693feacf8a88ffffc3a71/rpds_py-2026.6.3-cp315-cp315t-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:f826877d462181e5eb1c26a0026b8d0cab05d99844ecb6d8bf3627a2ca0c0442", size = 385340, upload-time = "2026-06-30T07:17:18.928Z" },
    { url = "https://files.pythonhosted.org/packages/e2/ae/7ec3a9d2d4351f99e37bcb06b6b6f954512646bfdbf9742e1de727865daf/rpds_py-2026.6.3-cp315-cp315t-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:79486287de1730dbaff3dbd124d0ca4d2ef7f9d29bf2544f1f93c09b5bcbbd12", size = 372179, upload-time = "2026-06-30T07:17:20.539Z" },
    { url = "https://files.pythonhosted.org/packages/d3/ac/9cee911dff2aaa9a5a8354f6610bf2e6a616de9197c5fff4f54f82585f1e/rpds_py-2026.6.3-cp315-cp315t-manylinux_2_31_riscv64.whl", hash = "sha256:808345f53cb952433ca2816f1604ff3515608a81784954f38d4452acfe8e61d5", size = 379993, upload-time = "2026-06-30T07:17:22.212Z" },
    { url = "https://files.pythonhosted.org/packages/83/6b/7c2a07ba88d1e9a936612f7a5d067467ed03d971d5a06f7d309dff044a7e/rpds_py-2026.6.3-cp315-cp315t-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:1967debc37f64f2c4dc90a7f563aec558b471966e12adcac4e1c4240496b6ebf", size = 398909, upload-time = "2026-06-30T07:17:23.66Z" },
    { url = "https://files.pythonhosted.org/packages/97/0b/776ffcb66783637b0031f6d58d6fb55913c8b5abf00aeecd46bf933fb477/rpds_py-2026.6.3-cp315-cp315t-musllinux_1_2_aarch64.whl", hash = "sha256:f0840b5b17057f7fd918b76183a4b5a0635f43e14eb2ce60dce1d4ee4707ea00", size = 546584, upload-time = "2026-06-30T07:17:25.264Z" },
    { url = "https://files.pythonhosted.org/packages/55/33/ba3bc04d7092bd553c9b2b195624992d2cc4f3de1f380b7b93cbee67bd79/rpds_py-2026.6.3-cp315-cp315t-musllinux_1_2_i686.whl", hash = "sha256:faa679d19a6696fd54259ad321251ad77a13e70e03dd834daa762a44fb6196ef", size = 614357, upload-time = "2026-06-30T07:17:26.888Z" },
    { url = "https://files.pythonhosted.org/packages/8b/71/14edf065f04630b1a8472f7653cad03f6c478bcf95ea0e6aed55451e33ea/rpds_py-2026.6.3-cp315-cp315t-musllinux_1_2_x86_64.whl", hash = "sha256:23a439f31ccbeff1574e24889128821d1f7917470e830cf6544dced1c662262a", size = 576533, upload-time = "2026-06-30T07:17:28.546Z" },
    { url = "https://files.pythonhosted.org/packages/ba/76/65002b08596c389105720a8c0d22298b8dc25a4baf89b2ce431343c8b1de/rpds_py-2026.6.3-cp315-cp315t-win32.whl", hash = "sha256:913ca42ccad3f8cc6e292b587ae8ae49c8c823e5dce51a736252fc7c7cdfa577", size = 201204, upload-time = "2026-06-30T07:17:30.193Z" },
    { url = "https://files.pythonhosted.org/packages/8c/97/d855d6b3c322d1f27e26f5241c42016b56cf01377ea8ed348285f54652f0/rpds_py-2026.6.3-cp315-cp315t-win_amd64.whl", hash = "sha256:ae3d4fe8c0b9213624fdce7279d70e3b148b682ca20719ebd193a23ebfa47324", size = 220719, upload-time = "2026-06-30T07:17:31.788Z" },
]

[[package]]
name = "rsa"
version = "4.9.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "pyasn1" },
]
```

```
sdist = { url = "https://files.pythonhosted.org/packages/da/8a/22b7beea3ee0d44b1916c0c1cb0ee3af23b700b6da9f04991899d0c555d4/rsa-4.9.1.tar.gz", hash =
"sha256:e7bdbfdb5497da4c07dfd35530e1a902659db6ff241e39d9953cad06ebd0ae75", size = 29034, upload-time = "2025-04-16T09:51:18.218Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/64/8d/0133e4eb4beed9e425d9a98ed6e081a55d195481b7632472be1af08d2f6b/rsa-4.9.1-py3-none-any.whl", h
ash = "sha256:68635866661c6836b8d39430f97a996acbd61bfa49406748ea243539fe239762", size = 34696, upload-time = "2025-04-16T09:51:17.142Z" },
]

[[package]]
name = "ruff"
version = "0.16.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/70/25/7113f6d5498888c5fb7db34081cba7d5971c4cb1bfb26819966eee68f003/ruff-0.16.1.tar.gz", hash
= "sha256:fedad7c801dabd3fb9741d76aca39246e6ddd9ca446a015875207bf19f1e6bc7", size = 4877500, upload-time = "2026-07-30T19:37:01.379Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/1b/bd/694da69368e0973de65df2ddc73ab18d43c469d5963d9b150911de6bc513/ruff-0.16.1-py3-none-linux_arm
v6l.whl", hash = "sha256:58edb313b88f0c5460a26adf5f39a37a3be789494a15e3e411e35fa78b89f9a0", size = 10839126, upload-time = "2026-07-30T19:36:13.697Z"
},
    { url = "https://files.pythonhosted.org/packages/3f/f0/b626e5d5bd0dd9576263658ef12885e2288afd1029a48e26ffed65ec1ac1/ruff-0.16.1-py3-none-macosx_10
_12_x86_64.whl", hash = "sha256:fde5a99e2f97479af66edd6622c6d5a2a7592c77cf4153d9e4428f5eeb55b60c", size = 11070253, upload-time = "2026-07-30T19:36:17
.14Z" },
    { url = "https://files.pythonhosted.org/packages/83/63/f40acfb6b35b88623e71684942b552c3edd96035f5d98f313815f7b277de/ruff-0.16.1-py3-none-macosx_11
_0_arm64.whl", hash = "sha256:e0d4c20532fca4f7fa609369161d968dd28f65d83dabbd61d8e9c7edbf7001f6", size = 10561425, upload-time = "2026-07-30T19:36:20.0
4Z" },
    { url = "https://files.pythonhosted.org/packages/aa/dd/14ec0e9c2b4d315547dd38765004b4863e354e1b52cb308272215d9f6f6d/ruff-0.16.1-py3-none-manylinux
_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:30affbcedf59ad5703d9c91f82266e02b47739f797e1a7b6e158e5526a6dae38", size = 10948879, upload-ti
me = "2026-07-30T19:36:22.476Z" },
    { url = "https://files.pythonhosted.org/packages/33/e9/9d870cbae575030fdef595f04b4b97573c525b5497cce4f4498cf2f85446/ruff-0.16.1-py3-none-manylinux
_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:24e9c631573cbca9d20f1283f8f479b2afa4a8503504822bd71a293889f16743", size = 10643691, upload-time
 = "2026-07-30T19:36:24.914Z" },
    { url = "https://files.pythonhosted.org/packages/c4/09/12743d544e2173f53ecd27217c65f90d2bc0f8424a66a60339e56bbc0457/ruff-0.16.1-py3-none-manylinux
_2_17_i686.manylinux2014_i686.whl", hash = "sha256:b41bdd48fb420987a9b5212e4957c26ad4abce401fa9ea9d4d85843727945f4f", size = 11435354, upload-time = "
2026-07-30T19:36:28.447Z" },
    { url = "https://files.pythonhosted.org/packages/7f/89/a1652b2daee52083c9554a6333b678a8b01d0400f976827bb87857f9449a/ruff-0.16.1-py3-none-manylinux
_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:b0d1e1393b7648079e13669de1c1f4fde06d4583e84d8fd5c1551e0a77a2aa75", size = 12259033, upload-ti
me = "2026-07-30T19:36:31.326Z" },
    { url = "https://files.pythonhosted.org/packages/16/96/ecdcb8c54ee7b123b487f807eb014e6e019155a0b81dfb669acd52f28ce3/ruff-0.16.1-py3-none-manylinux
_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:07bf434b1c95f4e093be4532068ef4fcf00924eb2ade8796075980902d6fd54a", size = 11667981, upload-time =
 "2026-07-30T19:36:34.394Z" },
    { url = "https://files.pythonhosted.org/packages/cd/90/c52e12e0d862e9572f2a33aa227409143520abe53111e9a6babbac7b4af8/ruff-0.16.1-py3-none-manylinux
_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:39897739f112253ee4fdd2e8aa9a4f9ded99fb2be367d5f31dfa4ded6025584c", size = 11468183, upload-time
 = "2026-07-30T19:36:37.339Z" },
    { url = "https://files.pythonhosted.org/packages/2c/6b/4ffb7ad1d83eb16cf8cbb3c8815d3f11c88460fd162d4b372a2059be1c2a/ruff-0.16.1-py3-none-manylinux
_2_31_riscv64.whl", hash = "sha256:82ae3c0c0d74daf17b968a10b7b3bb3ef297ab7de0c1f749646b25e690ccb150", size = 11470071, upload-time = "2026-07-30T19:36
:39.91Z" },
    { url = "https://files.pythonhosted.org/packages/9c/72/32ae7db4c0b5e32ab611787caa19d1546800676d79f7483b7100a3561bf4/ruff-0.16.1-py3-none-musllinux
_1_2_aarch64.whl", hash = "sha256:4d5f2ed10f8242d83fc08d521301089364e3375375705356f20c0e31606ef3ef", size = 10919503, upload-time = "2026-07-30T19:36:
42.65Z" },
    { url = "https://files.pythonhosted.org/packages/f7/ca/3d901ba6ad6fc38da39c3448fc6c59ac945679293a17c3ceb6d6c1cba13e/ruff-0.16.1-py3-none-musllinux
_1_2_armv7l.whl", hash = "sha256:a4665b309891f83f3e3c25447935f1213e9abbd4b5640af7a1f2def9f8d413c1", size = 10649861, upload-time = "2026-07-30T19:36:4
5.18Z" },
    { url = "https://files.pythonhosted.org/packages/92/79/894ef1ced26552d5f8c9cf6d85b0687840e1128c55aeab7b9c2d54a0d880/ruff-0.16.1-py3-none-musllinux
_1_2_i686.whl", hash = "sha256:26e9ca5c9bc3971f20d3cf18a957f52ffd6a5f6564ff15c4912a144dcac22494", size = 11148137, upload-time = "2026-07-30T19:36:47.
936Z" },
    { url = "https://files.pythonhosted.org/packages/2d/69/3609a09fa1cb46cc28b762363e440a354204e5dff01bd0c8d7437874d6b9/ruff-0.16.1-py3-none-musllinux
_1_2_x86_64.whl", hash = "sha256:67e1e1e3fa4f0c82f0e36d4cd61e661f6e7a6196cb1aa92fe0828fa7b8f257cd", size = 11559211, upload-time = "2026-07-30T19:36:5
0.448Z" },
    { url = "https://files.pythonhosted.org/packages/fc/8a/fb22af2fd78a736e241fabf67e30ce1799a64244026377a49e133af90762/ruff-0.16.1-py3-none-win32.whl
", hash = "sha256:d31765e131295b8445caf301e3e8a85b34d1b9b211b4109b7ba457888b051806", size = 10838258, upload-time = "2026-07-30T19:36:53.298Z" },
    { url = "https://files.pythonhosted.org/packages/d4/35/e57fd9fb5d423961df087a00b12d42c0a830288dc2f3b45ecca299158b4f/ruff-0.16.1-py3-none-win_amd64
.whl", hash = "sha256:09b05e8b90c2cb06ad63464350e7a45e8e44a2dfe52072ebfba6666ca8d3f596", size = 11961111, upload-time = "2026-07-30T19:36:56.107Z" },
    { url = "https://files.pythonhosted.org/packages/cb/46/240ea004bf6dc4feb40e9832f2205a476a47dd5b8a3f8211a5fc5f95e20e/ruff-0.16.1-py3-none-win_arm64
.whl", hash = "sha256:dbaadaac38c70239f056d306b7476f246b0bf000fa6b3876402acbf5b227eaf8", size = 11309414, upload-time = "2026-07-30T19:36:58.79Z" },
]

[[package]]
name = "setuptools"
version = "83.0.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/34/26/f5d29e25ffdb535afef2d35cdb55b325298f96debd670da4c325e08d70f4/setuptools-83.0.0.tar.gz",
 hash = "sha256:025bccbbf0fa05b6192bc64ae1e7b16e001fd6d6d4d5de03c97b1c1ade523bef", size = 1154254, upload-time = "2026-07-04T15:31:22.699Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/5d/40/e1e72872c6354b306daef1703549e8e83b4d43cfea356311bf722a043752/setuptools-83.0.0-py3-none-any
.whl", hash = "sha256:29b23c360f22f414dc7336bb39178cc7bcbf6021ed2733cde173f09dba19abb3", size = 1008090, upload-time = "2026-07-04T15:31:20.885Z" },
]

[[package]]
name = "six"
version = "1.17.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/94/e7/b2c673351809dca68a0e064b6af791aa332cf192da575fd474ed7d6f16a2/six-1.17.0.tar.gz", hash =
 "sha256:ff70335d468e7eb6ec65b95b99d3a2836546063f63acc5171de367e834932a81", size = 34031, upload-time = "2024-12-04T17:35:28.174Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/b7/ce/149a00dd41f10bc29e5921b496af8b574d8413afcd5e30dfa0ed46c2cc5e/six-1.17.0-py2.py3-none-any.wh
l", hash = "sha256:4721f391ed90541fddacab5acf947aa0d3dc7d27b2e1e8eda2be8970586c3274", size = 11050, upload-time = "2024-12-04T17:35:26.475Z" },
]

[[package]]
name = "sniffio"
version = "1.3.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/a2/87/a6771e1546d97e7e041b6ae58d80074f81b7d5121207425c964ddf5cfdbd/sniffio-1.3.1.tar.gz", has
h = "sha256:f4324edc670a0f49750a81b895f35c3adb843cca46f0530f79fc1babb23789dc", size = 20372, upload-time = "2024-02-25T23:20:04.057Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e9/44/75a9c9421471a6c4805dbf2356f7c181a29c1879239abab1ea2cc8f38b40/sniffio-1.3.1-py3-none-any.whl
", hash = "sha256:2f6da418d1f1e0fddd844478f41680e794e6051915791a034ff65e5f100525a2", size = 10235, upload-time = "2024-02-25T23:20:01.196Z" },
]

[[package]]
name = "sqlalchemy"
version = "2.0.51"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "greenlet", marker = "platform_machine == 'AMD64' or platform_machine == 'WIN32' or platform_machine == 'aarch64' or platform_machine ==
'amd64' or platform_machine == 'ppc64le' or platform_machine == 'win32' or platform_machine == 'x86_64'" },
    { name = "typing-extensions" },
```

```
]
sdist = { url = "https://files.pythonhosted.org/packages/02/f1/a7a892f18d4d224e6b26f706531eafccc41e37594d37d304786969ee13cb/sqlalchemy-2.0.51.tar.gz",
 hash = "sha256:804dccd8a4a6242c4e30ad961e540e18a588f6527202f2d6791b01845d59fdc9", size = 9912201, upload-time = "2026-06-15T15:41:20.012Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d5/70/e868bc5412acd101a8280f25c95f10eeae0771c4eb806b02491142810ee8/sqlalchemy-2.0.51-cp312-cp312-
macosx_11_0_arm64.whl", hash = "sha256:7d78702b26ba1c18b2d0fb2ea940ba7f17a9581b42e8361ff93920ebbee1235a", size = 2160291, upload-time = "2026-06-15T16
:08:48.918Z" },
    { url = "https://files.pythonhosted.org/packages/e5/1c/71ee0f8a6b9d7316a1ccd30430b4c62b6c2e36adc96017a4e3a72dce49d6/sqlalchemy-2.0.51-cp312-cp312-
manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:581921d849d6e6f994d560389192955e80e2950e18fcdfe2ccea863e01158
e6e", size = 3343835, upload-time = "2026-06-15T16:19:42.613Z" },
    { url = "https://files.pythonhosted.org/packages/2b/7c/7ab9f9aadc5944fdd06612484ed7918fe376ad871a5f50404dc1536e0194/sqlalchemy-2.0.51-cp312-cp312-
manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:1d21ce524ab86c23046e992a5b81cb54c21079c6df6e78b8fc77d77cac70a6b9
", size = 3358470, upload-time = "2026-06-15T16:26:38.011Z" },
    { url = "https://files.pythonhosted.org/packages/d0/7d/ff77169fee6186de145a7f2b87006c39638391130abbab2b1f63ac6ea583/sqlalchemy-2.0.51-cp312-cp312-
musllinux_1_2_aarch64.whl", hash = "sha256:c5d98a2709840027f5a347c3af0a7c3d5f6c1ff93af2ca1c54494e23cba8f389", size = 3289874, upload-time = "2026-06-1
5T16:19:45.212Z" },
    { url = "https://files.pythonhosted.org/packages/6f/3b/6c505903710d781b55bc3141ee34a062bf9745a6b5bc7333305b9ed63b33/sqlalchemy-2.0.51-cp312-cp312-
musllinux_1_2_x86_64.whl", hash = "sha256:1181256e0f16479691b5616d36375dc2620ad8332b25978763c3d206ad3f3f1d", size = 3321692, upload-time = "2026-06-15
T16:26:39.747Z" },
    { url = "https://files.pythonhosted.org/packages/3c/b7/c5ffe50aa2f4d947c9250e1519d939260329a07fe6272edfccd784b3d007/sqlalchemy-2.0.51-cp312-cp312-
win32.whl", hash = "sha256:9f380393be5abeb6815f68fd39271b95127173511b6706b0a630a9995d53f8f5", size = 2119674, upload-time = "2026-06-15T16:23:09.543Z"
 },
    { url = "https://files.pythonhosted.org/packages/25/dc/46a65916af68a06ef6b972c6050ba4c8f97070fe3fb33097d34229d9bef6/sqlalchemy-2.0.51-cp312-cp312-
win_amd64.whl", hash = "sha256:2cf39aabdf48e87c1c2c2ed6d20d33ffa0733b3071ce9c5f66357947dd009080", size = 2146670, upload-time = "2026-06-15T16:23:11.0
48Z" },
    { url = "https://files.pythonhosted.org/packages/54/fe/a210d52fd1a90ecfae8a78e9d8b27e18d733d60818a8bf250ff690b75120/sqlalchemy-2.0.51-cp313-cp313-
macosx_11_0_arm64.whl", hash = "sha256:7c2056838b6685b72fdb36c99996cf862753461a62f2e84f4196371d3b2d6a07", size = 2157184, upload-time = "2026-06-15T16
:08:50.374Z" },
    { url = "https://files.pythonhosted.org/packages/17/6b/2dce8369b199cb855110e056032f94a9f66dacc2237d3d39c115a86eac56/sqlalchemy-2.0.51-cp313-cp313-
manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:483b11bd46bf35fc14c52faf338b04300c9e6ce554bce9b11be85bfec3bc3
195", size = 3284735, upload-time = "2026-06-15T16:19:46.934Z" },
    { url = "https://files.pythonhosted.org/packages/53/ff/dbc495b8a14da840faffb353857a72d4190113cac33727906fb997047f0f/sqlalchemy-2.0.51-cp313-cp313-
manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:1bed1ee8b01da6088210aa9412023326fb98a599ba502e6118308601dcbef77f
", size = 3302756, upload-time = "2026-06-15T16:26:41.336Z" },
    { url = "https://files.pythonhosted.org/packages/cf/d5/fde8f4dddcf518ee15ab35a7c6a28acc32c8ba548d1d2aa451f96e6dbb0b/sqlalchemy-2.0.51-cp313-cp313-
musllinux_1_2_aarch64.whl", hash = "sha256:72ca54c952107ba5cd58854b67a5a6268631289d21651a1235396f3b98b47400", size = 3232055, upload-time = "2026-06-1
5T16:19:49.286Z" },
    { url = "https://files.pythonhosted.org/packages/67/d1/43d3a0ac955a58601c24fa23038b1c55ee3a1ec02c0f96ebb1eae2bcf614/sqlalchemy-2.0.51-cp313-cp313-
musllinux_1_2_x86_64.whl", hash = "sha256:b3e693d15533a45cd5906f0589f9c35090bef6ef45bf1e8195c424aa0ae06a8d", size = 3269850, upload-time = "2026-06-15
T16:26:43.017Z" },
    { url = "https://files.pythonhosted.org/packages/94/df/de669c7054cd47c4439ac34b1b2ee8b804a794791fbb10720e997a2c87c7/sqlalchemy-2.0.51-cp313-cp313-
win32.whl", hash = "sha256:b93ab07b5292dbe7e6b8da89475275e7042744283921344b56105f3eeb0f828b", size = 2117721, upload-time = "2026-06-15T16:23:12.36Z"
},
    { url = "https://files.pythonhosted.org/packages/d0/8a/403c51d064196bae20a0bc2476577f83a3f8dd299719a97417086b7f2ec5/sqlalchemy-2.0.51-cp313-cp313-
win_amd64.whl", hash = "sha256:0f053118c30e53161857a953e4de667d90e274980dccbe5dd3829bbbeece72a5", size = 2143615, upload-time = "2026-06-15T16:23:13.9
06Z" },
    { url = "https://files.pythonhosted.org/packages/b1/49/a739be2e1d02a96a658eb71ab45d921c874249252358ad24a5bffdd02525/sqlalchemy-2.0.51-cp314-cp314-
macosx_11_0_arm64.whl", hash = "sha256:6ea306caaae6bd5afd0a46050003c88f6bf33227377a49298c498c3cb88ff491", size = 2158999, upload-time = "2026-06-15T16
:08:51.759Z" },
    { url = "https://files.pythonhosted.org/packages/23/6b/2e0e38cf75c8780eca78d9b2e78164f8bcfd70125e5caa588ff5cbb9c9f4/sqlalchemy-2.0.51-cp314-cp314-
manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:c45a496d6bc05dec41dcd4c3a2b183723f47473255c159cd80b503c8f2464
24d", size = 3282539, upload-time = "2026-06-15T16:19:51.065Z" },
    { url = "https://files.pythonhosted.org/packages/dd/a1/e77854cb5336fd37dc3c6ae3b71de242c98caac5725120be0b526b31cbd0/sqlalchemy-2.0.51-cp314-cp314-
manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:4004ada0aafe8ae1991b2cd1d99c6d9146126e123bd6f883c260d974aa012e54
", size = 3287545, upload-time = "2026-06-15T16:26:44.735Z" },
    { url = "https://files.pythonhosted.org/packages/f6/ab/9e17272fd4dac8df3b83c4fbe52b998a1c9d89a843c8c35ff29b74ff7364/sqlalchemy-2.0.51-cp314-cp314-
musllinux_1_2_aarch64.whl", hash = "sha256:0f6bcad487aee1c638d707235682fc96f741de00663619881ab235400d03289e", size = 3230929, upload-time = "2026-06-1
5T16:19:52.625Z" },
    { url = "https://files.pythonhosted.org/packages/02/3c/52f408ea701781caee975606beccc48845f2aee8711ac29843d612c0306c/sqlalchemy-2.0.51-cp314-cp314-
musllinux_1_2_x86_64.whl", hash = "sha256:39a76529db6305693d8d4affa58ad5b5e2e18edd62daea628b29b97930b3513d", size = 3252888, upload-time = "2026-06-15
T16:26:46.454Z" },
    { url = "https://files.pythonhosted.org/packages/24/16/3efd2ee6bc4ca4693a30a1dd17a91b606cae15d517d2a4746611d9b73ce8/sqlalchemy-2.0.51-cp314-cp314-
win32.whl", hash = "sha256:08a204d8b5638717c26a24df18fcf40af45a6b22e35b70b1d62f0113c2e278e8", size = 2120551, upload-time = "2026-06-15T16:23:15.629Z"
 },
    { url = "https://files.pythonhosted.org/packages/7b/78/55b12e70f45bccc40d9e483925c065027b3b98ea4cbbdf6f8c2546feaf6c/sqlalchemy-2.0.51-cp314-cp314-
win_amd64.whl", hash = "sha256:96747bfbadb055466e5b46d572618170046b45ce5a4879167f50d70a5319a499", size = 2146318, upload-time = "2026-06-15T16:23:17.1
08Z" },
    { url = "https://files.pythonhosted.org/packages/21/db/a9574ed40fed418924b1b1a3e54f47ee3963053b3d3d325a0d36b41f2c08/sqlalchemy-2.0.51-cp314-cp314t
-macosx_11_0_arm64.whl", hash = "sha256:e5ea1a213be1fcd5e49d9904c3b9939211ded90bc2a64e93f4c01963474285de", size = 2178920, upload-time = "2026-06-15T1
5:59:56.285Z" },
    { url = "https://files.pythonhosted.org/packages/bf/90/a1bb5c7cbba76b7bc1fbd586d0a5479a7bc9c27b4a8298f22ec9423b2bb3/sqlalchemy-2.0.51-cp314-cp314t
-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:7c6b36ed71f41942bdcd2ad2522be46bfce09d5705be5640ecf19bbc7660
e4b7", size = 3566534, upload-time = "2026-06-15T15:58:35.024Z" },
    { url = "https://files.pythonhosted.org/packages/15/4b/481f1fed30e0e9e8dd24aecbb49f29eb57fe7657ece5cf06ee9b84bb97d8/sqlalchemy-2.0.51-cp314-cp314t
-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:0c2c62877097e1a0db401fba5cb4debee33265e5b2a55c4ccb489c02c53b4f7
2", size = 3535844, upload-time = "2026-06-15T16:02:43.973Z" },
    { url = "https://files.pythonhosted.org/packages/02/71/0aa64aeda645510af0a43f7d9ee70932f0d1dc4263aed34c50ee891d9df3/sqlalchemy-2.0.51-cp314-cp314t
-musllinux_1_2_aarch64.whl", hash = "sha256:0378d055e9e8cd6ce4d8dff683bdd3d7d413533c4ee51d67a2b1e0f9eacc0f23", size = 3475355, upload-time = "2026-06-
15T15:58:36.592Z" },
    { url = "https://files.pythonhosted.org/packages/05/db/6061db32316446135a3abae5f308d144ab988a34234726042da3e58b1c63/sqlalchemy-2.0.51-cp314-cp314t
-musllinux_1_2_x86_64.whl", hash = "sha256:6e46fc36029eff666391e0531e5387b62ce6c4f1d8e50b3fb3099eaca1b42522", size = 3486591, upload-time = "2026-06-1
5T16:02:45.346Z" },
    { url = "https://files.pythonhosted.org/packages/0d/c9/f14fdf71bb8957e0c7e39db69bbdf12b5c80f4ef775fdfa127bf4e0d6760/sqlalchemy-2.0.51-cp314-cp314t
-win32.whl", hash = "sha256:9161cfc9efce70d1715f47d6ff40f79c6778c00d53be4fbc09d70301e4b83ba7", size = 2151313, upload-time = "2026-06-15T16:03:39.127Z
" },
    { url = "https://files.pythonhosted.org/packages/6a/c6/673e618e6f4f297e126d9b56ea2f6478708f6c1af4e3223835c22e2c3697/sqlalchemy-2.0.51-cp314-cp314t
-win_amd64.whl", hash = "sha256:159bb6ba32059f57ad7375a8f50d844dd2f19d14954ecf820cd33e20debd46b2", size = 2186280, upload-time = "2026-06-15T16:03:40.
569Z" },
    { url = "https://files.pythonhosted.org/packages/e2/22/dbf013a12ec759e54a34a119e9e217435b3f71b2dd5c61a7ade0a25dae87/sqlalchemy-2.0.51-py3-none-any
.whl", hash = "sha256:bb024d8b621d0be75f4f44ecc7c95045026e76d66dc8f791bb5331d7fed59d5", size = 1944334, upload-time = "2026-06-15T16:09:22.418Z" },
]

[package.optional-dependencies]
asyncio = [
    { name = "greenlet" },
]

[[package]]
name = "sse-starlette"
version = "3.4.8"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "starlette" },
]
```

```
sdist = { url = "https://files.pythonhosted.org/packages/f8/00/b42a44342a054d58cb1115d7c8aa9cb4290dd9442f9c1b91a4b8173dba22/sse_starlette-3.4.8.tar.gz
", hash = "sha256:ed89ffbb75cbf78a5fe2f2109cd584792ee7f9dfac96f791db546df8f15f3f9c", size = 32548, upload-time = "2026-08-05T11:19:49.982Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/dd/3a/764912c58293d95b6dcdf4cc255f9d10de310580ced547b082eb9d72018c/sse_starlette-3.4.8-py3-none-a
ny.whl", hash = "sha256:6e82314c786709a3cd9520f2285cf9fff90e181e598e8a357b0cf80f66afba0d", size = 16516, upload-time = "2026-08-05T11:19:48.748Z" },
]

[[package]]
name = "starlette"
version = "1.4.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
    { name = "typing-extensions", marker = "python_full_version < '3.13'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/0f/3c/76d2fd1f1357ed0f0108d8a5aa233dcf16e2946a8559c84912fe08e01ac7/starlette-1.4.1.tar.gz", h
ash = "sha256:b7332de6e9375593a29ba9eee1e6ecfeb3eb2043e2e19a13b4b71da73ff35540", size = 2709041, upload-time = "2026-08-05T15:17:26.23Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/75/0a/67e95f21498de41433babf7b1db0eeab449eb58872dfb831b27747a70fd0/starlette-1.4.1-py3-none-any.w
hl", hash = "sha256:7d078e0fbefae0d2cecfb80a799d6fb84b1c0c6acd4f14ac79d17d0e7ec27f19", size = 74019, upload-time = "2026-08-05T15:17:24.357Z" },
]

[[package]]
name = "tenacity"
version = "9.1.4"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/47/c6/ee486fd809e357697ee8a44d3d69222b344920433d3b6666ccd9b374630c/tenacity-9.1.4.tar.gz", ha
sh = "sha256:adb31d4c263f2bd041081ab33b498309a57c77f9acf2db65aadf0898179cf93a", size = 49413, upload-time = "2026-02-07T10:45:33.841Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/d7/c1/eb8f9debc45d3b7918a32ab756658a0904732f75e555402972246b0b8e71/tenacity-9.1.4-py3-none-any.wh
l", hash = "sha256:6095a360c919085f28c6527de529e76a06ad89b23659fa881ae0649b867a9d55", size = 28926, upload-time = "2026-02-07T10:45:32.24Z" },
]

[[package]]
name = "tiktoken"
version = "0.13.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "regex" },
    { name = "requests" },
]
sdist = { url = "https://files.pythonhosted.org/packages/e4/e5/5f3cb2159769d0f4324c0e9e87f9de3c4b1cd45848a96b2eb3566ad5ca77/tiktoken-0.13.0.tar.gz", h
ash = "sha256:c9435714c3a84c2319499de9a300c0e604449dd0799ff246458b3bb6a7f433c1", size = 38986, upload-time = "2026-05-15T04:51:27.153Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/85/8e/144bde4e01df66b34bb865557c7cd754ed08b036217ebd79c9db5e9048a9/tiktoken-0.13.0-cp312-cp312-ma
cosx_10_13_x86_64.whl", hash = "sha256:32ac870a806cfb260a02d0cb70426aef02e038297f8ad50df5040bb5af360791", size = 1034888, upload-time = "2026-05-15T04
:50:31.579Z" },
    { url = "https://files.pythonhosted.org/packages/36/18/d4ac9d20956cdebca04841316660ed584c2fecdc2b81722a28bc7ad3b1e4/tiktoken-0.13.0-cp312-cp312-ma
cosx_11_0_arm64.whl", hash = "sha256:4d9980f11429ed2d737c463bb1fb78cf330caa026adf002f714aced7849a687b", size = 982970, upload-time = "2026-05-15T04:50
:32.961Z" },
    { url = "https://files.pythonhosted.org/packages/74/ed/6bb8d05b9f731f749fee5c6f5ca63e981143c826a5985877330507bd13b7/tiktoken-0.13.0-cp312-cp312-ma
nylinux_2_28_aarch64.whl", hash = "sha256:3f277ebea5edd7b8bf03c6f9431e1d67d517530115572b2dc1d465326e8f88c7", size = 1115741, upload-time = "2026-05-15
T04:50:34.475Z" },
    { url = "https://files.pythonhosted.org/packages/34/de/2ca96b07a82d972b74fe4b46de055b79c904e45c7eab699354a0bfa697dc/tiktoken-0.13.0-cp312-cp312-ma
nylinux_2_28_x86_64.whl", hash = "sha256:a116178fa7e1b4065bff05214360373a65cac22f965be7b3f73d00a0dbfe7649", size = 1136523, upload-time = "2026-05-15T
04:50:35.782Z" },
    { url = "https://files.pythonhosted.org/packages/ee/dc/9dafec002c2d4424378563cf4cf5c7fb93631d2a55013c8b87554ee4012c/tiktoken-0.13.0-cp312-cp312-mu
sllinux_1_2_aarch64.whl", hash = "sha256:2c397ddda233208345b01bd30f2fca79ff730e55731d0108a603f9bc57f6af3b", size = 1181954, upload-time = "2026-05-15T
04:50:36.99Z" },
    { url = "https://files.pythonhosted.org/packages/a1/d0/1f8578c45b2f24759b46f0b50d31878c63c73e6bf0f2227e10ec5c5408dc/tiktoken-0.13.0-cp312-cp312-mu
sllinux_1_2_x86_64.whl", hash = "sha256:95097e4f89b06403976e498abf61a0ee73a7497e73fb599cb211d8197a054d91", size = 1240069, upload-time = "2026-05-15T0
4:50:38.221Z" },
    { url = "https://files.pythonhosted.org/packages/aa/90/28d7f154888610aa9237e541986beb62b479df29d193a5a0617dbb1514d0/tiktoken-0.13.0-cp312-cp312-wi
n_amd64.whl", hash = "sha256:8f2d16e7a7c783ad81f36e457d046d1f1c8af70b22aec8a13238efe531977c41", size = 874748, upload-time = "2026-05-15T04:50:39.587Z
" },
    { url = "https://files.pythonhosted.org/packages/9c/83/b096c859c2a47c11731bf2f5885f4028b809dfe2396582883eed9cae372f/tiktoken-0.13.0-cp313-cp313-ma
cosx_10_13_x86_64.whl", hash = "sha256:5df5d1507bd245f1ccad4a074698240021239e455eb0bb4ced4e3d7181872154", size = 1034228, upload-time = "2026-05-15T04
:50:40.988Z" },
    { url = "https://files.pythonhosted.org/packages/53/61/c68e123b6d753e3fc2751e9b18e732c9d8bf1e1926762e736eee935d931c/tiktoken-0.13.0-cp313-cp313-ma
cosx_11_0_arm64.whl", hash = "sha256:8fe806a50664e83a6ffd56cbd1e4f5dcc6cd32a3e7538f70dc38b1a271384545", size = 982978, upload-time = "2026-05-15T04:50
:42.195Z" },
    { url = "https://files.pythonhosted.org/packages/ef/8b/96cc178cc584e65d363134500f297790b06cd48cdeb1e8fcf7bbe60f4715/tiktoken-0.13.0-cp313-cp313-ma
nylinux_2_28_aarch64.whl", hash = "sha256:125bc05005e747f993a83dc67934249932d6e4209854452cd4c0b1d53fba3ba2", size = 1116355, upload-time = "2026-05-15
T04:50:43.564Z" },
    { url = "https://files.pythonhosted.org/packages/86/f5/bab735d2c72ea55404b295d02d092644eb5f7cc6205e34d35eb9abfb9ab2/tiktoken-0.13.0-cp313-cp313-ma
nylinux_2_28_x86_64.whl", hash = "sha256:5e6358911cab4adee6712da27d65573496a4f68cf8a2b5fca6a4ad10fc5748cf", size = 1135772, upload-time = "2026-05-15T
04:50:44.782Z" },
    { url = "https://files.pythonhosted.org/packages/4e/b9/6de04ebdf904edfaad87788011b3735087a0c9ea671b9027e1e4e965e8c8/tiktoken-0.13.0-cp313-cp313-mu
sllinux_1_2_aarch64.whl", hash = "sha256:975cbd78d085d75d26b59660e262736dcaed1e35f8f142cd6291025c01d25486", size = 1182415, upload-time = "2026-05-15T
04:50:46.422Z" },
    { url = "https://files.pythonhosted.org/packages/0d/9c/470a05f3b1caf038f44880e334d47ab674e0c80d514c66b375d14d5afa10/tiktoken-0.13.0-cp313-cp313-mu
sllinux_1_2_x86_64.whl", hash = "sha256:75ab9bc99fa020a4c283424590ecd7f3afd70c1c281cb3fa3192a6c3af9f9615", size = 1239879, upload-time = "2026-05-15T0
4:50:48.052Z" },
    { url = "https://files.pythonhosted.org/packages/42/a6/c1936d16055436cb32e6c6128d68629622e00f4768562f55653752d34768/tiktoken-0.13.0-cp313-cp313-wi
n_amd64.whl", hash = "sha256:6b1615f0ff71953d19729ceb18865429c185b0a23c5353f1bbca34a394bf60f7", size = 874829, upload-time = "2026-05-15T04:50:49.202Z
" },
    { url = "https://files.pythonhosted.org/packages/d6/07/acb5992c3772b5a36284f742cfb7a5895aa4471d1848ac31464ad50d7fdf/tiktoken-0.13.0-cp313-cp313t-m
acosx_10_13_x86_64.whl", hash = "sha256:6eb4a5bfbc6426938026b1a334e898ac53541360d62d8c689870160cc80abd67", size = 1033600, upload-time = "2026-05-15T0
4:50:50.4Z" },
    { url = "https://files.pythonhosted.org/packages/14/e9/742e9aec30f59b9f161f7ff7cd072e02ea836c9e1c0854a8076dfcd40d5c/tiktoken-0.13.0-cp313-cp313t-m
acosx_11_0_arm64.whl", hash = "sha256:43cee3e5400573b2046fbf092cc7a5bc30164f9e4c95ce20714da929df48737a", size = 982516, upload-time = "2026-05-15T04:5
0:52.03Z" },
    { url = "https://files.pythonhosted.org/packages/72/74/ca1541b053e7648254d2e4b42a253e1bb4359f2c91a0a8d49228c794e1a0/tiktoken-0.13.0-cp313-cp313t-m
anylinux_2_28_aarch64.whl", hash = "sha256:7de52e3f566d19b3b11bd37eea552c6c305ad74081f736882bd44d148ed4c48d", size = 1115518, upload-time = "2026-05-1
5T04:50:53.543Z" },
    { url = "https://files.pythonhosted.org/packages/46/e3/93825eaf5a4a504795b787e5d5dea07fbeb3dabf97aa7b450be8bde59c89/tiktoken-0.13.0-cp313-cp313t-m
anylinux_2_28_x86_64.whl", hash = "sha256:51384448aa508e4df84c0f7c1dc3211c7f7b8096325660ee5fc82f3e11b381ce", size = 1136867, upload-time = "2026-05-15
T04:50:55.191Z" },
    { url = "https://files.pythonhosted.org/packages/8c/46/002b68de6827091d5ae90b048f326e8aad8d953520950e5ce1508879414f/tiktoken-0.13.0-cp313-cp313t-m
usllinux_1_2_aarch64.whl", hash = "sha256:e28157350f7ebf35008dd8e9e0fdb621f976e4230c881099c85e8cf07eaa50e2", size = 1181826, upload-time = "2026-05-15
T04:50:56.296Z" },
    { url = "https://files.pythonhosted.org/packages/db/c6/d393e3185a276505182f7abd93fe714f3c444a2be9180798fa052347504e/tiktoken-0.13.0-cp313-cp313t-m
usllinux_1_2_x86_64.whl", hash = "sha256:165cf1820ea4a354985c2490a5205d4cc74661c934aca79dd0368232fff94e0f", size = 1239489, upload-time = "2026-05-15T
04:50:57.918Z" },
    { url = "https://files.pythonhosted.org/packages/b7/4d/bc07d1f1635d4897a202acc0ae11c2886eaa7325c359ba4741b47bf8e225/tiktoken-0.13.0-cp313-cp313t-w
```

```
in_amd64.whl", hash = "sha256:6c43a675ca14f6f2749ba7f12075d37456015a24b859f2517b9beb4ef30807ec", size = 873820, upload-time = "2026-05-15T04:50:59.528
Z" },
    { url = "https://files.pythonhosted.org/packages/8c/93/0dd6adca026a616c3a92974566b43381eea4b475ce1f36c062b8271a9ac5/tiktoken-0.13.0-cp314-cp314-ma
cosx_10_13_x86_64.whl", hash = "sha256:eaaaef47c2406277181d2086484c317bf7fc433e2d5d03ff94f56b0dcec87471", size = 1034977, upload-time = "2026-05-15T04
:51:00.957Z" },
    { url = "https://files.pythonhosted.org/packages/d9/77/5ec6e6bc5b30bed6d93f7f2162d8f6b32437b3ba27cb527cfe004f6109c9/tiktoken-0.13.0-cp314-cp314-ma
cosx_11_0_arm64.whl", hash = "sha256:ca8b310bd93b3772cb1b7922d915446864860f562bdfe4825c63a0aed3fb28cd", size = 983635, upload-time = "2026-05-15T04:51
:02.629Z" },
    { url = "https://files.pythonhosted.org/packages/94/b0/c8ae9aff00d625c50659b4513e707a0462c4bf5d4d6cc1b802103225c02e/tiktoken-0.13.0-cp314-cp314-ma
nylinux_2_28_aarch64.whl", hash = "sha256:32e0c12305105002c047b3bb1070b0dd9a73b0cb3b2856a8972b810e7a4f5881", size = 1116036, upload-time = "2026-05-15
T04:51:04.082Z" },
    { url = "https://files.pythonhosted.org/packages/1b/ac/6a5dddd1d0a6018ecb389bd0353e6b4a515eb4d2286611bd0ace1937b9e1/tiktoken-0.13.0-cp314-cp314-ma
nylinux_2_28_x86_64.whl", hash = "sha256:5ba5fd62507a932d1241346179e3b39bc7bf7408f03c272652d93b3bedf5db24", size = 1135544, upload-time = "2026-05-15T
04:51:05.229Z" },
    { url = "https://files.pythonhosted.org/packages/f4/b8/585032b4384b2f7dcdaddcb52865c83a701a420d09e3c2b4a2be1c450c57/tiktoken-0.13.0-cp314-cp314-mu
sllinux_1_2_aarch64.whl", hash = "sha256:d108bc2d470fc53c8ecd24f2c0fd2b5f98c33e87cdb6aa2e9b8c5dced703d273", size = 1182217, upload-time = "2026-05-15T
04:51:06.517Z" },
    { url = "https://files.pythonhosted.org/packages/cd/b6/993ff1ded3958215fd341a847b8e5ffeb5de473f435296870d314fc91ac4/tiktoken-0.13.0-cp314-cp314-mu
sllinux_1_2_x86_64.whl", hash = "sha256:cb99cb5127449f58d0a2d5f5ccfb390d8dbdfd919c221246caaee29d8725ed51", size = 1239404, upload-time = "2026-05-15T0
4:51:07.843Z" },
    { url = "https://files.pythonhosted.org/packages/dd/3d/fef7e06e3b33e7538db0ced734cf9fe23b6832d2ac4990c119c377aec55e/tiktoken-0.13.0-cp314-cp314-wi
n_amd64.whl", hash = "sha256:115c4f26ffa11caac8b54eea35c2ad38c612c20a48d35dd15d70a02ac6f51f58", size = 918686, upload-time = "2026-05-15T04:51:08.925Z
" },
    { url = "https://files.pythonhosted.org/packages/c1/82/a7fc44582bc32ab00de988a2299bf77c077f59068b233109e34b7d6ca7e6/tiktoken-0.13.0-cp314-cp314t-m
acosx_10_13_x86_64.whl", hash = "sha256:472527e9132952f2fbf77cd290658bacf003d4d5a3fabc18e5fbd407cbae4d9b", size = 1034454, upload-time = "2026-05-15T0
4:51:10.035Z" },
    { url = "https://files.pythonhosted.org/packages/37/d0/24d8a890c14f432a05cea669c17bebeaa99f96a7c79523b590f564246411/tiktoken-0.13.0-cp314-cp314t-m
acosx_11_0_arm64.whl", hash = "sha256:4e2f67d27c9626cdd25fe33d9313c5cdb3d8d82da646b68d6eb8e7e9c20e6448", size = 982976, upload-time = "2026-05-15T04:5
1:11.23Z" },
    { url = "https://files.pythonhosted.org/packages/49/b7/2ab43f62788a9266187a9bfc1d3af99ad83e5eaa25fbef168a69cd5ad14f/tiktoken-0.13.0-cp314-cp314t-m
anylinux_2_28_aarch64.whl", hash = "sha256:2b920b35805cd64585a37c3dc7ce65fba4d2d36016be01e1d7942482ca29093a", size = 1115526, upload-time = "2026-05-1
5T04:51:12.608Z" },
    { url = "https://files.pythonhosted.org/packages/64/39/1494321ed323ce7a14d88e3cd6cb9058625977df1c6961ddc492bd10a9f3/tiktoken-0.13.0-cp314-cp314t-m
anylinux_2_28_x86_64.whl", hash = "sha256:493af3aa28a4aaf2e3d2600a2ee717252c9bf5ab38fff94eb5a02db5ab77e5ad", size = 1136466, upload-time = "2026-05-15
T04:51:13.926Z" },
    { url = "https://files.pythonhosted.org/packages/96/d9/dfd086aa2d918c563a140720e0ce296cada1634efd2783d5cf51e05f984e/tiktoken-0.13.0-cp314-cp314t-m
usllinux_1_2_aarch64.whl", hash = "sha256:6644c9c2b5cf3916f5a3641d7d12fdb3f006a7b3d9ff6acdaec44e29ab1ff91e", size = 1181863, upload-time = "2026-05-15
T04:51:15.025Z" },
    { url = "https://files.pythonhosted.org/packages/2f/68/a18b4f307086954fdae32714cb4f85562e34f9d34ab206e61f1816aa6018/tiktoken-0.13.0-cp314-cp314t-m
usllinux_1_2_x86_64.whl", hash = "sha256:5cb65b60b9408563676d874a3a4ee573370066f0dc4e29d84e82e989c6517424", size = 1239218, upload-time = "2026-05-15T
04:51:16.103Z" },
    { url = "https://files.pythonhosted.org/packages/16/5b/f2aa703a4fc5d2dff73460a7d46cc2f3f44aa0f3dd8eeb20d2a0ecf68862/tiktoken-0.13.0-cp314-cp314t-w
in_amd64.whl", hash = "sha256:85b78cc3a2c3d48723ca751fa981f1fedccd54194ca0471b957364353a898b07", size = 918110, upload-time = "2026-05-15T04:51:17.237
Z" },
]

[[package]]
name = "tinytag"
version = "2.3.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/9f/a4/a1d39cc10b43cbbae268127a1c38d689bc6a85cf966f9445bc9f1f5f517a/tinytag-2.3.0.tar.gz", has
h = "sha256:84850f8045424b944475b9754bc35c7e09bcae1ab08d1f88d9293aa33af39a27", size = 44379, upload-time = "2026-07-30T23:35:03.407Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/90/48/f9a955e0d27376dd0a6bcf7973818b19e5965dd847580aef2f08c6d6f64b/tinytag-2.3.0-py3-none-any.whl
", hash = "sha256:231ba5b2fb7a6db478f6dd344ebf20dfdfcd5907f142d475605d516dd7e8b9a8", size = 37155, upload-time = "2026-07-30T23:35:02.003Z" },
]

[[package]]
name = "tqdm"
version = "4.70.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "colorama", marker = "sys_platform == 'win32'" },
]
sdist = { url = "https://files.pythonhosted.org/packages/21/3b/6c24bec5be5e743ffd99576daa5cc077722fc7d5bbc00bd133fa0c698dc6/tqdm-4.70.0.tar.gz", hash
= "sha256:55b0b0dbd97462d06ebee91e4dac24ed4d4702be82b24f07e6c1d27e08cea220", size = 795438, upload-time = "2026-07-27T11:33:15.271Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/f9/1c/01bfd571a64e7f270e6bab5e33777debe0edc56759233ce84f27dec92d14/tqdm-4.70.0-py3-none-any.whl",
 hash = "sha256:7f585706bfddbdebf89daac705b2dfcc16890130727d3197ca62c732b4310953", size = 80184, upload-time = "2026-07-27T11:33:13.167Z" },
]

[[package]]
name = "truststore"
version = "0.10.4"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/53/a3/1585216310e344e8102c22482f6060c7a6ea0322b63e026372e6dcefcfd6/truststore-0.10.4.tar.gz",
 hash = "sha256:9d91bd436463ad5e4ee4aba766628dd6cd7010cf3e2461756b3303710eebc301", size = 26169, upload-time = "2025-08-12T18:49:02.73Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/19/97/56608b2249fe206a67cd573bc93cd9896e1efb9e98bce9c163bcdc704b88/truststore-0.10.4-py3-none-any
.whl", hash = "sha256:adaeaecf1cbb5f4de3b1959b42d41f6fab57b2b1666adb59e89cb0b53361d981", size = 18660, upload-time = "2025-08-12T18:49:01.46Z" },
]

[[package]]
name = "typing-extensions"
version = "4.16.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/f6/cc/6253133b5bb138fc3306cebfbda2c520f545d36b5be2c7255cc528bb45d6/typing_extensions-4.16.0.t
ar.gz", hash = "sha256:dc983d19a509c94dba722ee6abd33940f7c05a89e243c47e907eb4db6f1a43e5", size = 113555, upload-time = "2026-07-02T08:40:05.92Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/49/d3/b8441a820a491ddfc024b0b0cf0393375b75ea13866d9c66727e54c2fc80/typing_extensions-4.16.0-py3-n
one-any.whl", hash = "sha256:481caa481374e813c1b176ada14e97f1f67a4539ce9cfeb3f350d78d6370c2e8", size = 45571, upload-time = "2026-07-02T08:40:04.659Z"
 },
]

[[package]]
name = "typing-inspect"
version = "0.9.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "mypy-extensions" },
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/dc/74/1789779d91f1961fa9438e9a8710cdae6bd138c80d7303996933d117264a/typing_inspect-0.9.0.tar.g
z", hash = "sha256:b23fc42ff6f6ef6954e4852c1fb512cdd18dbea03134f91f856a95ccc9461f78", size = 13825, upload-time = "2023-05-24T20:25:47.612Z" }
wheels = [
```

```
    { url = "https://files.pythonhosted.org/packages/65/f3/107a22063bf27bdccf2024833d3445f4eea42b2e598abfbd46f6a63b6cb0/typing_inspect-0.9.0-py3-none-any.whl", hash = "sha256:9ee6fc59062311ef8547596ab6b955e1b8aa46242d854bfc78f4f6b0eff35f9f", size = 8827, upload-time = "2023-05-24T20:25:45.287Z" },
]

[[package]]
name = "typing-inspection"
version = "0.4.2"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "typing-extensions" },
]
sdist = { url = "https://files.pythonhosted.org/packages/55/e3/70399cb7dd41c10ac53367ae42139cf4b1ca5f36bb3dc6c9d33acdb43655/typing_inspection-0.4.2.tar.gz", hash = "sha256:ba561c48a67c5958007083d386c3295464928b01faa735ab8547c5692e87f464", size = 75949, upload-time = "2025-10-01T02:14:41.687Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/dc/9b/47798a6c91d8bdb567fe2698fe81e0c6b7cb7ef4d13da4114b41d239f65d/typing_inspection-0.4.2-py3-none-any.whl", hash = "sha256:4ed1cacbdc298c220f1bd249ed5287caa16f34d44ef4e9c3d0cbad5b521545e7", size = 14611, upload-time = "2025-10-01T02:14:40.154Z" },
]

[[package]]
name = "tzdata"
version = "2026.3"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/92/ff/5a28bdfd8c3ebec42564ac7d0e54ca3db65044a9314a97f9564fa7a1e926/tzdata-2026.3.tar.gz", hash = "sha256:4a1518b8993086a7982523e071643f3c0e5f213e75b21318e78bcabfff9d1415", size = 198674, upload-time = "2026-07-10T08:50:37.887Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e5/6d/b53b99a9f2766d095985947a5782f1702cabb129a34f7a802d7197af832f/tzdata-2026.3-py2.py3-none-any.whl", hash = "sha256:dc096730c87af6cab1b171c9d532be840741ff5d459015e7f6947bd7d7e54931", size = 348168, upload-time = "2026-07-10T08:50:36.46Z" },
]

[[package]]
name = "urllib3"
version = "2.7.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/53/0c/06f8b233b8fd13b9e5ee11424ef85419ba0d8ba0b3138bf360be2ff56953/urllib3-2.7.0.tar.gz", hash = "sha256:231e0ec3b63ceb14667c67be60f2f2c40a518cb38b03af60abc813da26505f4c", size = 433602, upload-time = "2026-05-07T16:13:18.596Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/7f/3e/5db95bcf282c52709639744ca2a8b149baccf648e39c8cc87553df9eae0c/urllib3-2.7.0-py3-none-any.whl", hash = "sha256:9fb4c81ebbb1ce9531cce37674bbc6f1360472bc18ca9a553ede278ef7276897", size = 131087, upload-time = "2026-05-07T16:13:17.151Z" },
]

[[package]]
name = "uuid-utils"
version = "0.17.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/e7/91/63938e0e7e7876658e5e40178e7c0735b53527886fe11797a11699c55edd/uuid_utils-0.17.0.tar.gz",
 hash = "sha256:abb5667a36119019b3fa320c4d10c21ebccfcc87c8a739e6a0056cee7f48dde2", size = 43220, upload-time = "2026-07-09T13:49:58.433Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/20/80/a7e685968e3cec99d6fe2fb25d0f5726310e1bba356da68c13dfd8b7d140/uuid_utils-0.17.0-cp312-cp312-macosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:9205068badf453d2f0821fd5d340389b4679992d7ff79d4f3e5608996dd1b287", size = 556403, upload-time = "2026-07-09T13:48:27.022Z" },
    { url = "https://files.pythonhosted.org/packages/56/47/3102d93bcb7b0bfe6bede63ff8f221a7f91348e10a37f682773be27c56d9/uuid_utils-0.17.0-cp312-cp312-macosx_10_12_x86_64.whl", hash = "sha256:0fcca4e838af9ac9243b3358d7c14afa4dca286a87781124c272d6c4cad9c968", size = 285608, upload-time = "2026-07-09T13:48:28.769Z" },
    { url = "https://files.pythonhosted.org/packages/55/fb/d59695f0f8db065b93c63316eaafa05a22d75a0486978a33736c52c646d5/uuid_utils-0.17.0-cp312-cp312-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:0f3729e839209f3457d0d8b6a35a376fdf65577a5aecaf4cc3587d3305759ba6", size = 319926, upload-time = "2026-07-09T13:48:29.965Z" },
    { url = "https://files.pythonhosted.org/packages/5a/03/62fabcd1e990e07a0e220e8d552af45bc16f107fa8e55c2014a706bb1a1e/uuid_utils-0.17.0-cp312-cp312-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:3dac0ad0cd9a2818d1775215365a4e8c2f8ada215529dd26f3f8cceeb67a6988", size = 327172, upload-time = "2026-07-09T13:48:31.187Z" },
    { url = "https://files.pythonhosted.org/packages/d9/37/a5081391338b459e2f8d8b12581f00f8caa6317fab510e0e85c18c59e938/uuid_utils-0.17.0-cp312-cp312-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:e671b2322ef09106ecb1ca0f4c398b134d5e2c1f80d7a4f3336847a3072c0e94", size = 439075, upload-time = "2026-07-09T13:48:32.295Z" },
    { url = "https://files.pythonhosted.org/packages/59/30/91795bd01e17a13661280d4899fbf38fb05e3f38e873f9aaec106ec30aa0/uuid_utils-0.17.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:8eb3e5caca8d3a6f72ea4cce024583f989f6f2e9186f98800213fff0176e8bcc", size = 320247, upload-time = "2026-07-09T13:48:33.64Z" },
    { url = "https://files.pythonhosted.org/packages/e5/11/09102b78303e4eb62069d6d88ef9fd661dc523e8f429e1fd67eaa78a6f44/uuid_utils-0.17.0-cp312-cp312-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:8b72c2002202038666bf647f9a790906214c7c11cd0d6efef77b7d07bef3034a", size = 344738, upload-time = "2026-07-09T13:48:34.786Z" },
    { url = "https://files.pythonhosted.org/packages/74/f9/be95bad6954b60328878c3800258f01a6accd24fd75112d13f023462d53f/uuid_utils-0.17.0-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:4e2ac1c0b56f2c91b6f158e29ed96b1503223fe8aa6e79b1be1dc55bd8a5131c", size = 496845, upload-time = "2026-07-09T13:48:36.057Z" },
    { url = "https://files.pythonhosted.org/packages/2d/02/8a19a34e0530d987488a068a71576a236f5c8c746630b870b57f71eb24ef/uuid_utils-0.17.0-cp312-cp312-musllinux_1_2_armv7l.whl", hash = "sha256:6c142bd0cb4dba31c10babe00d59f7ef6460f0ef55eaa9c1a9da270684af996a", size = 603233, upload-time = "2026-07-09T13:48:37.512Z" },
    { url = "https://files.pythonhosted.org/packages/f4/a8/b1abab36ff73b0248d82179816467f6d39a2e80fd64329a895ca94f3508e/uuid_utils-0.17.0-cp312-cp312-musllinux_1_2_i686.whl", hash = "sha256:e252db239eb41c32248e096e0d170bce5896a4fd3405556362bc3dd83d912206", size = 561401, upload-time = "2026-07-09T13:48:38.977Z" },
    { url = "https://files.pythonhosted.org/packages/61/91/70e7b528b351cc03a9ca43e6116371cdde31bb12bcead7ca2ca1367366cc/uuid_utils-0.17.0-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:237722b6581bb5b4eb4cefbcbe5c6e2980a440aabe781fbe50ebf1cb71eee4cc", size = 525314, upload-time = "2026-07-09T13:48:40.599Z" },
    { url = "https://files.pythonhosted.org/packages/d6/f6/9167e90cf9937d6558f92d022ff3024a69d938a514d9c8faa4080f73b001/uuid_utils-0.17.0-cp312-cp312-win32.whl", hash = "sha256:46a73cacdf512f473a81f65dbf84186e08cfe6e9118fa582b6c6b33a8288a30d", size = 166831, upload-time = "2026-07-09T13:48:41.862Z" },
    { url = "https://files.pythonhosted.org/packages/5c/7d/0b889654d9ee3413f810cf4685e241285f650d98a4103ac9f3c6bcc95f29/uuid_utils-0.17.0-cp312-cp312-win_amd64.whl", hash = "sha256:e59b60a0a4cb7541480e02090d37dc2df3b72df4c2e776fff64ce3a4e3dd4637", size = 172944, upload-time = "2026-07-09T13:48:42.992Z" },
    { url = "https://files.pythonhosted.org/packages/be/35/8c6e1bf65e4d400352885dadc656ad6d0af96e89231e3f04686bc2197128/uuid_utils-0.17.0-cp312-cp312-win_arm64.whl", hash = "sha256:d561a4c5747a1e6c7fa7c49a0292e78b4e8c456332caa084fc7abad8de828652", size = 172459, upload-time = "2026-07-09T13:48:44.271Z" },
    { url = "https://files.pythonhosted.org/packages/d2/dd/614fb9912157ac0128e6050859ccf06d9f13df9a944a803e8f80f6157e38/uuid_utils-0.17.0-cp313-cp313-macosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:d11a7bc1e02da8984d32e6de9e0826c6edac00eac17de270f372bf32f9a0af63", size = 557259, upload-time = "2026-07-09T13:48:45.664Z" },
    { url = "https://files.pythonhosted.org/packages/3e/11/d072711704de3d21bec08b6c2f36a215200ca1d5e01a390ea1ac434080a0/uuid_utils-0.17.0-cp313-cp313-macosx_10_12_x86_64.whl", hash = "sha256:7a49f47ac26df3e431c56b825c1bae8e6d3d591fdbb7438c227cc9845a7e3d73", size = 286271, upload-time = "2026-07-09T13:48:47.018Z" },
    { url = "https://files.pythonhosted.org/packages/18/6d/8a63e5eb2d5a6ba69a6c2036e305075bd6f5a022e7ea25fc6ce0eb7c51d2/uuid_utils-0.17.0-cp313-cp313-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:32df1944808877702ceea398c103881c09a679bb672a215e01c2a84231266bf9", size = 320025, upload-time = "2026-07-09T13:48:48.208Z" },
    { url = "https://files.pythonhosted.org/packages/f7/2d/bdc2caf9719d9090d7c46043242ae6136cba4f7a7ee384992ab905ad9aa1/uuid_utils-0.17.0-cp313-cp313-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:98c88d3edd08e7245562e9815996dbc6f0bd4745e1c76462f24af5ae4e187dd1", size = 327931, upload-time = "2026-07-09T13:48:49.673Z" },
    { url = "https://files.pythonhosted.org/packages/b6/33/9219d09d51ead282b578b2a4e0a515c2cce3ec52076cada8bfb7e35727d5/uuid_utils-0.17.0-cp313-cp313-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:5a4370089c8b2e42f1db51d76408c7fa8eaa2934bf854d17983d16179c07c098", size = 438537, upload-time = "2026-07-09T13:48:50.842Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/d8/79/e8e0f8b3955f2081c116157119d87659937893242eb834aa170da04d660b/uuid_utils-0.17.0-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:09a55b7a5ae764985cb46467496a1787678d0a1400356157a080ad95b1a36869", size = 320656, upload-time = "2026-07-09T13:48:52.164Z" },
    { url = "https://files.pythonhosted.org/packages/d5/5e/d1ceddc430ff04b6e21704b2030d4438074a2f478b265dab43da957791c1/uuid_utils-0.17.0-cp313-cp313-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:56aa6488b931246fae11924e4bd0e2b32677e63945eecb71c29e3c2ca0dc3131", size = 345310, upload-time = "2026-07-09T13:48:54.076Z" },
    { url = "https://files.pythonhosted.org/packages/d5/62/89438e12f389a843e626b7e37691319a057b3d6b80914609106891faadda/uuid_utils-0.17.0-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:309a35f12d99dde19032bc2259cda6431c85eeac0879134dc777cc3087d7e1cb", size = 496771, upload-time = "2026-07-09T13:48:55.365Z" },
    { url = "https://files.pythonhosted.org/packages/87/d2/eedcd99f522d60e238ead03844f0d51743ba84d33044959e230b756bf212/uuid_utils-0.17.0-cp313-cp313-musllinux_1_2_armv7l.whl", hash = "sha256:21c79b61ff750abcf057163dd764ccb6196cde7a26cda1b31b45cd97769e03b3", size = 603631, upload-time = "2026-07-09T13:48:56.746Z" },
    { url = "https://files.pythonhosted.org/packages/0e/a8/bb1b38aaddd7243b6e562c6694f499bf094800918316192fd8cb2cdc2620/uuid_utils-0.17.0-cp313-cp313-musllinux_1_2_i686.whl", hash = "sha256:4134353bfe3026ddab8e886002dc52bc5a0ab04611aabb0eaae23c32e6e57f64", size = 562008, upload-time = "2026-07-09T13:48:58.241Z" },
    { url = "https://files.pythonhosted.org/packages/b4/77/5f7ed930dc105e293845c09e4d5bd84076318a12f45a46783e1af64906d7/uuid_utils-0.17.0-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:7c89359affecebe2e39e6a116d069b363c936511a9572b308402489a26957d89", size = 525527, upload-time = "2026-07-09T13:48:59.784Z" },
    { url = "https://files.pythonhosted.org/packages/fd/25/1b55697adf6811a6f92cff6340e6b03e31fd6bc51066a5c10698c29b3679/uuid_utils-0.17.0-cp313-cp313-pyemscripten_2025_0_wasm32.whl", hash = "sha256:6a019a31bc4db89a0903a3e4f6b218571f3a6ff0ad4b3d3fe1c8f91a05ff6e3e", size = 97965, upload-time = "2026-07-09T13:49:01.217Z" },
    { url = "https://files.pythonhosted.org/packages/26/bf/cd729343de4684230be8a966bad7bfc2cf10ce3e643b1189a8b5370dbe35/uuid_utils-0.17.0-cp313-cp313-win32.whl", hash = "sha256:b3131a82d0c7611f0aa480a6d36929e001a3f54ba0fc029a8118a5863cce513c", size = 167316, upload-time = "2026-07-09T13:49:02.354Z" },
    { url = "https://files.pythonhosted.org/packages/76/f0/e602ae0a1b139a7826e5189b93d91902564def06d5006324fd2faf82c8fc/uuid_utils-0.17.0-cp313-cp313-win_amd64.whl", hash = "sha256:9e311f908d2f842fca4c7dcebc4f10306b8089b204ef04cf6704b4332c9ff6ff", size = 173630, upload-time = "2026-07-09T13:49:03.529Z" },
    { url = "https://files.pythonhosted.org/packages/1a/52/024ebece265b387154115dc4f1d9727174ef82623069f4bec8b7ed7e73f7/uuid_utils-0.17.0-cp313-cp313-win_arm64.whl", hash = "sha256:c351737e2e65497c7200ab4ffb8af97e9f48be6488309abdd265fe08d66ee92f", size = 173214, upload-time = "2026-07-09T13:49:04.836Z" },
    { url = "https://files.pythonhosted.org/packages/56/44/e2fd3fdf356e1b55d2acf1b956b4f3f29ffb215a99c387eba04b1c5fba66/uuid_utils-0.17.0-cp314-cp314-macosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:673d89cc434cc9b97a0b4cf61272f6fca70a81f64eb0afbface2a0d9f77f06cd", size = 562232, upload-time = "2026-07-09T13:49:06.201Z" },
    { url = "https://files.pythonhosted.org/packages/19/28/65e0980d668a6d44e699f59d1acf43d6b5d4893592c115ce7c680bb4dfa1/uuid_utils-0.17.0-cp314-cp314-macosx_10_12_x86_64.whl", hash = "sha256:387cf7437c94ddec08651a0f1081381299c7075bc48a6251d8922bf39973378a", size = 287858, upload-time = "2026-07-09T13:49:07.45Z" },
    { url = "https://files.pythonhosted.org/packages/8f/8d/5e97bcebc90fb6a10f98af3dc1ba552e04183aba59e2edc0b9cf486dd998/uuid_utils-0.17.0-cp314-cp314-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:220b52746d99e11964badac3c0869016e0c24bafb70a7dd5c2c072a6be3da9cc", size = 321587, upload-time = "2026-07-09T13:49:09.489Z" },
    { url = "https://files.pythonhosted.org/packages/8b/d7/88b2a2370cc3d455ba0515fb6f5c8f7ac0c0f55a86801b6e56a432f22c17/uuid_utils-0.17.0-cp314-cp314-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:0ab4a66e7a035ad6625cfc1fbdb34f5c2d25a80ae1ef4bfee458ea2036333c6d", size = 328964, upload-time = "2026-07-09T13:49:11.292Z" },
    { url = "https://files.pythonhosted.org/packages/bd/0f/181c5da673953dfc0958cb4fb3a4984a9098673ddb05cac68e994bc8511b/uuid_utils-0.17.0-cp314-cp314-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:5641071337eb11d61a001ea08793bf72216f3241f0a433ed2764804b2a3e3cc7", size = 442909, upload-time = "2026-07-09T13:49:12.644Z" },
    { url = "https://files.pythonhosted.org/packages/ec/38/5c5e665af542884a8fd3c61725c38453239e13940326b5b70f3ef8881a97/uuid_utils-0.17.0-cp314-cp314-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:9082e709014946b1f6e96ae6ecd93652efca2d2a6a3ab67dbe151c8b4bf193a4", size = 323076, upload-time = "2026-07-09T13:49:13.897Z" },
    { url = "https://files.pythonhosted.org/packages/f5/35/7de97de18cbf226c2a4f2104ad15e56ca4491717c81c0b71795c0c585b4e/uuid_utils-0.17.0-cp314-cp314-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:1fd6f0e8a162dc0e9255b6aebe3cd175e76c33202f1bf39da9e6294b93db0099", size = 347360, upload-time = "2026-07-09T13:49:15.237Z" },
    { url = "https://files.pythonhosted.org/packages/26/a1/9915d5dd59fdd1957ded5d188c0ea0b9db5a1d84d42c8d8828a7b83b366e/uuid_utils-0.17.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:d63010803d7c368963bbe6f7ec379593e76dd581d7db0f29118d88713c9e0354", size = 499267, upload-time = "2026-07-09T13:49:16.774Z" },
    { url = "https://files.pythonhosted.org/packages/c0/05/88108405262ec850cea0f95733445d6873e5772af3292baabd9ef8457740/uuid_utils-0.17.0-cp314-cp314-musllinux_1_2_armv7l.whl", hash = "sha256:a46bedc273b6f58f11dee816ff74999625ef8d007890f411b7a4975bf1c89330", size = 604940, upload-time = "2026-07-09T13:49:18.147Z" },
    { url = "https://files.pythonhosted.org/packages/89/d5/6dbcd300de47cc443cff2656cd5327a385751213dcb2101cfee7388170b2/uuid_utils-0.17.0-cp314-cp314-musllinux_1_2_i686.whl", hash = "sha256:405233a5f625b3d995648f4647fa6befa4567cf3f74e1f6b9837e16f7310f0e0", size = 564172, upload-time = "2026-07-09T13:49:19.593Z" },
    { url = "https://files.pythonhosted.org/packages/ab/94/e8057f2288a415fba8a978bca4b589f5cb6b91a028a5dc07a1775938b33f/uuid_utils-0.17.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:b6c5d2d71e1f17329150ad9427d27f4a3f29a01792e7ecdc64a98ac5368fc4d5", size = 528533, upload-time = "2026-07-09T13:49:21.075Z" },
    { url = "https://files.pythonhosted.org/packages/f0/6b/31713148c77e48e62f51aa042a98a54a8be0396912ea5130f83f52ae722d/uuid_utils-0.17.0-cp314-cp314-pyemscripten_2026_0_wasm32.whl", hash = "sha256:f7e9b8728ba07a3cb2f29d5aa1a266c2664eb8ef0fd43afa34627c92f7fac8f0", size = 99197, upload-time = "2026-07-09T13:49:22.351Z" },
    { url = "https://files.pythonhosted.org/packages/f3/f3/ca6f6ac5428312df8ed632f6dd9f9e6aba23090471fcdeae53eab027e8b3/uuid_utils-0.17.0-cp314-cp314-win32.whl", hash = "sha256:58838921e377791ef22c64cc92141bfae030f43651ff9272f0f28a208a9e6a5a", size = 169540, upload-time = "2026-07-09T13:49:23.563Z" },
    { url = "https://files.pythonhosted.org/packages/c6/cd/7ede0db66411fa09817d79b680f7454ea9bee2d374e1922e4efd065760a3/uuid_utils-0.17.0-cp314-cp314-win_amd64.whl", hash = "sha256:42275ebd0e8e74e32cdbfb8bd88fc99576567d51d54a508020611fd8f4f463a0", size = 175984, upload-time = "2026-07-09T13:49:24.703Z" },
    { url = "https://files.pythonhosted.org/packages/f0/81/533b5f80cd4918c0693f4e1b7b90ceb1caa45f4266ae8b528135d7ecca5d/uuid_utils-0.17.0-cp314-cp314-win_arm64.whl", hash = "sha256:b5d11cccba076a32321ef1380dea956821f0b51794ef59df64e58fb1cd543aae", size = 174749, upload-time = "2026-07-09T13:49:25.886Z" },
    { url = "https://files.pythonhosted.org/packages/a0/13/f400ac39d06fd8be5b099c09e41bb975205926722a3e8d53348817cb7ff9/uuid_utils-0.17.0-cp314-cp314t-macosx_10_12_x86_64.macosx_11_0_arm64.macosx_10_12_universal2.whl", hash = "sha256:fae8b282f0cb22a5de222999f7723f4e5ec04f6fcdf4aaef879b5b36625ae2b0", size = 562610, upload-time = "2026-07-09T13:49:27.374Z" },
    { url = "https://files.pythonhosted.org/packages/03/8c/c71c8312304c56f6d0bcba87cd402fa79bec35d18ffc8c41954196ca68e5/uuid_utils-0.17.0-cp314-cp314t-macosx_10_12_x86_64.whl", hash = "sha256:967955620df45e6cffe2e9950cb9903cb455649396f896b26b04363a91a5054b", size = 289473, upload-time = "2026-07-09T13:49:28.989Z" },
    { url = "https://files.pythonhosted.org/packages/bb/cd/522117e2e5184ca1d4f0f85ee833e9e21bd8c6b99eff8a4d1a8e5a194e33/uuid_utils-0.17.0-cp314-cp314t-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:375cde148430d60a4a07c03abaa0774c4fddfdd90de99b4ba02f24088bc9d750", size = 321600, upload-time = "2026-07-09T13:49:30.4Z" },
    { url = "https://files.pythonhosted.org/packages/6d/f4/0d81f9bd346fc717bc561c08fa6457e0328966eb76e536b938fe77d56459/uuid_utils-0.17.0-cp314-cp314t-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:975c17da26c5b9d46c336b03c52a057ac28378d6f9d98b58d32a038589bb3912", size = 329569, upload-time = "2026-07-09T13:49:31.732Z" },
    { url = "https://files.pythonhosted.org/packages/5e/41/26e1363f36a94c9e8ec2dd21d5f63088d3e7c723adbb12dcc8fdc77be417/uuid_utils-0.17.0-cp314-cp314t-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:3150d836290c88f1d26eb59c4db280d87417dd3bfaadd2889c77416c8f0ff6fa", size = 442051, upload-time = "2026-07-09T13:49:33.024Z" },
    { url = "https://files.pythonhosted.org/packages/2b/a7/2c1ed1b34d7df7fdcc11c28fd26d94d44843b37d9af2435ff9fd8abdbc08/uuid_utils-0.17.0-cp314-cp314t-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:9472a8de37faf8bd216c628e0e68c8f6bef730d3ba0a5060f3b0fa460c992ac2", size = 324372, upload-time = "2026-07-09T13:49:34.554Z" },
    { url = "https://files.pythonhosted.org/packages/78/bf/328d3c6bb22c496944a1b3b732207d71aa6964eb604e5e3b9dcb91ed0a00/uuid_utils-0.17.0-cp314-cp314t-manylinux_2_5_i686.manylinux1_i686.whl", hash = "sha256:d27c531edb8d1f38ca2eddaa1fa24913a460aeb721f2efd4ef42a124ce94e354", size = 348548, upload-time = "2026-07-09T13:49:35.898Z" },
    { url = "https://files.pythonhosted.org/packages/3e/76/a07de5cb7b90582fdbbc830fd19be129cbbb9897cfe239fef469d7bd2d09/uuid_utils-0.17.0-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:5670c52a438e21483ce715776144914a4e2a2a5c62d9dee15f8a3e90cf128ae6", size = 498985, upload-time = "2026-07-09T13:49:37.142Z" },
    { url = "https://files.pythonhosted.org/packages/f4/62/9966e46ae34fcec6b06119631fb3c09705ea78835035ce3a82d3348eb61a/uuid_utils-0.17.0-cp314-cp314t-musllinux_1_2_armv7l.whl", hash = "sha256:6f29689a76fe7a49cbd629a794d0ec1eab48814e323a00a146a741b0195bde68", size = 605183, upload-time = "2026-07-09T13:49:38.648Z" },
    { url = "https://files.pythonhosted.org/packages/d7/4e/bb962ba0fe31e903b199f22cf4c1a6cba35a8987aef526d287277ab8ca8b/uuid_utils-0.17.0-cp314-cp314t-musllinux_1_2_i686.whl", hash = "sha256:4441600447d340ae103a353f01dbcd22ff680e5ee1a22988efe8d7b791d8fdb3", size = 565412, upload-time = "2026-07-09T13:49:40.115Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/ce/9e/122adfeeeae8a84ccfd43bce627b104d12a2180a93bffd2c0e1b54dad7a6/uuid_utils-0.17.0-cp314-cp314t
-musllinux_1_2_x86_64.whl", hash = "sha256:e7b04935a79c03c41ad08d0a5f390aac968bfb561f1268897bc5b0f077971efd", size = 529885, upload-time = "2026-07-09
T13:49:41.513Z" },
    { url = "https://files.pythonhosted.org/packages/b3/4f/257304dded339dc35fc9bf35722ac68fd4fdb930f255b8f7bccdf74ebba9/uuid_utils-0.17.0-cp314-cp314t
-win32.whl", hash = "sha256:239d8a281fe10bae33205b5d43185834d556b18434e0a113b5dc1dfb2fd97e91", size = 169472, upload-time = "2026-07-09T13:49:42.871Z"
 },
    { url = "https://files.pythonhosted.org/packages/35/c8/e78c06db7e9ce317ce7b8759ff2058333eac75caa8c22b75f0059589c9be/uuid_utils-0.17.0-cp314-cp314t
-win_amd64.whl", hash = "sha256:e288a06cbbbcd01b44386e767985c9e21d2ad9bf59829aa7058d9a2a494804ab", size = 176271, upload-time = "2026-07-09T13:49:44.1
05Z" },
    { url = "https://files.pythonhosted.org/packages/a7/11/bd1c70e1ad3301163cebe66c8d26de26e6814d52f642a849448bd2833626/uuid_utils-0.17.0-cp314-cp314t
-win_arm64.whl", hash = "sha256:1776a80d16369999b21627028cc5dbce819be83e1e079fdd7a51b587d2916db9", size = 175004, upload-time = "2026-07-09T13:49:45.5
91Z" },
]

[[package]]
name = "uvicorn"
version = "0.52.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "click" },
    { name = "h11" },
]
sdist = { url = "https://files.pythonhosted.org/packages/03/18/ccce41535dee1be77735592bd19965f3972c82e07ee703d324709496b716/uvicorn-0.52.1.tar.gz", ha
sh = "sha256:112ec661814189acbccd3f7b86460147cc065fc92c0821afa78918780e4354dd", size = 100571, upload-time = "2026-08-01T18:19:30.732Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/c7/d5/68e6e9bca63c0badf67002890a46d3784c958de45b65e1275ec583ca1f06/uvicorn-0.52.1-py3-none-any.wh
l", hash = "sha256:e4403f9d93188cf9d1088e9f40e3acd12630e2df8675316704379a7fc20fff6a", size = 79859, upload-time = "2026-08-01T18:19:29.294Z" },
]

[package.optional-dependencies]
standard = [
    { name = "httptools" },
    { name = "python-dotenv" },
    { name = "pyyaml" },
    { name = "uvloop", marker = "platform_python_implementation != 'PyPy' and sys_platform != 'cygwin' and sys_platform != 'win32'" },
    { name = "watchfiles" },
    { name = "websockets" },
]

[[package]]
name = "uvloop"
version = "0.22.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/06/f0/18d39dbd1971d6d62c4629cc7fa67f74821b0dc1f5a77af43719de7936a7/uvloop-0.22.1.tar.gz", has
h = "sha256:6c84bae345b9147082b17371e3dd5d42775bddce91f885499017f4607fdaf39f", size = 2443250, upload-time = "2025-10-16T22:17:19.342Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/3d/ff/7f72e8170be527b4977b033239a83a68d5c881cc4775fca255c677f7ac5d/uvloop-0.22.1-cp312-cp312-maco
sx_10_13_universal2.whl", hash = "sha256:fe94b4564e865d968414598eea1a6de60adba0c040ba4ed05ac1300de402cd42", size = 1359936, upload-time = "2025-10-16T
22:16:29.436Z" },
    { url = "https://files.pythonhosted.org/packages/c3/c6/e5d433f88fd54d81ef4be58b2b7b0cea13c442454a1db703a1eea0db1a59/uvloop-0.22.1-cp312-cp312-maco
sx_10_13_x86_64.whl", hash = "sha256:51eb9bd88391483410daad430813d982010f9c9c89512321f5b60e2cddbddddd6", size = 752769, upload-time = "2025-10-16T22:16
:30.493Z" },
    { url = "https://files.pythonhosted.org/packages/24/68/a6ac446820273e71aa762fa21cdcc09861edd3536ff47c5cd3b7afb10eeb/uvloop-0.22.1-cp312-cp312-many
linux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:700e674a166ca5778255e0e1dc4e9d79ab2acc57b9171b79e65feba7184b3370"
, size = 4317413, upload-time = "2025-10-16T22:16:31.644Z" },
    { url = "https://files.pythonhosted.org/packages/5f/6f/e62b4dfc7ad6518e7eff2516f680d02a0f6eb62c0c212e152ca708a0085e/uvloop-0.22.1-cp312-cp312-many
linux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:7b5b1ac819a3f946d3b2ee07f09149578ae76066d70b44df3fa990add49a82e4", s
ize = 4426307, upload-time = "2025-10-16T22:16:32.917Z" },
    { url = "https://files.pythonhosted.org/packages/90/60/97362554ac21e20e81bcef1150cb2a7e4ffdaf8ea1e5b2e8bf7a053caa18/uvloop-0.22.1-cp312-cp312-musl
linux_1_2_aarch64.whl", hash = "sha256:e047cc068570bac9866237739607d1313b9253c3051ad84738cbb095be0537b2", size = 4131970, upload-time = "2025-10-16T22
:16:34.015Z" },
    { url = "https://files.pythonhosted.org/packages/99/39/6b3f7d234ba3964c428a6e40006340f53ba37993f46ed6e111c6e9141d18/uvloop-0.22.1-cp312-cp312-musl
linux_1_2_x86_64.whl", hash = "sha256:512fec6815e2dd45161054592441ef76c830eddaad55c8aa30952e6fe1ed07c0", size = 4296343, upload-time = "2025-10-16T22:
16:35.149Z" },
    { url = "https://files.pythonhosted.org/packages/89/8c/182a2a593195bfd39842ea68ebc084e20c850806117213f5a299dfc513d9/uvloop-0.22.1-cp313-cp313-maco
sx_10_13_universal2.whl", hash = "sha256:561577354eb94200d75aca23fbde86ee11be36b00e52a4eaf8f50fb0c86b7705", size = 1358611, upload-time = "2025-10-16T
22:16:36.833Z" },
    { url = "https://files.pythonhosted.org/packages/d2/14/e301ee96a6dc95224b6f1162cd3312f6d1217be3907b79173b06785f2fe7/uvloop-0.22.1-cp313-cp313-maco
sx_10_13_x86_64.whl", hash = "sha256:1cdf5192ab3e674ca26da2eada35b288d2fa49fdd0f357a19f0e7c4e7d5077c8", size = 751811, upload-time = "2025-10-16T22:16
:38.275Z" },
    { url = "https://files.pythonhosted.org/packages/b7/02/654426ce265ac19e2980bfd9ea6590ca96a56f10c76e63801a2df01c0486/uvloop-0.22.1-cp313-cp313-many
linux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:6e2ea3d6190a2968f4a14a23019d3b16870dd2190cd69c8180f7c632d21de68d"
, size = 4288562, upload-time = "2025-10-16T22:16:39.375Z" },
    { url = "https://files.pythonhosted.org/packages/15/c0/0be24758891ef825f2065cd5db8741aaddabe3e248ee6acc5e8a80f04005/uvloop-0.22.1-cp313-cp313-many
linux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:0530a5fbad9c9e4ee3f2b33b148c6a64d47bbad8000ea63704fa8260f4cf728e", s
ize = 4366890, upload-time = "2025-10-16T22:16:40.547Z" },
    { url = "https://files.pythonhosted.org/packages/d2/53/8369e5219a5855869bcee5f4d317f6da0e2c669aecf0ef7d371e3d084449/uvloop-0.22.1-cp313-cp313-musl
linux_1_2_aarch64.whl", hash = "sha256:bc5ef13bbc10b5335792360623cc378d52d7e62c2de64660616478c32cd0598e", size = 4119472, upload-time = "2025-10-16T22
:16:41.694Z" },
    { url = "https://files.pythonhosted.org/packages/f8/ba/d69adbe699b768f6b29a5eec7b47dd610bd17a69de51b251126a801369ea/uvloop-0.22.1-cp313-cp313-musl
linux_1_2_x86_64.whl", hash = "sha256:1f38ec5e3f18c8a10ded09742f7fb8de0108796eb673f30ce7762ce1b8550cad", size = 4239051, upload-time = "2025-10-16T22:
16:43.224Z" },
    { url = "https://files.pythonhosted.org/packages/90/cd/b62bdeaa429758aee8de8b00ac0dd26593a9de93d302bff3d21439e9791d/uvloop-0.22.1-cp314-cp314-maco
sx_10_13_universal2.whl", hash = "sha256:3879b88423ec7e97cd4eba2a443aa26ed4e59b45e6b76aabf13fe2f27023a142", size = 1362067, upload-time = "2025-10-16T
22:16:44.503Z" },
    { url = "https://files.pythonhosted.org/packages/0d/f8/a132124dfda0777e489ca86732e85e69afcd1ff7686647000050ba670689/uvloop-0.22.1-cp314-cp314-maco
sx_10_13_x86_64.whl", hash = "sha256:4baa86acedf1d62115c1dc6ad1e17134476688f08c6efd8a2ab076e815665c74", size = 752423, upload-time = "2025-10-16T22:16
:45.968Z" },
    { url = "https://files.pythonhosted.org/packages/a3/94/94af78c156f88da4b3a733773ad5ba0b164393e357cc4bd0ab2e2677a7d6/uvloop-0.22.1-cp314-cp314-many
linux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:297c27d8003520596236bdb2335e6b3f649480bd09e00d1e3a99144b691d2a35"
, size = 4272437, upload-time = "2025-10-16T22:16:47.451Z" },
    { url = "https://files.pythonhosted.org/packages/b5/35/60249e9fd07b32c665192cec7af29e06c7cd96fa1d08b84f012a56a0b38e/uvloop-0.22.1-cp314-cp314-many
linux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:c1955d5a1dd43198244d47664a5858082a3239766a839b2102a269aaff7a4e25", s
ize = 4292101, upload-time = "2025-10-16T22:16:49.318Z" },
    { url = "https://files.pythonhosted.org/packages/02/62/67d382dfcb25d0a98ce73c11ed1a6fba5037a1a1d533dcbb7cab033a2636/uvloop-0.22.1-cp314-cp314-musl
linux_1_2_aarch64.whl", hash = "sha256:b31dc2fccbd42adc73bc4e7cdbae4fc5086cf378979e53ca5d0301838c5682c6", size = 4114158, upload-time = "2025-10-16T22
:16:50.517Z" },
    { url = "https://files.pythonhosted.org/packages/f0/7a/f1171b4a882a5d13c8b7576f348acfe6074d72eaf52cccef752f748d4a9f/uvloop-0.22.1-cp314-cp314-musl
linux_1_2_x86_64.whl", hash = "sha256:93f617675b2d03af4e72a5333ef89450dfaa5321303ede6e67ba9c9d26878079", size = 4177360, upload-time = "2025-10-16T22:
16:52.646Z" },
    { url = "https://files.pythonhosted.org/packages/79/7b/b01414f31546caf0919da80ad57cbfe24c56b151d12af68cee1b04922ca8/uvloop-0.22.1-cp314-cp314t-mac
osx_10_13_universal2.whl", hash = "sha256:37554f70528f60cad66945b885eb01f1bb514f132d92b6eeed1c90fd54ed6289", size = 1454790, upload-time = "2025-10-16
T22:16:54.355Z" },
    { url = "https://files.pythonhosted.org/packages/d4/31/0bb232318dd838cad3fa8fb0c68c8b40e1145b32025581975e18b11fab40/uvloop-0.22.1-cp314-cp314t-mac
osx_10_13_x86_64.whl", hash = "sha256:b76324e2dc033a0b2f435f33eb88ff9913c156ef78e153fb210e03c13da746b3", size = 796783, upload-time = "2025-10-16T22:1
```

```
6:55.906Z" },
    { url = "https://files.pythonhosted.org/packages/42/38/c9b09f3271a7a723a5de69f8e237ab8e7803183131bc57c890db0b6bb872/uvloop-0.22.1-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:badb4d8e58ee08dad957002027830d5c3b06aea446a6a3744483c2b3b745345c", size = 4647548, upload-time = "2025-10-16T22:16:57.008Z" },
    { url = "https://files.pythonhosted.org/packages/c1/37/945b4ca0ac27e3dc4952642d4c900edd030b3da6c9634875af6e13ae80e5/uvloop-0.22.1-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:b91328c72635f6f9e0282e4a57da7470c7350ab1c9f48546c0f2866205349d21", size = 4467065, upload-time = "2025-10-16T22:16:58.206Z" },
    { url = "https://files.pythonhosted.org/packages/97/cc/48d232f33d60e2e2e0b42f4e73455b146b76ebe216487e862700457fbf3c/uvloop-0.22.1-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:daf620c2995d193449393d6c62131b3fbd40a63bf7b307a1527856ace637fe88", size = 4328384, upload-time = "2025-10-16T22:16:59.36Z" },
    { url = "https://files.pythonhosted.org/packages/e4/16/c1fd27e9549f3c4baf1dc9c20c456cd2f822dbf8de9f463824b0c0357e06/uvloop-0.22.1-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:6cde23eeda1a25c75b2e07d39970f3374105d5eafbaab2a4482be82f272d5a5e", size = 4296730, upload-time = "2025-10-16T22:17:00.744Z" },
]

[[package]]
name = "virtualenv"
version = "21.7.1"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "distlib" },
    { name = "filelock" },
    { name = "platformdirs" },
    { name = "python-discovery" },
]
sdist = { url = "https://files.pythonhosted.org/packages/ea/fa/18004e5cb15541ad2a68ff219c755233b012b12d4ec8663d06a258082bec/virtualenv-21.7.1.tar.gz", hash = "sha256:d0dbfaa5483487baea28d7210ef8d24c9d1bd0f10f449eeb215568825a9b334e", size = 5525237, upload-time = "2026-07-30T15:40:36.36Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/4b/a7/ded126c19495158a05c7202b3389139839d4cf78d622d453867778e0f7a8/virtualenv-21.7.1-py3-none-any.whl", hash = "sha256:6394973f990536e34c05157179146c020284c42fe01da1dfeb0ba16c345280d9", size = 5504576, upload-time = "2026-07-30T15:40:34.512Z" },
]

[[package]]
name = "watchfiles"
version = "1.2.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "anyio" },
]
sdist = { url = "https://files.pythonhosted.org/packages/cd/41/5e1a4bb12aac5f1493fa1bdc11154eca3b258ca4eba65d39c473fe19d8e9/watchfiles-1.2.0.tar.gz", hash = "sha256:c995fba777f1ea992f090f9236e9284cf7a5d1a0130dd5a3d82c598cacd76838", size = 108252, upload-time = "2026-05-18T04:32:04.251Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/b8/2f/e42c992d2afda3108ea1c02acecc991b9f31d05c14adc2a7cee9ee211fc4/watchfiles-1.2.0-cp312-cp312-macosx_10_12_x86_64.whl", hash = "sha256:bc13eb17538be00c874699dc0abe4ee2bc8d50bb1166a6b9e175ef3fd7eb8f26", size = 400115, upload-time = "2026-05-18T04:32:02.06Z" },
    { url = "https://files.pythonhosted.org/packages/5f/8f/6af2ea19065c91d8b0ea3516fdfc8c0d349f407e8e9fbf4e5a17360de8ad/watchfiles-1.2.0-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:2d95ddc1eb6914154253d239089900813f6a767e174b8e6a50e7fdacb7e4236c", size = 393659, upload-time = "2026-05-18T04:30:50.951Z" },
    { url = "https://files.pythonhosted.org/packages/13/01/b32a967c56fb3e3e5be3db52c3d3b87fa4513aa367d8ed1ad96d42952e5f/watchfiles-1.2.0-cp312-cp312-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:8f70d8b291ef6e88d19b1f297a6905ddb978888d9272b0d05e6f53309856bcfc", size = 453207, upload-time = "2026-05-18T04:31:04.231Z" },
    { url = "https://files.pythonhosted.org/packages/04/98/97557a812180338cb1abd32e1cffcc4588f59b5f23e0cb006b2ba95ba64a/watchfiles-1.2.0-cp312-cp312-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:56d8641cf834c2836922899105bd3ce3d0dfc69291d52edf0b4d0436829b34c0", size = 459273, uploaded-time = "2026-05-18T04:31:50.377Z" },
    { url = "https://files.pythonhosted.org/packages/e8/a8/b4b08dcb7653b8087c6586f7ce649505900e866bbcfe40dc9587af02e686/watchfiles-1.2.0-cp312-cp312-manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:2581a94056e55d7d0a31a823ea92bf73749c489ca2285bfdc0fbe6b2bb49d50c", size = 489927, upload-time = "2026-05-18T04:31:42.485Z" },
    { url = "https://files.pythonhosted.org/packages/50/94/3dceea03545d2e5ddfd839f0ddd5e1cecbf1697b5a428d5ba11cef6af95d/watchfiles-1.2.0-cp312-cp312-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:41bc1199f7523b3f82843c88cbb979180c949caef0342cf90968f178e5d49b01", size = 570476, upload-time = "2026-05-18T04:31:03.071Z" },
    { url = "https://files.pythonhosted.org/packages/cc/f2/d39a5450c3532092b91f81d274360e613c2371bc874a89c7a1a3c5e8d138/watchfiles-1.2.0-cp312-cp312-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:7571e4464cb6e434958f867f7f730b8ab0b75e3f8e5eac0499168486ab3c33a8", size = 465650, upload-time = "2026-05-18T04:30:12.701Z" },
    { url = "https://files.pythonhosted.org/packages/22/24/ed72f68cbc1333ca9b9f2200aa048bb6658ae41709bc1caad4310f4bdffd/watchfiles-1.2.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:e53a384f76b631c3ae5334ce6a52f0baa3a911eb94a4eac7f160079868b716d5", size = 456398, upload-time = "2026-05-18T04:30:13.784Z" },
    { url = "https://files.pythonhosted.org/packages/0d/64/982ef4a4e5bab5b6e5b6becc8cd5e732f6130a78b855f0abec6439a9a135/watchfiles-1.2.0-cp312-cp312-manylinux_2_31_riscv64.whl", hash = "sha256:d20029a60a71a052a24c4db7673bc4de39ab89adbaccbfb5d67987c5d73f424d", size = 465140, upload-time = "2026-05-18T04:31:52.111Z" },
    { url = "https://files.pythonhosted.org/packages/a0/0c/95282abf4ed680b6096010bcfc30c5fa7a041fc5aa5a2ad17a2cc6c75bba/watchfiles-1.2.0-cp312-cp312-musllinux_1_1_aarch64.whl", hash = "sha256:2cb93af48550faf1cea04c303107c8b75833de7013e57ce27d3b8d21d8d0f58c", size = 630259, upload-time = "2026-05-18T04:31:25.676Z" },
    { url = "https://files.pythonhosted.org/packages/30/45/607c1de1530c4bdcf2cf1d1ecc2505ddba5d96bd43ba9f2b0e79876f850f/watchfiles-1.2.0-cp312-cp312-musllinux_1_1_x86_64.whl", hash = "sha256:2995c176de7692b86a2e4c58d9ec718f753150a979cb4a754e2b4ffa38e70906", size = 659859, upload-time = "2026-05-18T04:30:24.333Z" },
    { url = "https://files.pythonhosted.org/packages/fa/08/d9e2e0f9e8e6791d33aefc694ad7eefa7f901f63caff84a81ded38692f9c/watchfiles-1.2.0-cp312-cp312-win32.whl", hash = "sha256:7a2cffd17d27d2ecbb310c2b1d8174f222a5495b1a721894afa88ec11e25b898", size = 275480, upload-time = "2026-05-18T04:30:31.307Z" },
    { url = "https://files.pythonhosted.org/packages/1c/e6/9d42569c0102645cc8cea5d8c7d8a1e9d4ada2cb7f05f75e554b8aa2202a/watchfiles-1.2.0-cp312-cp312-win_amd64.whl", hash = "sha256:f155b3a1b2a5fc89cdc70d47ee5d54e3b75e88efa34982028a35daef9ba00379", size = 288718, upload-time = "2026-05-18T04:32:10.745Z" },
    { url = "https://files.pythonhosted.org/packages/0a/26/88e0dc6ee3898169d7fa22bb6a69cabf2502d2ee25cb8c876d1262d204f8/watchfiles-1.2.0-cp312-cp312-win_arm64.whl", hash = "sha256:8fa585ede612ee9f9e91b18bebf9ba11b9ae29a4e3a0d0cf6fca3e382133f0d5", size = 281026, upload-time = "2026-05-18T04:30:22.23Z" },
    { url = "https://files.pythonhosted.org/packages/d1/4d/70a7feced9f87e2ff26dba42667290f41694fc64646c67261fbb8cab5d5c/watchfiles-1.2.0-cp313-cp313-macosx_10_12_x86_64.whl", hash = "sha256:01ea8d66f0693b9b60a6541c8d10263091ca9a9060d242f3c1f3143f9aad2c98", size = 399730, upload-time = "2026-05-18T04:31:38.162Z" },
    { url = "https://files.pythonhosted.org/packages/31/3a/0da302f2307aee316922806ebd5726c542cbd787c938271cf14a074c7daf/watchfiles-1.2.0-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:7ba0480b9a74af058f43b337e937a451e109295c420916d68ad24e3dc02f5e44", size = 392842, upload-time = "2026-05-18T04:30:27.051Z" },
    { url = "https://files.pythonhosted.org/packages/db/ef/d5bdb705c224dbc256aa0c1ec47bf4e61ec52558f2afb44a71a1fe4d7015/watchfiles-1.2.0-cp313-cp313-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:4f34e26a19f91f710c08e0183429f0d1d15df734e6bc78c31e77b9ea9c433658", size = 452989, upload-time = "2026-05-18T04:31:11.945Z" },
    { url = "https://files.pythonhosted.org/packages/71/29/5495f2c1661949ef7a35e4d71111d129cfe7606414a26887a919d0a55406/watchfiles-1.2.0-cp313-cp313-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:b4e77f6a55f858504069abd35d336a637555c09bca453dde1ee1e5ada8a6a1fb", size = 458978, uploaded-time = "2026-05-18T04:30:52.606Z" },
    { url = "https://files.pythonhosted.org/packages/d5/8c/7f9c07c433811c2fffd93e13fdfb7135de9aab5f2ae41be08960fa0047dc/watchfiles-1.2.0-cp313-cp313-manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:0cb4d80e212f116474a545c21c912b445f16bb0cef9e6a73a498164223e14e2f", size = 490248, upload-time = "2026-05-18T04:31:36.003Z" },
    { url = "https://files.pythonhosted.org/packages/3c/11/d93632febc52fbc21be90231bb7c17fd5387f46c9076fd40a5f9c2ae6910/watchfiles-1.2.0-cp313-cp313-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:b974946a10af379d425e2eef5b62f5c6ebeaccf91d45eaad6f5b27ecd4f91aa0", size = 571847, upload-time = "2026-05-18T04:31:10.862Z" },
    { url = "https://files.pythonhosted.org/packages/55/b4/383173e73aabb07ad1d9c7aa859d95437ac46a6d6a1e11005facda0c9d19/watchfiles-1.2.0-cp313-cp313-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:86bc13c25a8d1fcd70b51d0ce7c9b65e90de5666fcbfd3e34957cc73ee19aeb5", size = 465974, upload-time = "2026-05-18T04:30:17.006Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/a7/6c/89b1a230a78f57c52dd8893adb1f92f94411721b6ec12596c56d98c74356/watchfiles-1.2.0-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:ca148d73dea36c9763aaa351e4d7a51780ec1584217c45276f4fe8239c768b71", size = 454782, uploaded-time = "2026-05-18T04:30:35.656Z" },
    { url = "https://files.pythonhosted.org/packages/24/62/1732118367cfff0a9fce3bf62ff4bfded09ef5df21d9d446b858b3f70a96/watchfiles-1.2.0-cp313-cp313-manylinux_2_31_riscv64.whl", hash = "sha256:c525543d91961c6955b2636b308569e84a1d1c5f5f2932041ab9ef46422f43e3", size = 465182, upload-time = "2026-05-18T04:30:20.846Z" },
    { url = "https://files.pythonhosted.org/packages/28/96/716f7e5f51339bf22963f3345f9f27d7f3b30e2eadc597e257c881dd3c53/watchfiles-1.2.0-cp313-cp313-musllinux_1_1_aarch64.whl", hash = "sha256:a204794696ffb8f9b10fba6f7cb5216d42f3b2b71860ccac6b6e42f5f10973b0", size = 629841, upload-time = "2026-05-18T04:31:05.397Z" },
    { url = "https://files.pythonhosted.org/packages/4c/fe/c40783950fd771ccf66ab3ec2722d188a9af1c7f96c6e811f36e40c6e03f/watchfiles-1.2.0-cp313-cp313-musllinux_1_1_x86_64.whl", hash = "sha256:10d86db20695afe7997ac9e1717637d6714a8d0220458c33f3d2061f54cec427", size = 658028, upload-time = "2026-05-18T04:31:48.22Z" },
    { url = "https://files.pythonhosted.org/packages/71/72/4508db1856d1d87fcbb3b63f4839bab1b5682cb0e8d224d122263c09654a/watchfiles-1.2.0-cp313-cp313-win32.whl", hash = "sha256:eb283ee99e21ad6443c8cdb06ac5b34b1308c329cbdf03fa02b445363714c799", size = 275183, upload-time = "2026-05-18T04:30:59.57Z" },
    { url = "https://files.pythonhosted.org/packages/f9/36/14b76ca57652e5cc5fd1c11f32a261292c08a0d19a00351013c2549cbfb2/watchfiles-1.2.0-cp313-cp313-win_amd64.whl", hash = "sha256:a0f27f01bee51861392bb6b7c4fdb290b27d1eb194e9e28788d68102a0e898d9", size = 288059, upload-time = "2026-05-18T04:32:07.937Z" },
    { url = "https://files.pythonhosted.org/packages/1b/8d/0a85e395398d8d20fadfe5c5d32c726eee17a519e78fb356f2cf7531bffe/watchfiles-1.2.0-cp313-cp313-win_arm64.whl", hash = "sha256:3651aa7058595e9cfb75d35dd5ada2bf9f48a5b8a0f3562821d3e210c507e077", size = 280186, upload-time = "2026-05-18T04:31:54.484Z" },
    { url = "https://files.pythonhosted.org/packages/37/68/36db056f1fdcc5f07302f56e631774d6835bcd6fa3ace402304621d5f9e5/watchfiles-1.2.0-cp313-cp313t-macosx_10_12_x86_64.whl", hash = "sha256:faea288b6f0ab1902ef08f4ca6de005dccf856c4e0c4f21b8c5fce02d90a1b08", size = 399031, upload-time = "2026-05-18T04:30:44.576Z" },
    { url = "https://files.pythonhosted.org/packages/c1/64/01a9d6f66a82a5c101ce939274106cc72759d62427e153f01edd2b9f87c2/watchfiles-1.2.0-cp313-cp313t-macosx_11_0_arm64.whl", hash = "sha256:01859b11fd9fbca670f4d5da00fbac282cfea9bd67a2125d8b2833a3b5617ea9", size = 391205, upload-time = "2026-05-18T04:30:25.413Z" },
    { url = "https://files.pythonhosted.org/packages/84/2c/0a44fe058cb4bb7b8ede6b6670698bbb7c0400740e378d00022189b7b31d/watchfiles-1.2.0-cp313-cp313t-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:fff610d7bb2256a317bb1e96f0d7862c7aa8076733ee5df0fd41bbe76a24a4f4", size = 451892, upload-time = "2026-05-18T04:32:14.005Z" },
    { url = "https://files.pythonhosted.org/packages/67/a1/351e0d56cd35e6488b5c8b4fb11a809a5bc923e8fe8fed9faf8920be0c89/watchfiles-1.2.0-cp313-cp313t-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:b141a4891c995a039cd89e9a49e62df1dc8a559a5d1a6e4c7106d16c12777a55", size = 458867, upload-time = "2026-05-18T04:31:22.279Z" },
    { url = "https://files.pythonhosted.org/packages/d5/7d/9d09605187f1b838998624049fcf8bf47b73c1a3b76901fcac1782f62277/watchfiles-1.2.0-cp313-cp313t-manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:f22943b7770483f6ea0721c6b11d022947a98eb0acae14694de034f4d0d38925", size = 490217, upload-time = "2026-05-18T04:31:43.657Z" },
    { url = "https://files.pythonhosted.org/packages/60/5d/a17a16eccb182f04188cd308ec24b1a71a9b5c4e7098269cf35d9fa56d02/watchfiles-1.2.0-cp313-cp313t-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:1bc6195825b7dcd217968bb1f801a60fd4c16e8eeab5bedc7fe917d7d5995ab4", size = 571458, upload-time = "2026-05-18T04:32:11.875Z" },
    { url = "https://files.pythonhosted.org/packages/d3/3d/4dd457062083ab1938e5dfd45032eb425cee2ac817287ca8ff4356183e5d/watchfiles-1.2.0-cp313-cp313t-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:d4a4b147f5dca2a5d325a06a832fb43f345751adfbc63204aec30e0d9ca965a2", size = 464707, upload-time = "2026-05-18T04:30:43.492Z" },
    { url = "https://files.pythonhosted.org/packages/c6/71/ea8c57b128f5383de74d0c7d2d9c57ad7c9a65a930c451bd25d524b295b7/watchfiles-1.2.0-cp313-cp313t-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:4543579a9bdb0c9560039b4ffddbdb39545707659fbc430ce4c10f3f68d557f9", size = 454663, upload-time = "2026-05-18T04:30:16.061Z" },
    { url = "https://files.pythonhosted.org/packages/53/fd/2e812bf938406d7db351f0703ddd3fc6c061cf30d96153a77bc79a943a44/watchfiles-1.2.0-cp313-cp313t-manylinux_2_31_riscv64.whl", hash = "sha256:20aa0e708b920bde876a4aa82dc7dd6ebea228a63a67cda6632c2fc87b787efa", size = 463537, upload-time = "2026-05-18T04:31:44.9Z" },
    { url = "https://files.pythonhosted.org/packages/86/56/d17a7f1dd1bc3035f1072694a551301272f1739c2d8e319c927cb9e29b38/watchfiles-1.2.0-cp313-cp313t-musllinux_1_1_aarch64.whl", hash = "sha256:d413349d565dab74297f2a63e84a097936be69bf8f3b3801f27f380e32040f44", size = 629194, upload-time = "2026-05-18T04:31:14.141Z" },
    { url = "https://files.pythonhosted.org/packages/be/06/f1ff66bf5cae50aa4062779a0ecd0bbaf15e46619571907407894 7d9a17d/watchfiles-1.2.0-cp313-cp313t-musllinux_1_1_x86_64.whl", hash = "sha256:f28b2725eb8cce327b9b3ab02415c853011dc55c95832fe90de6bc56f5315f72", size = 656194, upload-time = "2026-05-18T04:31:47.14Z" },
    { url = "https://files.pythonhosted.org/packages/e7/54/a9c7ea9a82a4ac65e7004c0a03920b5cdd2f9c3b678757d9cd425aa51d53/watchfiles-1.2.0-cp314-cp314-macosx_10_12_x86_64.whl", hash = "sha256:b8c8358484d5fa12ef34f05b7f4168eaf1932f408725ff6d023c33ec17bd79d4", size = 400205, upload-time = "2026-05-18T04:32:05.153Z" },
    { url = "https://files.pythonhosted.org/packages/aa/5d/c9ab3534374a4a67450696905d6ef16a04405448b8dc52bd752ae50423d4/watchfiles-1.2.0-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:9f04b092229ad2c50126dd3c922c8822e51e605993764a33058d4a791ab42281", size = 392508, upload-time = "2026-05-18T04:30:54.849Z" },
    { url = "https://files.pythonhosted.org/packages/26/ca/1ad30103535cf0cecd7b993e8d50edc5351b1820e38f2d22e3df58962feb/watchfiles-1.2.0-cp314-cp314-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:7a7ce236284f002a156f70add88efe5c70879cccbb658be0822c54b1306fc09d", size = 452448, upload-time = "2026-05-18T04:30:53.727Z" },
    { url = "https://files.pythonhosted.org/packages/37/a1/ceee2cdf2afbd715fa07758d39c9859513eae411b23196f7fd039e5feedd/watchfiles-1.2.0-cp314-cp314-manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:b9909cc2b48468b575eefa944919e1fe8a36c5849d5c7c168f80a8c1db69398e", size = 459605, upload-time = "2026-05-18T04:30:23.312Z" },
    { url = "https://files.pythonhosted.org/packages/e8/f6/421e30fd1cb3907a84ed92ab3f1983e37ba2dca015e9a894a048418417a2/watchfiles-1.2.0-cp314-cp314-manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:0a37faaed405c67e28e6be45a1fa4f206ef5a2860f27c237db9fa30704c38242", size = 490757, upload-time = "2026-05-18T04:30:47.358Z" },
    { url = "https://files.pythonhosted.org/packages/41/b0/55ed1b97ed08be7bba6f9a541cac15f2a858e1d74d2b07b6da70a82aab00/watchfiles-1.2.0-cp314-cp314-manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:9649193aa27bd9ff2e80ff29bfaa93085496c7a3a377592823cc58b77ee88add", size = 568672, upload-time = "2026-05-18T04:30:38.915Z" },
    { url = "https://files.pythonhosted.org/packages/d1/cf/d8ae8a80dd7bafab395ea7681c10237311bbf34d37704a8c744e7cf31fc7/watchfiles-1.2.0-cp314-cp314-manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:4e4ff8e37f99cf1da89e255e07c9c4b37c214038c4283707bdec308cb1b0ea1f", size = 464197, upload-time = "2026-05-18T04:30:09.914Z" },
    { url = "https://files.pythonhosted.org/packages/7c/8a/3076c496ca8dafe0e8cd03fcebdfc47be4b1174b4e5b24ff6e396e6b3af2/watchfiles-1.2.0-cp314-cp314-manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:054dc20fd2e3132b4c3883b4a00d72fd6e1f56fdaf89fccd12e8057d74cd74d7", size = 453181, upload-time = "2026-05-18T04:30:14.829Z" },
    { url = "https://files.pythonhosted.org/packages/e5/10/9745e17c98e7b8a86454df0a3c7b5686bd650383f1e9f26e4ebcbd6cc0c0/watchfiles-1.2.0-cp314-cp314-manylinux_2_31_riscv64.whl", hash = "sha256:e140ed30ebde76796b686e67c182cff10ea2fbab186fafd1560f74bb5a473a6e", size = 465109, upload-time = "2026-05-18T04:30:28.123Z" },
    { url = "https://files.pythonhosted.org/packages/8f/95/8ef4a95481d3e0cb52d62a06fa6e972e81424be2d9698b91a2fecca9904c/watchfiles-1.2.0-cp314-cp314-musllinux_1_1_aarch64.whl", hash = "sha256:bb7e52ecf68ba46d22df23467b87cffeb2146908aa523ebfe803019618cfda06", size = 630653, upload-time = "2026-05-18T04:31:49.304Z" },
    { url = "https://files.pythonhosted.org/packages/fd/e4/3b3bf36b0f829b50c6ebcb8d031583863c59f923d6a6af3d485e470d0fac/watchfiles-1.2.0-cp314-cp314-musllinux_1_1_x86_64.whl", hash = "sha256:23282a321c8baf9b3a3c4afff673f9fe65eb7fdc2338d765ccad9d3d1916a5ba", size = 657838, upload-time = "2026-05-18T04:31:06.497Z" },
    { url = "https://files.pythonhosted.org/packages/21/b1/6cbbb50c1f3002ab568777d44aa21206dfb8807a840990c4037523b51812/watchfiles-1.2.0-cp314-cp314-win32.whl", hash = "sha256:c0db965c5f79aa49fe672d297cf1febc5ad149b658594944f49a54a2b96270a7", size = 275108, upload-time = "2026-05-18T04:30:06.891Z" },
    { url = "https://files.pythonhosted.org/packages/92/45/190ce6db8dcb4536682cf75d3889ff1a27182a58cb519d343cb6d9ea63d8/watchfiles-1.2.0-cp314-cp314-win_amd64.whl", hash = "sha256:71283b39fd17e5408eb123bd37aeecfd9d54c81fc184421943208aadb879d103", size = 288441, upload-time = "2026-05-18T04:32:12.901Z" },
    { url = "https://files.pythonhosted.org/packages/74/0d/3eae1c2313ab08378431d907c3f8095ecca00f3eda33111cf4f0f2591799/watchfiles-1.2.0-cp314-cp314-win_arm64.whl", hash = "sha256:c5c19526f4e54a00f2666a6c0e9e40d582c09e865055ea7378bf0009aab857b3", size = 280684, upload-time = "2026-05-18T04:31:26.902Z" },
    { url = "https://files.pythonhosted.org/packages/b1/75/fb64e6c25d6b5ca636d03df34ffb1c6e9873303e76d27967e045f8df088f/watchfiles-1.2.0-cp314-cp314t-macosx_10_12_x86_64.whl", hash = "sha256:d73a585accffa5ae39c17264c36ec3166d2fad7000c780f5ef83b2722afb9dd2", size = 398857, upload-time = "2026-05-18T04:32:17.108Z" },
    { url = "https://files.pythonhosted.org/packages/73/4e/9f7adf01754cbf81843722ccfec169d8f26c69778281a302855cecd2ee08/watchfiles-1.2.0-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:ae99b14c5f21e026e0e9d96f40e07d8570ebee6cafd9d8fc318354606daa7a28", size = 392413, upload-time = "2026-05-18T04:31:07.911Z" },
    { url = "https://files.pythonhosted.org/packages/47/c8/bec626bcc2d69f44b9acb24ce7d60ed7b16b73628eea747fcbd169d8edda/watchfiles-1.2.0-cp314-cp314t-manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:4429f3b105524a10b72c3a819b091c495d2811d419c1e1e8df773a5a5974f831", size = 452409, upload-time = "2026-05-18T04:31:20.142Z" },
    { url = "https://files.pythonhosted.org/packages/00/b7/b6362068e81e7c556d155a34c35d40ac3ef42d747b06d7f6e5bf58e359c2/watchfiles-1.2.0-cp314-cp314t-
```

```
manylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:43d818978d06062d9b22c4fab2ebe44cf5213d42dc8e62bda8c2760cfa2eeb33", size = 458827, uplo
ad-time = "2026-05-18T04:32:06.219Z" },
    { url = "https://files.pythonhosted.org/packages/67/f8/9a813fa42afb1e0b4625e75f0479826644d3ee8dc287e093799bc01f390c/watchfiles-1.2.0-cp314-cp314t-
manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:b9f732dc58b2dbe69e464ccf8fff7a03b0dd0be439da4c0720d3558527d3d6b4", size = 490104, upload-t
ime = "2026-05-18T04:31:56.034Z" },
    { url = "https://files.pythonhosted.org/packages/2f/bf/27dfb6094ca4c9aad21298b5525b6c53cb36121ee454331d05161e58d130/watchfiles-1.2.0-cp314-cp314t-
manylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:8f200104103feb097de4cab8fe4f5dd18a2026934c7dea98c55a2f5fd6d5a33b", size = 571360, up
load-time = "2026-05-18T04:31:57.133Z" },
    { url = "https://files.pythonhosted.org/packages/fb/39/44a096d67270ea93df91d33877dbe91fbda3aa4f8ec2edf799d93eda8736/watchfiles-1.2.0-cp314-cp314t-
manylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:63ac26eefbf4af1741247d6fb68b11c49a25b2f7413fbd318a83a12aaa9cf666", size = 464644, upload
-time = "2026-05-18T04:30:57.33Z" },
    { url = "https://files.pythonhosted.org/packages/0e/80/c7472203bad6268e3ef1ad26073970484789893 8ad7ea8b63a5131f46b50/watchfiles-1.2.0-cp314-cp314t-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:0c4997d4e4a55f0d02b6cde327322daf3a0400e5df6c6b15948994bf72497925", size = 454771, uplo
ad-time = "2026-05-18T04:30:48.736Z" },
    { url = "https://files.pythonhosted.org/packages/51/cf/3b10b268b4b7f0fc26e9debb5eef1998b515887840f444cd3ec80c688755/watchfiles-1.2.0-cp314-cp314t-
manylinux_2_31_riscv64.whl", hash = "sha256:4c887eba18b7945ac73067a8b4a66f21cd46c2539b2bc68588f7be6c7eb6d26b", size = 463494, upload-time = "2026-05-1
8T04:31:33.826Z" },
    { url = "https://files.pythonhosted.org/packages/3d/3e/a4302545cd589262a0dc7d140e86f7688eba3f9c72776c27f7e23b8864c4/watchfiles-1.2.0-cp314-cp314t-
musllinux_1_1_aarch64.whl", hash = "sha256:3416ff151bb6b5a8d8d11664974fbef4d9305b9b2957839ab5a270468fd8df30", size = 629383, upload-time = "2026-05-18
T04:31:15.596Z" },
    { url = "https://files.pythonhosted.org/packages/db/99/d5649df0a9a410d45b7c882304d0b790903ac9b6e8f2cfd12114e0c6b9f2/watchfiles-1.2.0-cp314-cp314t-
musllinux_1_1_x86_64.whl", hash = "sha256:0e831a271c035d89789cffc386b6aa1375f39f1cd25eb7ca0997e4970d152fc5", size = 656093, upload-time = "2026-05-18T
04:31:58.707Z" },
    { url = "https://files.pythonhosted.org/packages/92/b9/362702539275019a54dd2e94511b31a9b89c5f9e6a21966de7eb692549fc/watchfiles-1.2.0-cp315-cp315-m
acosx_10_12_x86_64.whl", hash = "sha256:37a6721cdf3f65dbb13aa9503510ccb4451603ac837e44d265d7992a597e1374", size = 400109, upload-time = "2026-05-18T04
:31:16.879Z" },
    { url = "https://files.pythonhosted.org/packages/8f/75/71d5ba62db781e5587bded1d944c675374bc4aa37ff33d5018d98e8b6538/watchfiles-1.2.0-cp315-cp315-m
acosx_11_0_arm64.whl", hash = "sha256:2b37d10b5a63bd4d87e18472d80fa525bd670586fae62e5dd580452764879b65", size = 392167, upload-time = "2026-05-18T04:3
1:28.058Z" },
    { url = "https://files.pythonhosted.org/packages/3c/01/c66dd95d0423fe30d31820e2d1d5bda773764131bbb6ac0cb1cf303ac328/watchfiles-1.2.0-cp315-cp315-m
anylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:0a105bc2283f67e8fbec74253ec2d94925de92ed72c0393f1206bf326b7b7b69", size = 452372, upl
oad-time = "2026-05-18T04:31:00.836Z" },
    { url = "https://files.pythonhosted.org/packages/91/15/2fe99557e72f85627c6a8eed50d889e8d101623e060a22ad75b875cb932d/watchfiles-1.2.0-cp315-cp315-m
anylinux_2_17_armv7l.manylinux2014_armv7l.whl", hash = "sha256:5327989a465505f05cfe06f04fa9d0c2fd5432bb243e10e6f012b1bdca3c8579", size = 459596, uploa
d-time = "2026-05-18T04:31:34.96Z" },
    { url = "https://files.pythonhosted.org/packages/ed/23/d4acfa0023367428ed48351b3b9b267893037b6cadae55620c61c24bcfd4/watchfiles-1.2.0-cp315-cp315-m
anylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:ecb47f183a8025b2aa18b546725c3657e542112ae9c0613a2af79b4fa8d04ad7", size = 490869, upload-ti
me = "2026-05-18T04:31:59.923Z" },
    { url = "https://files.pythonhosted.org/packages/a4/5f/3164cbdce06c9fb95c4f7b9e2f9760b5e2797af43a9ecc317ef42a23a278/watchfiles-1.2.0-cp315-cp315-m
anylinux_2_17_ppc64le.manylinux2014_ppc64le.whl", hash = "sha256:8520a4ab0e37f770afc34459c4f8f7019e153f9124dc101c15538365875d1ab2", size = 571641, upl
oad-time = "2026-05-18T04:32:00.948Z" },
    { url = "https://files.pythonhosted.org/packages/41/e6/85d3731c55e65cd7690f3f803d24c139588aaf863e4bf2148fe7a7fa1a19/watchfiles-1.2.0-cp315-cp315-m
anylinux_2_17_s390x.manylinux2014_s390x.whl", hash = "sha256:71cd71740ed2c15211ebb237ced4e39a1cdf6f80566e5fe95428da1626f4fde6", size = 464444, upload-
time = "2026-05-18T04:30:34.298Z" },
    { url = "https://files.pythonhosted.org/packages/f4/7d/562641012b8b09872742c3b8adf9629ec479fd78f8d68ae4a0c13da8add6/watchfiles-1.2.0-cp315-cp315-m
anylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:f88af53d6ddaf72179ef613ddc905e6f4785f712b49b80b3bef9f3525e6194b4", size = 453593, uploa
d-time = "2026-05-18T04:31:23.464Z" },
    { url = "https://files.pythonhosted.org/packages/56/fe/cb8ef3d6f929d14158fdaaad9925985b7310abc9384dcd4d82dd0016fb59/watchfiles-1.2.0-cp315-cp315-m
anylinux_2_31_riscv64.whl", hash = "sha256:cee9d5efd929efdac5f7e58f72b3376f676b64050a91c5b99a7094c5b2317488", size = 465096, upload-time = "2026-05-18
T04:31:30.384Z" },
    { url = "https://files.pythonhosted.org/packages/25/91/80908e835e100527a9267147b08c0eee1fa6ab0ffec15edc04d1d44885f7/watchfiles-1.2.0-cp315-cp315-m
usllinux_1_1_aarch64.whl", hash = "sha256:b718bf356bbc15e559bd8ef41782b573b8ae0e3f177ab244b440568d7ea02cfb", size = 630638, upload-time = "2026-05-18T
04:30:49.89Z" },
    { url = "https://files.pythonhosted.org/packages/46/4b/95ab2f256bb4af3cb2eb23b9317bda984ee6e0f11733a5c004a6c95b06e3/watchfiles-1.2.0-cp315-cp315-m
usllinux_1_1_x86_64.whl", hash = "sha256:922c0e019fe68b3ae392965a766b02a71ba1168c932cebc3733cd52c5fe5b377", size = 657684, upload-time = "2026-05-18T0
4:31:32.027Z" },
]

[[package]]
name = "websockets"
version = "15.0.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/21/e6/26d09fab466b7ca9c7737474c52be4f76a40301b08362eb2dbc19dcc16c1/websockets-15.0.1.tar.gz",
 hash = "sha256:82544de02076bafba038ce055ee6412d68da13ab47f0c60cab827346de828dee", size = 177016, upload-time = "2025-03-05T20:03:41.606Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/51/6b/4545a0d843594f5d0771e86463606a3988b5a09ca5123136f8a76580dd63/websockets-15.0.1-cp312-cp312-
macosx_10_13_universal2.whl", hash = "sha256:3e90baa811a5d73f3ca0bcbf32064d663ed81318ab225ee4f427ad4e26e5aff3", size = 175437, upload-time = "2025-03-
05T20:02:16.706Z" },
    { url = "https://files.pythonhosted.org/packages/f4/71/809a0f5f6a06522af902e0f2ea2757f71ead94610010cf570ab5c98e99ed/websockets-15.0.1-cp312-cp312-
macosx_10_13_x86_64.whl", hash = "sha256:592f1a9fe869c778694f0aa806ba0374e97648ab57936f092fd9d87f8bc03665", size = 173096, upload-time = "2025-03-05T2
0:02:18.832Z" },
    { url = "https://files.pythonhosted.org/packages/3d/69/1a681dd6f02180916f116894181eab8b2e25b31e484c5d0eae637ec01f7c/websockets-15.0.1-cp312-cp312-
macosx_11_0_arm64.whl", hash = "sha256:0701bc3cfcb9164d04a14b149fd74be7347a530ad3bbf15ab2c678a2cd3dd9a2", size = 173332, upload-time = "2025-03-05T20:
02:20.187Z" },
    { url = "https://files.pythonhosted.org/packages/a6/02/0073b3952f5bce97eafbb35757f8d0d54812b6174ed8dd952aa08429bcc3/websockets-15.0.1-cp312-cp312-
manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:e8b56bdcdb4505c8078cb6c7157d9811a85790f2f2b3632c7d1462ab5783d215", size = 183152, up
load-time = "2025-03-05T20:02:22.286Z" },
    { url = "https://files.pythonhosted.org/packages/74/45/c205c8480eafd114b428284840da0b1be9ffd0e4f87338dc95dc6ff961a1/websockets-15.0.1-cp312-cp312-
manylinux_2_5_i686.manylinux1_i686.manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:0af68c55afbd5f07986df82831c7bff04846928ea8d1fd7f3005263
8788bc9b5", size = 182096, upload-time = "2025-03-05T20:02:24.368Z" },
    { url = "https://files.pythonhosted.org/packages/14/8f/aa61f528fba38578ec553c145857a181384c72b98156f858ca5c8e82d9d3/websockets-15.0.1-cp312-cp312-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:64dee438fed052b52e4f98f76c5790513235efaa1ef7f3f
2192c392cd7c91b65", size = 182523, upload-time = "2025-03-05T20:02:25.669Z" },
    { url = "https://files.pythonhosted.org/packages/ec/6d/0267396610add5bc0d0d3e77f546d4cd287200804fe02323797de77dbce9/websockets-15.0.1-cp312-cp312-
musllinux_1_2_aarch64.whl", hash = "sha256:d5f6b181bb38171a8ad1d6aa58a67a6aa9d4b38d0f8c5f496b9e42561dfc62fe", size = 182790, upload-time = "2025-03-05
T20:02:26.99Z" },
    { url = "https://files.pythonhosted.org/packages/02/05/c68c5adbf679cf610ae2f74a9b871ae84564462955d991178f95a1ddb7dd/websockets-15.0.1-cp312-cp312-
musllinux_1_2_i686.whl", hash = "sha256:5d54b09eba2bada6011aea5375542a157637b91029687eb4fdb2dab11059c1b4", size = 182165, upload-time = "2025-03-05T20
:02:30.291Z" },
    { url = "https://files.pythonhosted.org/packages/29/93/bb672df7b2f5faac89761cb5fa34f5cec45a4026c383a4b5761c6cea5c16/websockets-15.0.1-cp312-cp312-
musllinux_1_2_x86_64.whl", hash = "sha256:3be571a8b5afed347da347bfcf27ba12b069d9d7f42cb8c7028b5e98bbb12597", size = 182160, upload-time = "2025-03-05T
20:02:31.634Z" },
    { url = "https://files.pythonhosted.org/packages/ff/83/de1f7709376dc3ca9b7eeb4b9a07b4526b14876b6d372a4dc62312bebee0/websockets-15.0.1-cp312-cp312-
win32.whl", hash = "sha256:c338ffa0520bdb12fbc527265235639fb76e7bc7faafbb93f6ba80d9c06578a9", size = 176395, upload-time = "2025-03-05T20:02:33.017Z"
},
    { url = "https://files.pythonhosted.org/packages/7d/71/abf2ebc3bbfa40f391ce1428c7168fb20582d0ff57019b69ea20fa698043/websockets-15.0.1-cp312-cp312-
win_amd64.whl", hash = "sha256:fcd5cf9e305d7b8338754470cf69cf81f420459dbae8a3b40cee57417f4614a7", size = 176841, upload-time = "2025-03-05T20:02:34.49
8Z" },
    { url = "https://files.pythonhosted.org/packages/cb/9f/51f0cf64471a9d2b4d0fc6c534f323b664e7095640c34562f5182e5a7195/websockets-15.0.1-cp313-cp313-
macosx_10_13_universal2.whl", hash = "sha256:ee443ef070bb3b6ed74514f5efaa37a252af57c90eb33b956d35c8e9c10a1931", size = 175440, upload-time = "2025-03-
05T20:02:36.695Z" },
    { url = "https://files.pythonhosted.org/packages/8a/05/aa116ec9943c718905997412c5989f7ed671bc0188ee2ba89520e8765d7b/websockets-15.0.1-cp313-cp313-
macosx_10_13_x86_64.whl", hash = "sha256:5a939de6b7b4e18ca683218320fc67ea886038265fd1ed30173f5ce3f8e85675", size = 173098, upload-time = "2025-03-05T2
0:02:37.985Z" },
    { url = "https://files.pythonhosted.org/packages/ff/0b/33cef55ff24f2d92924923c99926dcce78e7bd922d649467f0eda8368923/websockets-15.0.1-cp313-cp313-
```

```
macosx_11_0_arm64.whl", hash = "sha256:746ee8dba912cd6fc889a8147168991d50ed70447bf18bcda7039f7d2e3d9151", size = 173329, upload-time = "2025-03-05T20:
02:39.298Z" },
    { url = "https://files.pythonhosted.org/packages/31/1d/063b25dcc01faa8fada1469bdf769de3768b7044eac9d41f734fd7b6ad6d/websockets-15.0.1-cp313-cp313-
manylinux_2_17_aarch64.manylinux2014_aarch64.whl", hash = "sha256:595b6c3969023ecf9041b2936ac3827e4623bfa3ccf007575f04c5a6aa318c22", size = 183111, up
load-time = "2025-03-05T20:02:40.595Z" },
    { url = "https://files.pythonhosted.org/packages/93/53/9a87ee494a51bf63e4ec9241c1ccc4f7c2f45fff85d5bde2ff74fcb68b9e/websockets-15.0.1-cp313-cp313-
manylinux_2_5_i686.manylinux1_i686.manylinux_2_17_i686.manylinux2014_i686.whl", hash = "sha256:3c714d2fc58b5ca3e285461a4cc0c9a66bd0e24c5da9911e3015828
6c9b5be7f", size = 182054, upload-time = "2025-03-05T20:02:41.926Z" },
    { url = "https://files.pythonhosted.org/packages/ff/b2/83a6ddf56cdcbad4e3d841fcc55d6ba7d19aeb89c50f24dd7e859ec0805f/websockets-15.0.1-cp313-cp313-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl", hash = "sha256:0f3c1e2ab208db911594ae5b4f79addeb3501604a165019
dd221c0bdcabe4db8", size = 182496, upload-time = "2025-03-05T20:02:43.304Z" },
    { url = "https://files.pythonhosted.org/packages/98/41/e7038944ed0abf34c45aa4635ba28136f06052e08fc2168520bb8b25149f/websockets-15.0.1-cp313-cp313-
musllinux_1_2_aarch64.whl", hash = "sha256:229cf1d3ca6c1804400b0a9790dc66528e08a6a1feec0d5040e8b9eb14422375", size = 182829, upload-time = "2025-03-05
T20:02:48.812Z" },
    { url = "https://files.pythonhosted.org/packages/e0/17/de15b6158680c7623c6ef0db361da965ab25d813ae54fcfeae2e5b9ef910/websockets-15.0.1-cp313-cp313-
musllinux_1_2_i686.whl", hash = "sha256:756c56e867a90fb00177d530dca4b097dd753cde348448a1012ed6c5131f8b7d", size = 182217, upload-time = "2025-03-05T20
:02:50.14Z" },
    { url = "https://files.pythonhosted.org/packages/33/2b/1f168cb6041853eef0362fb9554c3824367c5560cbdaad89ac40f8c2edfc/websockets-15.0.1-cp313-cp313-
musllinux_1_2_x86_64.whl", hash = "sha256:558d023b3df0bffe50a04e710bc87742de35060580a293c2a984299ed83bc4e4", size = 182195, upload-time = "2025-03-05T
20:02:51.561Z" },
    { url = "https://files.pythonhosted.org/packages/86/eb/20b6cdf273913d0ad05a6a14aed4b9a85591c18a987a3d47f20fa13dcc47/websockets-15.0.1-cp313-cp313-
win32.whl", hash = "sha256:ba9e56e8ceeeedb2e080147ba85ffcd5cd0711b89576b83784d8605a7df455fa", size = 176393, upload-time = "2025-03-05T20:02:53.814Z"
},
    { url = "https://files.pythonhosted.org/packages/1b/6c/c65773d6cab416a64d191d6ee8a8b1c68a09970ea6909d16965d26bfed1e/websockets-15.0.1-cp313-cp313-
win_amd64.whl", hash = "sha256:e09473f095a819042ecb2ab9465aee615bd9c2028e4ef7d933600a8401c79561", size = 176837, upload-time = "2025-03-05T20:02:55.23
7Z" },
    { url = "https://files.pythonhosted.org/packages/fa/a8/5b41e0da817d64113292ab1f8247140aac61cbf6cfd085d6a0fa77f4984f/websockets-15.0.1-py3-none-any
.whl", hash = "sha256:f7a866fbc1e97b5c617ee4116daaa09b722101d4a3c170c787450ba409f9736f", size = 169743, upload-time = "2025-03-05T20:03:39.41Z" },
]

[[package]]
name = "win32-setctime"
version = "1.2.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/b3/8f/705086c9d734d3b663af0e9bb3d4de6578d08f46b1b101c2442fd9aecaa2/win32_setctime-1.2.0.tar.g
z", hash = "sha256:ae1fdf948f5640aae05c511ade119313fb6a30d7eabe25fef9764dca5873c4c0", size = 4867, upload-time = "2024-12-07T15:28:28.314Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/e1/07/c6fe3ad3e685340704d314d765b7912993bcb8dc198f0e7a89382d37974b/win32_setctime-1.2.0-py3-none-
any.whl", hash = "sha256:95d644c4e708aba81dc3704a116d8cbc974d70b3bdb8be1d150e36be6e9d1390", size = 4083, upload-time = "2024-12-07T15:28:26.465Z" },
]

[[package]]
name = "wrapt"
version = "2.3.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/2b/b0/c1f5a970721f06b85c0cd5142e0ff8fe067708abd779b0c4f4be7d61d09f/wrapt-2.3.0.tar.gz", hash
= "sha256:681a2d0eefd721998f90642762b8e75c2159ec531b20ad5e437245ea7b06a107", size = 131509, upload-time = "2026-07-28T06:06:14.895Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/5b/4a/d17a0fad1bf1c5f2c887ff71fef75654141b0880bff71d157d955b5bec3a/wrapt-2.3.0-cp312-cp312-macosx
_10_13_x86_64.whl", hash = "sha256:0a45ffae742ce91a16e11cb6c7cd71e7f9994f3cbd283b962ab093f5c6dcf525", size = 82139, upload-time = "2026-07-28T06:04:35
.082Z" },
    { url = "https://files.pythonhosted.org/packages/6e/55/51b92daaf6defb57f4dc56bdcce985400f75c6984a03ca5e78ccac717028/wrapt-2.3.0-cp312-cp312-macosx
_11_0_arm64.whl", hash = "sha256:69e477046f2237ef0bc6547544ee73008dc764ca26eff44f09e976d221b34d5d", size = 82723, upload-time = "2026-07-28T06:04:36.5
02Z" },
    { url = "https://files.pythonhosted.org/packages/28/7f/cfd9bc4b1f5e424eeea83d0493e43f3b1b02707ce8e50c47945873982bd5/wrapt-2.3.0-cp312-cp312-manyli
nux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:5d221a6e6ddd302b8397433184e96b59f259f50024b854db1c411a881586b6b8", size =
172381, upload-time = "2026-07-28T06:04:37.674Z" },
    { url = "https://files.pythonhosted.org/packages/cb/89/ff7814f6eb6856b479946117d1138a2fbb46cdb6b1f379db359056c69743/wrapt-2.3.0-cp312-cp312-manyli
nux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:392158c9a7f2ab1b8699418bfc0fe6f83548788c418b27d7bf2019ad3405cebb",
size = 174120, upload-time = "2026-07-28T06:04:38.987Z" },
    { url = "https://files.pythonhosted.org/packages/12/1e/8eded8615d39e3ce81f626937a3a87b280a2a86239a2bf14a4b4bb345034/wrapt-2.3.0-cp312-cp312-manyli
nux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:e5301c35cf75655eb33498f2bd6ae8703ca19940e3167dc9cdf740c712a39c60", size = 163035, upload-
time = "2026-07-28T06:04:40.361Z" },
    { url = "https://files.pythonhosted.org/packages/35/ea/a0af2d9da62897af2a055484920de05dade30d2ba2c0d65cbdea875d3d8b/wrapt-2.3.0-cp312-cp312-muslli
nux_1_2_aarch64.whl", hash = "sha256:418f54bb09d1762db02c7009b4051149893af3153a87f92d70356703c11eea02", size = 171887, upload-time = "2026-07-28T06:04
:41.614Z" },
    { url = "https://files.pythonhosted.org/packages/7e/dd/63cd4c864c65ef4906df64bd2d378f4a62b54f28063f282dfb3bf93caead/wrapt-2.3.0-cp312-cp312-muslli
nux_1_2_riscv64.whl", hash = "sha256:1598becd30f8f2777d18564064eb4f4dbe1ab0e05a8f09786d0ef505ac782bf3", size = 161113, upload-time = "2026-07-28T06:04
:42.864Z" },
    { url = "https://files.pythonhosted.org/packages/ca/ee/82f1fc9e431b5c2c5a6d201aa865dbeae3984c311c6d11a185f0c8367cf6/wrapt-2.3.0-cp312-cp312-muslli
nux_1_2_x86_64.whl", hash = "sha256:3da470536bf9645143323dd41b32db55c6f4304ad382094c1a1da8a92061e10d", size = 170530, upload-time = "2026-07-28T06:04:
44.212Z" },
    { url = "https://files.pythonhosted.org/packages/37/a5/5dc590e863a419930d988f8b7ca3e75a6befcfb10b6003b3a152f3d5f732/wrapt-2.3.0-cp312-cp312-win32.
whl", hash = "sha256:fb8e2e6704a1e0b1b989546c69e2688371ef4a07fa5f61bde3eb6211186f5ac1", size = 78323, upload-time = "2026-07-28T06:04:45.484Z" },
    { url = "https://files.pythonhosted.org/packages/51/f9/4a6925a07951df56394f7e6ebe14f69f1c5ef9d87aa63e0839acf15aa63a/wrapt-2.3.0-cp312-cp312-win_am
d64.whl", hash = "sha256:cdc021cb0b62471d6aac7f2bd92f3b4658073775f9ee7fcd325c511129e7bcc8", size = 81180, upload-time = "2026-07-28T06:04:47.021Z" },
    { url = "https://files.pythonhosted.org/packages/a8/4f/8b5de0395b2a72216751d41c9861df6facaeb611b619d8810ed2b3b23eb2/wrapt-2.3.0-cp312-cp312-win_ar
m64.whl", hash = "sha256:67bfe2485f50368c3fcd2275fc1fd100e350d601e0058921a7c82678a465aeab", size = 80155, upload-time = "2026-07-28T06:04:48.373Z" },
    { url = "https://files.pythonhosted.org/packages/8e/6e/0f88a072483e76b881e3fdcd6b6ffb4a5791002514fe541e72b1b73c859a/wrapt-2.3.0-cp313-cp313-macosx
_10_13_x86_64.whl", hash = "sha256:0d3fb71e65b001adfc42684522eeccd9c21d8ba679945abc993439567b66e59f", size = 81960, upload-time = "2026-07-28T06:04:49
.622Z" },
    { url = "https://files.pythonhosted.org/packages/d7/ff/b7e2776e7c294075eb712cc9ef573d1b818f393006d09787262b8fc871c4/wrapt-2.3.0-cp313-cp313-macosx
_11_0_arm64.whl", hash = "sha256:51a7a4181c1295774812271fbcd7c909df372bc25579d4ed9eb875caaf0ae86f", size = 82435, upload-time = "2026-07-28T06:04:50.9
Z" },
    { url = "https://files.pythonhosted.org/packages/d8/90/343bb5d0f1f9669bc252a6073f085b4abf862511bd5c9c9eaec754341f1d/wrapt-2.3.0-cp313-cp313-manyli
nux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:9045917809c63fdf7abe3a2ceaed3d670b8ee4500ddd9291192d30aeb34467c5", size =
170350, upload-time = "2026-07-28T06:04:52.187Z" },
    { url = "https://files.pythonhosted.org/packages/59/f8/13b79a392930bd0dd6b86cbfbfe1c40944110456e1dc6d809e5c46ece904/wrapt-2.3.0-cp313-cp313-manyli
nux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:54ca1d5573f69b5fe1d74f1f65799c68015e82f685efec9fd8cfa40a094c44d0",
size = 170022, upload-time = "2026-07-28T06:04:53.599Z" },
    { url = "https://files.pythonhosted.org/packages/b2/fc/4f1b6918f5290db959d6e0c07f77385d87cede29c39c9cf8f145e9c82954/wrapt-2.3.0-cp313-cp313-manyli
nux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:242b60c21e30866e6a2fa606c612b47c553fa60c0eaeeeb7797fb842ac0ce609", size = 161043, upload-
time = "2026-07-28T06:04:54.936Z" },
    { url = "https://files.pythonhosted.org/packages/01/e1/45d3cf74414780bdff6d0380467e003f6eb0f028b6c9403db868dbc7209c/wrapt-2.3.0-cp313-cp313-muslli
nux_1_2_aarch64.whl", hash = "sha256:e3f3d7ec0a51fbfe00d3aef047641ff2c58b25565b4717fc1f90e050be01cba8", size = 168576, upload-time = "2026-07-28T06:04
:56.261Z" },
    { url = "https://files.pythonhosted.org/packages/f3/73/2fa58dd97f191c997755e2c6d569a68f0c433db4e4b36099bdd7227b6cac/wrapt-2.3.0-cp313-cp313-muslli
nux_1_2_riscv64.whl", hash = "sha256:261f53870cd4fb2bf38f9f972c56c728fd224cb7c65721307de59d9e7e6741ae", size = 159140, upload-time = "2026-07-28T06:04
:57.754Z" },
    { url = "https://files.pythonhosted.org/packages/29/a8/08a56e2000a8816d449dcbad8c8b081697acbbd490821ceca0f9d8e8d20c/wrapt-2.3.0-cp313-cp313-muslli
nux_1_2_x86_64.whl", hash = "sha256:8159ec0b0cb7608175eb150de94c19e34f4d47ac655f5ca9baf45df6b688ffd3", size = 169263, upload-time = "2026-07-28T06:04:
59.161Z" },
    { url = "https://files.pythonhosted.org/packages/9e/d4/354e1725e35a73b2af4fa70a3e024c7a5d1bf1802dfb862dcb668aae0253/wrapt-2.3.0-cp313-cp313-win32.
whl", hash = "sha256:10461884b3014fbfc8eb7d09a93c5f246363e6711d9d881f95eb8c27fdef049f", size = 78241, upload-time = "2026-07-28T06:05:00.507Z" },
    { url = "https://files.pythonhosted.org/packages/6c/7e/34c87fa2174848dfee820322aaa318bab08913998ccecc8d2f57b4ad4639/wrapt-2.3.0-cp313-cp313-win_am
```

```
d64.whl", hash = "sha256:ac870cc97b73bb00ac353329e9559a4bebc47c4c86792ed9b23b58c15b6ad838", size = 81113, upload-time = "2026-07-28T06:05:01.839Z" },
    { url = "https://files.pythonhosted.org/packages/11/86/fcc9a530579e008c9478bb565a6cdfbfd33536660f069c8b91a6607c5050/wrapt-2.3.0-cp313-cp313-win_ar
m64.whl", hash = "sha256:a65e8db2b4e90c2e7ade93108351c98ef420bf7a94ee08c95ac8a3cbbc43579", size = 80182, upload-time = "2026-07-28T06:05:03.152Z" },
    { url = "https://files.pythonhosted.org/packages/96/50/3864848b95b28ef73e17551fc8dccbff2628a834f52cf26a57f9c419fb83/wrapt-2.3.0-cp313-cp313t-macos
x_10_13_x86_64.whl", hash = "sha256:fd1f2f557dd3491fe75905e578f4db967393d40d1a8f468edc4d40ac7f2d5944", size = 83921, upload-time = "2026-07-28T06:05:0
4.476Z" },
    { url = "https://files.pythonhosted.org/packages/3b/4c/3d1921a60c3e8c71c540ff136e6a47a1fbccf7f671e818394889f7871d9c/wrapt-2.3.0-cp313-cp313t-macos
x_11_0_arm64.whl", hash = "sha256:9f5d2aec29dfc76c37e23897dee92766a3fd4f3bff3ae7fc9c6b4bf37d8c1360", size = 84412, upload-time = "2026-07-28T06:05:05.
921Z" },
    { url = "https://files.pythonhosted.org/packages/fa/1a/4a796ff7adb26ada6d4b758c94d47a38320b085e7099afc088efbbcdb006/wrapt-2.3.0-cp313-cp313t-manyl
inux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:646d20d413ffcd1b0a2f700076e2d0252d872dcb7754860a73e45a59ea883614", size =
 207168, upload-time = "2026-07-28T06:05:07.256Z" },
    { url = "https://files.pythonhosted.org/packages/1d/3e/d7777776806c579b761bac2f91721dda9f04c7a1b380213c5935cc750ae6/wrapt-2.3.0-cp313-cp313t-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:379f670f45b7bb8993edd9f6fc36c6cc65edb81cffa0b504be34acb0303fff0a",
 size = 214351, upload-time = "2026-07-28T06:05:08.945Z" },
    { url = "https://files.pythonhosted.org/packages/63/27/2d64d394df7bf181955b3bb562bf33c4492fb4be113f53071106d43ad8b5/wrapt-2.3.0-cp313-cp313t-manyl
inux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:6208f302f110295d64b22a7ac96500c791bf492dce4366e622e4912b077c9687", size = 199020, upload
-time = "2026-07-28T06:05:10.418Z" },
    { url = "https://files.pythonhosted.org/packages/3e/3d/fb31d3db7d9834d265fb1a27a2adf0ddf51557c67458c97b22439ad6ae3d/wrapt-2.3.0-cp313-cp313t-musll
inux_1_2_aarch64.whl", hash = "sha256:ed635a9ca4f3a5a2b900c10c69e823373bc00ebc114b459383596d3487da3570", size = 209969, upload-time = "2026-07-28T06:0
5:11.983Z" },
    { url = "https://files.pythonhosted.org/packages/1f/d1/8724b5da582e62070dc9bf4d8bf1972f317297eefd7ba1f2b5c6393ccf6c/wrapt-2.3.0-cp313-cp313t-musll
inux_1_2_riscv64.whl", hash = "sha256:e3b9eaa742ae7a0aaaaad4ca4b69469d757af2d6e6663ef1dadc47adec0aeb41", size = 196324, upload-time = "2026-07-28T06:0
5:13.557Z" },
    { url = "https://files.pythonhosted.org/packages/0d/5c/3d9ef411149543016ee6bcf3af707f787cebd946527452b94bf122e9b7b4/wrapt-2.3.0-cp313-cp313t-musll
inux_1_2_x86_64.whl", hash = "sha256:d0f7284f88f4833705132d06d3b425a43095c2cbd07c58166aac3ab646ba12a4", size = 202610, upload-time = "2026-07-28T06:05
:15.048Z" },
    { url = "https://files.pythonhosted.org/packages/13/9b/4fc042ceb757866dd4a5fc057b3b736f2b360d3703ce9f830d83dc9226e0/wrapt-2.3.0-cp313-cp313t-win32
.whl", hash = "sha256:7ebb274aba688b043429eb1500ff8a76ce0cb8ac0812ca3e301f06247b8722b3", size = 79178, upload-time = "2026-07-28T06:05:16.469Z" },
    { url = "https://files.pythonhosted.org/packages/6b/ff/b94878f8eed809ca042685276bcea9f24e8c2ca7c9653bb80bbb920a68a5/wrapt-2.3.0-cp313-cp313t-win_a
md64.whl", hash = "sha256:c4bded758ad6f03b965830944a2f0bc5b2eb3767fe5a7310134315d1a6610e98", size = 82634, upload-time = "2026-07-28T06:05:18.026Z" },
    { url = "https://files.pythonhosted.org/packages/80/fb/663e1de5332a71685a729754312d327d4cada767c36e1c5a2db4c8de49e6/wrapt-2.3.0-cp313-cp313t-win_a
rm64.whl", hash = "sha256:d2cc64539da63e39ffb9c7ede849b6e8ddaaf7b3876b5cfb04efd85a5f3f4eb6", size = 81387, upload-time = "2026-07-28T06:05:19.417Z" },
    { url = "https://files.pythonhosted.org/packages/58/10/b073beaea89bc0d3670a75ff51139430a54b6af7ba7796507730634536dd/wrapt-2.3.0-cp314-cp314-macosx
_10_15_x86_64.whl", hash = "sha256:ea52a0d0f08c584943d5764be0e84efa912c8da23c23e1e285ff2f5641c18fcc", size = 81978, upload-time = "2026-07-28T06:05:21
.133Z" },
    { url = "https://files.pythonhosted.org/packages/b3/31/0916d9cebf848ed3f1a0c1888faee421747df77331e4db2bc527a9a85988/wrapt-2.3.0-cp314-cp314-macosx
_11_0_arm64.whl", hash = "sha256:fd85b0aa88efdb189d6ae2f35f4526943a8f091c38599c9c31478241c819e6a1", size = 82518, upload-time = "2026-07-28T06:05:22.5
62Z" },
    { url = "https://files.pythonhosted.org/packages/f5/73/31c1bf0f3384062751c2094dadb314916d70aa9b6bfd26d994b4a7b393fa/wrapt-2.3.0-cp314-cp314-manyli
nux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:141ed6211286a9660d8d6702de598b43f0934b4f0eda16393f100a80f501d945", size =
170187, upload-time = "2026-07-28T06:05:23.904Z" },
    { url = "https://files.pythonhosted.org/packages/ed/25/fce087d54b79b8905f3c3c9dd5f454bbd8d8acb80b960c4a6aee5b4659b3/wrapt-2.3.0-cp314-cp314-manyli
nux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:2e49885a62ec4ee854d1b9e6371fda6afd219917225752abf729a3f36d4df9a5",
size = 169288, upload-time = "2026-07-28T06:05:25.378Z" },
    { url = "https://files.pythonhosted.org/packages/c7/30/0d09e6dddc6b7a7230ac77f50254b5980ab4fcd22976f72f8cc8a0404458/wrapt-2.3.0-cp314-cp314-manyli
nux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:1d6159c9b2fefec02314e1332dbbbfaf960e369dfd26bcf7f8b258b5732065b3", size = 160932, upload-
time = "2026-07-28T06:05:27.022Z" },
    { url = "https://files.pythonhosted.org/packages/2c/ca/0913af0d2ec0c43865d32d615f518fea66c13c5c930e489e9b0de248e9a8/wrapt-2.3.0-cp314-cp314-muslli
nux_1_2_aarch64.whl", hash = "sha256:24da48596326ef8e448cfa837b454f638713d3531262375f00e5a9681682fc07", size = 169017, upload-time = "2026-07-28T06:05
:28.501Z" },
    { url = "https://files.pythonhosted.org/packages/c3/f2/3d1e47ea81b822210f5df1bf942fd90780a75c055243d569b664529dea88/wrapt-2.3.0-cp314-cp314-muslli
nux_1_2_riscv64.whl", hash = "sha256:cd3a2edf0427013736b8127955cec62608c56e53ea47e82812ea32059cda407f", size = 159065, upload-time = "2026-07-28T06:05
:30.01Z" },
    { url = "https://files.pythonhosted.org/packages/43/a5/ef2066ced8e5fca204e2b361e9708e36555b40949c583d997ea3b590817d/wrapt-2.3.0-cp314-cp314-muslli
nux_1_2_x86_64.whl", hash = "sha256:4fa0df3bff4e7ce45759f33fd39335fe2f60477bb9ecf7b8aa41e7d07ee36a23", size = 168821, upload-time = "2026-07-28T06:05:
31.649Z" },
    { url = "https://files.pythonhosted.org/packages/d5/e1/016104650d4e572fa91506eb396b3dd8efbccc9284fdc1c9479c3d21db28/wrapt-2.3.0-cp314-cp314-win32.
whl", hash = "sha256:2935d5454b3f179a29b12cf390ee47246740ba2c3a7545b1b46ba31a5f2a4a0b", size = 78700, upload-time = "2026-07-28T06:05:33.391Z" },
    { url = "https://files.pythonhosted.org/packages/3d/97/6fdc20a9f2ca304748b3f0819cbf377d55260562777bf0b615431bc3c181/wrapt-2.3.0-cp314-cp314-win_am
d64.whl", hash = "sha256:cc2cea812e5cb179a796b766747e7d3b21088760d8deb95676d482b8c8e6fa7d", size = 81422, upload-time = "2026-07-28T06:05:34.774Z" },
    { url = "https://files.pythonhosted.org/packages/5e/a4/9cbd53bf05746bea2c392af39cb052427a8ec95cbd494d930733d8f44681/wrapt-2.3.0-cp314-cp314-win_ar
m64.whl", hash = "sha256:22cc5c0a717bd4da87018ae0bffd4c19c6fb679d3ff357216ba566ab26c76cab", size = 80639, upload-time = "2026-07-28T06:05:36.228Z" },
    { url = "https://files.pythonhosted.org/packages/43/bb/6c5e4a0f66ea0d2b2dd267e8dd05a0014eea56840b3c8595d40b0a5d1f91/wrapt-2.3.0-cp314-cp314t-macos
x_10_15_x86_64.whl", hash = "sha256:a6b5984cd65dd639546f0eb4b8eacf1c31cb2fe9fb5c27bffe240987cdb2cf84", size = 84030, upload-time = "2026-07-28T06:05:3
7.714Z" },
    { url = "https://files.pythonhosted.org/packages/6a/eb/a1aedf03283bc9cbf8a1783995ddc54e3c5a86878f19002d2c428494f4c5/wrapt-2.3.0-cp314-cp314t-macos
x_11_0_arm64.whl", hash = "sha256:c88abcf53daef80e01a75c7530e727fa6e2c1888fe83e3dcdba4c96216a1f5c7", size = 84419, upload-time = "2026-07-28T06:05:39.
131Z" },
    { url = "https://files.pythonhosted.org/packages/63/61/50d511c0dc5105563849e86daa3e16ac7feef699f79fb05af45ea70107d5/wrapt-2.3.0-cp314-cp314t-manyl
inux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl", hash = "sha256:85de890ff968196e92dd1ae73a9fb8970495e7650a457b1c9ef0ac3dd550bce2", size =
 207171, upload-time = "2026-07-28T06:05:40.69Z" },
    { url = "https://files.pythonhosted.org/packages/3f/59/9b538cf7795217e810699d16bc88b96a830d9b5c403eb2ec2db6b5f2ae81/wrapt-2.3.0-cp314-cp314t-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:50f416b74d092bb9f41b424e90dd457f365f7ba4b11de62a23679769a21bd85c",
 size = 214329, upload-time = "2026-07-28T06:05:42.287Z" },
    { url = "https://files.pythonhosted.org/packages/b3/28/9935d62b1499e5c8b3d191e99ba4eb31ca237a0b699142011a837e9dc7ea/wrapt-2.3.0-cp314-cp314t-manyl
inux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:39febbee6d77301d31da6996b152ce52452da7c7ef72aba10c2fa976dff9c295", size = 199079, upload
-time = "2026-07-28T06:05:43.958Z" },
    { url = "https://files.pythonhosted.org/packages/2b/01/4446b80fa2ffa47a3449b250d004ba1c1937f07f64a179608fec735df866/wrapt-2.3.0-cp314-cp314t-musll
inux_1_2_aarch64.whl", hash = "sha256:93513bec052c6cd987f9f580c3df068c8bc4ebae6543736be3ca7ec5959cafcd", size = 209992, upload-time = "2026-07-28T06:0
5:45.677Z" },
    { url = "https://files.pythonhosted.org/packages/d4/07/56f26c9f9979586a021e8148747004aba4498f49458c90b0502969b904e1/wrapt-2.3.0-cp314-cp314t-musll
inux_1_2_riscv64.whl", hash = "sha256:729126e667da34d251b8ebf8a45ef0c5ddadc21542b3d6e1abf4259ece6508df", size = 196334, upload-time = "2026-07-28T06:0
5:47.608Z" },
    { url = "https://files.pythonhosted.org/packages/8b/41/6d7bcc895b0f28b2250e10908f060687b9165429dcd7f22ddb3d4c031b74/wrapt-2.3.0-cp314-cp314t-musll
inux_1_2_x86_64.whl", hash = "sha256:626b69db2021aa01671ec7bbc9740e558522bd44c18cf2ce69bf3d666a014109", size = 202644, upload-time = "2026-07-28T06:05
:49.183Z" },
    { url = "https://files.pythonhosted.org/packages/cd/25/7860927edba06b758b8852a6f02e832be715563c67a6795d94350bc81099/wrapt-2.3.0-cp314-cp314t-win32
.whl", hash = "sha256:629d73378082c00a8173031f9fb30a3ac6abbc894a5bfdfae71fabc60642d501", size = 79685, upload-time = "2026-07-28T06:05:50.976Z" },
    { url = "https://files.pythonhosted.org/packages/c4/0f/270bafe92fde3b069a39bc01e39ee79340895b335640df861d43d2a51885/wrapt-2.3.0-cp314-cp314t-win_a
md64.whl", hash = "sha256:42869085687f0aefd57c0f636c3f9354f8ffb321a8ba9cb52d19beb796e561c5", size = 83104, upload-time = "2026-07-28T06:05:52.405Z" },
    { url = "https://files.pythonhosted.org/packages/55/b3/af176d79a8515a8a720eccdad9a96f6e31a30abf2865430c8c42adf2fd13/wrapt-2.3.0-cp314-cp314t-win_a
rm64.whl", hash = "sha256:b1e5aa486e269b00ed35e64771c7d0ab8096cfd2643405ca8cd60ebedc099a51", size = 81774, upload-time = "2026-07-28T06:05:53.902Z" },
    { url = "https://files.pythonhosted.org/packages/00/39/3daf9f47be208606586de4568ba6713db53ebc8fd7a575aea1fe57983b69/wrapt-2.3.0-py3-none-any.whl",
 hash = "sha256:d8c7ed08477429752b8c44991f40ad7838b18332a160698740a6bfbc10d998a2", size = 61866, upload-time = "2026-07-28T06:06:12.9Z" },
]

[[package]]
name = "xxhash"
version = "3.8.1"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/8e/63/71aa56b151a1b28770037a61bd4e461c2619cfc8866a4fcaf1548605e325/xxhash-3.8.1.tar.gz", hash
 = "sha256:b0de4bf3aa66363552d52c6a89003c479911f12098cd48a53d44a0f7a25f7c46", size = 86223, upload-time = "2026-07-06T10:49:58.937Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/42/91/f65c34a7aa7b4e7cf4854f8e6ef3f7ee32ceac41d4f008da0780db0612f6/xxhash-3.8.1-cp312-cp312-macos
```

```
x_10_13_x86_64.whl", hash = "sha256:e6e49370822c1f4d8d90e678b06dbcb08b51a026a7c4b55479e7d467f2e813bc", size = 34680, upload-time = "2026-07-06T10:44:4
0.932Z" },
    { url = "https://files.pythonhosted.org/packages/57/04/b10a245a4c09a9cfa88f8e9ae755029413ad1ac17047f9a61906e5ae0799/xxhash-3.8.1-cp312-cp312-macos
x_11_0_arm64.whl", hash = "sha256:220d68130f83f7cc86d6edfdeab176adc73d7200bf3a8ec10c629e8cf605c215", size = 32397, upload-time = "2026-07-06T10:44:42.
196Z" },
    { url = "https://files.pythonhosted.org/packages/3a/75/45ab795b5945b6388583bd75202106af505537935566c15a1577797a0e08/xxhash-3.8.1-cp312-cp312-manyl
inux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:4d365ee1892c1fa803536f8c6ce21d24b29c9718ec75eb856095c07830f8c478", size = 22054
9, upload-time = "2026-07-06T10:44:43.603Z" },
    { url = "https://files.pythonhosted.org/packages/13/44/5ba2bd0a14ddf4193fc7d8ec29625f659f22c06d60b28f04bf46305d8330/xxhash-3.8.1-cp312-cp312-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:852bfe059720632e2f16a6a4745e41d20937b2bf2a42a401e2412046bb6971cc",
 size = 241186, upload-time = "2026-07-06T10:44:45.534Z" },
    { url = "https://files.pythonhosted.org/packages/23/32/c4147def4d1e4538b906f82731e0ba23424377fc50a7cddd03cd284c8f63/xxhash-3.8.1-cp312-cp312-manyl
inux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:2f8c25a7061d952de589bd0ea0eaadee32378ff83dd6a677b267f9cd86f401f8", si
ze = 264852, upload-time = "2026-07-06T10:44:47.199Z" },
    { url = "https://files.pythonhosted.org/packages/6c/bd/71ed14f4f0318bb7fd7b2ec51999413487fa8da8d41208e84d50d1ef0f98/xxhash-3.8.1-cp312-cp312-manyl
inux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:868a8dcaff1a84ba78038e1cef14fc88ccf84d9b4d12ea604696e0693296aa56",
 size = 242663, upload-time = "2026-07-06T10:44:48.846Z" },
    { url = "https://files.pythonhosted.org/packages/91/09/70af22c565a8473b3f2ae73f88e7721af281bc4a575236dbd1970c9f76f6/xxhash-3.8.1-cp312-cp312-manyl
inux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:6536d8677d2fff7e64cd0b98b976df9de7aee0e69590044c2af5f51b76b7a170", size
= 473510, upload-time = "2026-07-06T10:44:50.695Z" },
    { url = "https://files.pythonhosted.org/packages/18/96/34db781c8f0cf99c544ca1f2bc2e5bf55426e1eb4ca6de8ea5da56a9f352/xxhash-3.8.1-cp312-cp312-manyl
inux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:82c0cedd280eab2e8291270e6c04894dbc096f8159a39dcf1807429f026ca3cc", si
ze = 220469, upload-time = "2026-07-06T10:44:52.422Z" },
    { url = "https://files.pythonhosted.org/packages/93/5f/9a184f615fa5a4dce30c01534f62946ce5a11ce40f73785cbd356ccabaa9/xxhash-3.8.1-cp312-cp312-manyl
inux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:daa86e4b68221d38e669bb236ba112d0335353829fb627c82e5909e4bbe8694c", size = 310290, upload
-time = "2026-07-06T10:44:54.142Z" },
    { url = "https://files.pythonhosted.org/packages/a9/dc/9b9a9789011ee153723a5eb9e7dd7fcbae2ba9b3fe7a729249ca7c252056/xxhash-3.8.1-cp312-cp312-musll
inux_1_2_aarch64.whl", hash = "sha256:2bc7113e6f2b6b3922dd61796ca9f36af09da3773898e7003038dc992fc83b8d", size = 238173, upload-time = "2026-07-06T10:4
4:55.693Z" },
    { url = "https://files.pythonhosted.org/packages/ec/4d/71c6005ada9dcb608a4e1902e8475ecadb5f3fbfa04e1e244d276a2d0c43/xxhash-3.8.1-cp312-cp312-musll
inux_1_2_armv7l.whl", hash = "sha256:5eed32dad81d6ba8e62dc7b9ffa0500199385d7810a8dd9d4eafaceb8c6e20bb", size = 269026, upload-time = "2026-07-06T10:44
:57.424Z" },
    { url = "https://files.pythonhosted.org/packages/2f/87/d6c036ba25dfbd9c8633be5aa86fc9474bbb9e2c68212a841d090abe7344/xxhash-3.8.1-cp312-cp312-musll
inux_1_2_i686.whl", hash = "sha256:83697b0ea1f10e7f5d8b26a4906fa851393c61546c63839643a2b7fe2d868061", size = 224970, upload-time = "2026-07-06T10:44:5
9.085Z" },
    { url = "https://files.pythonhosted.org/packages/48/62/4c1f035a41c5752aa05e195b6c904c07b94fe9061a16de61e72a6e6b135f/xxhash-3.8.1-cp312-cp312-musll
inux_1_2_ppc64le.whl", hash = "sha256:36fc69160465ae75c6ec4ac9f781bb2aa16ae7ff869e73c26fee85fbb11b9887", size = 240820, upload-time = "2026-07-06T10:4
5:00.746Z" },
    { url = "https://files.pythonhosted.org/packages/da/14/d39d565069b87e86d21a2af2a31d04db79249d25aa8d5b62959056a89857/xxhash-3.8.1-cp312-cp312-musll
inux_1_2_riscv64.whl", hash = "sha256:445e0f5a31f2f3546ae0895d4811e159518cdc9d824c11419898d40cfadb677e", size = 300619, upload-time = "2026-07-06T10:4
5:02.716Z" },
    { url = "https://files.pythonhosted.org/packages/13/22/75467acc887edc8cf71c97ab1708feb3df7a88bda589b9f399765c6387d2/xxhash-3.8.1-cp312-cp312-musll
inux_1_2_s390x.whl", hash = "sha256:dfe0580fbfd5e4af87d0cc52d2044f155d55ebd8c8a93568758a2ea7d8e15975", size = 443267, upload-time = "2026-07-06T10:45:
04.653Z" },
    { url = "https://files.pythonhosted.org/packages/a4/b6/1da3baa5fa6ef705e3425fddd382be7dfc4dfba2686df90a20f16e9c7b1b/xxhash-3.8.1-cp312-cp312-musll
inux_1_2_x86_64.whl", hash = "sha256:095e1323fa108be1292c54c86da3ef3c7a7dc015b105a52133973bc07a6ad11a", size = 217338, upload-time = "2026-07-06T10:45
:06.304Z" },
    { url = "https://files.pythonhosted.org/packages/78/dd/b5295a9f97484e7a1c2b283a742ca45e3104991c55a1ef670dde161829ba/xxhash-3.8.1-cp312-cp312-win32
.whl", hash = "sha256:bf28f55e427e0483acb1f666bd0d869b6d5e5a716680c216ad7befe3d4cfba2e", size = 31970, upload-time = "2026-07-06T10:45:07.823Z" },
    { url = "https://files.pythonhosted.org/packages/ec/31/3fa0b807d7e21515cd975e7fe5c039d52ac3e9401a96d6ad68dae6305215/xxhash-3.8.1-cp312-cp312-win_a
md64.whl", hash = "sha256:2256e80e4960ee282f63428adb349cb7f8bd8efe4db770d88eb815f4b9860724", size = 32741, upload-time = "2026-07-06T10:45:09.42Z" },
    { url = "https://files.pythonhosted.org/packages/b8/05/86feada74e239600e6875aa507afb40482a89b92700aa74a92da83bdcb77/xxhash-3.8.1-cp312-cp312-win_a
rm64.whl", hash = "sha256:9df56e6df96a60590935e22373041cccc91fd55858763dcffb55bf63b3a2b396", size = 29234, upload-time = "2026-07-06T10:45:10.809Z" },
    { url = "https://files.pythonhosted.org/packages/6b/8c/446bb782cd0d27007a917b5569a08dd73219c3e8d6e459014db104b27bdb/xxhash-3.8.1-cp313-cp313-andro
id_21_arm64_v8a.whl", hash = "sha256:3c682fcd96eb4bf64be32a4d95f96107e1588005831bd8a741b324fdda01b913", size = 38562, upload-time = "2026-07-06T10:45:
12.425Z" },
    { url = "https://files.pythonhosted.org/packages/d7/ec/c0c45627eaa6be7a5d6117423adf8f7a15b17ee74b4b17072cca5959a225/xxhash-3.8.1-cp313-cp313-andro
id_21_x86_64.whl", hash = "sha256:036a024d8b9c01f70782e09ed98d532e76fd23f950ae7154bd950fe94e90ebec", size = 36656, upload-time = "2026-07-06T10:45:13.
932Z" },
    { url = "https://files.pythonhosted.org/packages/f6/94/8324c04cc7597154caaeba6c094e01fbd2e7601d01e7a13eea9f5420e77b/xxhash-3.8.1-cp313-cp313-ios_1
3_0_arm64_iphoneos.whl", hash = "sha256:d6a5c0bce213b23b0166fe0d35bcbbe23ce4b968f257cc7eb6fd57cb8e1e6297", size = 31169, upload-time = "2026-07-06T10:
45:15.687Z" },
    { url = "https://files.pythonhosted.org/packages/40/a4/beb6bb26e1184e126dbe7a5682330214ef54dcfbf882078aa9f4b5428d42/xxhash-3.8.1-cp313-cp313-ios_1
3_0_arm64_iphonesimulator.whl", hash = "sha256:5177aa44eddaa97c6ef0cc00c6d540edb64d51781d2f8fb941612ec61a92c9ed", size = 32177, upload-time = "2026-07
-06T10:45:17.035Z" },
    { url = "https://files.pythonhosted.org/packages/56/0f/fc4c92a5a528f839b34b6419b2e53c8597f2a629d5a1f5d721f65bfa1fd6/xxhash-3.8.1-cp313-cp313-ios_1
3_0_x86_64_iphonesimulator.whl", hash = "sha256:7801b7223db017b9c0c9ccf37e44524edb35a1544a1c032add22c061c6af0276", size = 34642, upload-time = "2026-0
7-06T10:45:18.39Z" },
    { url = "https://files.pythonhosted.org/packages/d4/58/edbfb141d4000767ac6a9694f8ac0763e2c2e983e65c9e31620ba56e2667/xxhash-3.8.1-cp313-cp313-macos
x_10_13_x86_64.whl", hash = "sha256:9e80238259655bf69d7bcd08226a970d7f42605f3157786bfa76dd13472d7fa0", size = 34684, upload-time = "2026-07-06T10:45:2
0.033Z" },
    { url = "https://files.pythonhosted.org/packages/07/3f/5072f1f0f5714186f0ac2a0b5a4929ce30d4b845e94886b6c01b6ebda0be/xxhash-3.8.1-cp313-cp313-macos
x_11_0_arm64.whl", hash = "sha256:bcab50a389cc04d87f90092af78a6adba2ab3deca63175a3344ca83514045315", size = 32401, upload-time = "2026-07-06T10:45:21.
414Z" },
    { url = "https://files.pythonhosted.org/packages/49/c7/802ea2f9c2ed59219934d6d65c470d502b1788043eae277a52af8658bda6/xxhash-3.8.1-cp313-cp313-manyl
inux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:a2489d3a776fa380cb8e71f54c7fda268a9baf3de9b1395093fd280f95735907", size = 22061
7, upload-time = "2026-07-06T10:45:23.234Z" },
    { url = "https://files.pythonhosted.org/packages/99/a8/e10488efd31fcb13fcd6acbc6e788f10c6f8e3a0cc4ae3eb89dc19c55a12/xxhash-3.8.1-cp313-cp313-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:32ab1e5432690276e71192be7401b55f96db2d0eedea5d44eb1f164505669cc0",
 size = 241295, upload-time = "2026-07-06T10:45:25.364Z" },
    { url = "https://files.pythonhosted.org/packages/18/cc/14180b17d44892a631f8ae7323c30bfbb1328efc8209e528a480293528ac/xxhash-3.8.1-cp313-cp313-manyl
inux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:b30e01a0b97a4bc3f519a4d7a82da3dc53251fb0de5eeea8660dcd4ff094c0c2", si
ze = 264688, upload-time = "2026-07-06T10:45:27.09Z" },
    { url = "https://files.pythonhosted.org/packages/a9/72/a14019d0c5f6c41ee407a503036ae32787c91325ca218a96a9b5627be651/xxhash-3.8.1-cp313-cp313-manyl
inux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:1f44275ddb0978b67a58a951501903f04d49335a91f7681c9ce122ecb8ccb329",
 size = 242740, upload-time = "2026-07-06T10:45:28.753Z" },
    { url = "https://files.pythonhosted.org/packages/68/08/92550e556c6fcfcb96c6a336945eb53a431ed43120ed749636debb16c5cf/xxhash-3.8.1-cp313-cp313-manyl
inux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:e3b87cbd974512c0c5fc7b469c36b2cdc9ee6d76e4ec78bccb2c7184611c49b0", size
= 473599, upload-time = "2026-07-06T10:45:30.524Z" },
    { url = "https://files.pythonhosted.org/packages/29/83/e361d3c1acd1b21e1d489616de6fa4aaf843365d8179f612e3743eac20a9/xxhash-3.8.1-cp313-cp313-manyl
inux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:98ee81b4b7f3023c9cb04a78cc67610baffcb5812d92f2096cb5a5efc6f19437", si
ze = 220559, upload-time = "2026-07-06T10:45:32.979Z" },
    { url = "https://files.pythonhosted.org/packages/05/01/006a4243c2c2a6831827f9999f6d1c23feeef100eb023c1f886022a00bf3/xxhash-3.8.1-cp313-cp313-manyl
inux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:2666f059a1588a99267e33605365ed89cea92f424b3522806a9f4bd8ad2e3d62", size = 310383, upload
-time = "2026-07-06T10:45:35.875Z" },
    { url = "https://files.pythonhosted.org/packages/d8/20/af388e8bf9f9a0f89eeef7d2a1935d176ee1c20bc6adeda05035879379cf/xxhash-3.8.1-cp313-cp313-musll
inux_1_2_aarch64.whl", hash = "sha256:b0093cf7eeb91b84776e8742113afa4bdf47533d36cf719179aaaf1f56f6f8bf", size = 238228, upload-time = "2026-07-06T10:4
5:38.02Z" },
    { url = "https://files.pythonhosted.org/packages/63/6b/4666579a87eebd1744663c404297355fa0658617b015cedfa58810ee7036/xxhash-3.8.1-cp313-cp313-musll
inux_1_2_armv7l.whl", hash = "sha256:3a800912a2e5e975d4128969d645c4a2a80aa886ccd6c9b1c6f44529e327e8cf", size = 269137, upload-time = "2026-07-06T10:45
:39.954Z" },
    { url = "https://files.pythonhosted.org/packages/de/d3/e963a8a46f900a137d91b02144d8ea07a8f812971b138204a3b2f8b8e55c/xxhash-3.8.1-cp313-cp313-musll
inux_1_2_i686.whl", hash = "sha256:0fe37f72a207223d22a4eddc3149d4298993385aa9daef25c039246ca5a309f3", size = 225068, upload-time = "2026-07-06T10:45:4
1.718Z" },
    { url = "https://files.pythonhosted.org/packages/aa/80/9d181dbcde4b0fe48375f48833a5832d4b8cd2b349b15110c92ee472d874/xxhash-3.8.1-cp313-cp313-musll
```

```
inux_1_2_ppc64le.whl", hash = "sha256:5db43f249b4be9f99ef4b967863f37094fb40e67effafb78ba4f0356b6396104", size = 240874, upload-time = "2026-07-06T10:4
5:43.414Z" },
    { url = "https://files.pythonhosted.org/packages/39/15/ce3ab5a1cd27ead25a5196e55a7284220f6ad6e316da494ffd900b2b600f/xxhash-3.8.1-cp313-cp313-musll
inux_1_2_riscv64.whl", hash = "sha256:c4ed42965c2cd9081f011be22f69d0e65d3b6165fe7734072fd0c232840bbd4e", size = 300702, upload-time = "2026-07-06T10:4
5:45.135Z" },
    { url = "https://files.pythonhosted.org/packages/96/c0/2281a8ab5f2a62dbf57a23c58a01ccc1d98abf40f71193c8a81f59e759b5/xxhash-3.8.1-cp313-cp313-musll
inux_1_2_s390x.whl", hash = "sha256:3557bec8fcb11738a8920eeb68974bc76b75262f6947998d3147954ce0a4b893", size = 443351, upload-time = "2026-07-06T10:45:
47.188Z" },
    { url = "https://files.pythonhosted.org/packages/81/2e/071a58c1a53a52d4f7a3aa0987be0c396dffd40da8204805fe1b130a81f4/xxhash-3.8.1-cp313-cp313-musll
inux_1_2_x86_64.whl", hash = "sha256:00de40f3b42240db23a82a5c682b55d7263d84a26a953240c1aee463409660e3", size = 217396, upload-time = "2026-07-06T10:45
:48.925Z" },
    { url = "https://files.pythonhosted.org/packages/68/44/36ab58134badd9d3433fc7b53c4ca8d113d8e807782885628640f8297a4d/xxhash-3.8.1-cp313-cp313-win32
.whl", hash = "sha256:b5196cc2574cfec572a5f3fb7cfa5ade27305ae3d06516a082132441aff4c83a", size = 31974, upload-time = "2026-07-06T10:45:50.591Z" },
    { url = "https://files.pythonhosted.org/packages/96/2a/2a0b84798448e766f7b89ceed073cb0cb5a43fc9ebbacbdea74a38de18e3/xxhash-3.8.1-cp313-cp313-win_a
md64.whl", hash = "sha256:538f5f865df6cd8c32dd63158a0e5b4f5dd08d732a7da8b7228a5a0776c8ce55", size = 32739, upload-time = "2026-07-06T10:45:52.221Z" },
    { url = "https://files.pythonhosted.org/packages/d4/60/bb51dbf7c363ff88a7cbd50b7959718219577ef44d7cf255929ffc4a2194/xxhash-3.8.1-cp313-cp313-win_a
rm64.whl", hash = "sha256:a6617f30641ba0d8baa1635fbefb1dffc5165ec36d26921bd5cee13497cd937a", size = 29239, upload-time = "2026-07-06T10:45:53.714Z" },
    { url = "https://files.pythonhosted.org/packages/56/d3/827ca123c2ee5443a6aaed3c5dd199237dc2f010e2bebd7ec09ef36f3a5f/xxhash-3.8.1-cp313-cp313t-maco
sx_10_13_x86_64.whl", hash = "sha256:bfcd82852c62a60e314670a9602de354c4460f8adad916e2e42a20860c7870bc", size = 34964, upload-time = "2026-07-06T10:45:
55.535Z" },
    { url = "https://files.pythonhosted.org/packages/05/67/67ae2a3ccdeb8b8ef025d35aee9edd1d26c3abe5051d47da9286232afbf8/xxhash-3.8.1-cp313-cp313t-maco
sx_11_0_arm64.whl", hash = "sha256:08ea2081f5e88615fec8622a9f87fbe21b8ea58d88cfc02163ca11026ee62a92", size = 32697, upload-time = "2026-07-06T10:45:57
.288Z" },
    { url = "https://files.pythonhosted.org/packages/38/5a/3d3994346e1f45493679cb5c1ffc2bf454e410e9d1e8a662d253becee91e/xxhash-3.8.1-cp313-cp313t-many
linux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:2e32855b6f9e5b18f449e59d45e3d5778bdeb660632ef2693cca267a11246c75", size = 2259
54, upload-time = "2026-07-06T10:45:58.897Z" },
    { url = "https://files.pythonhosted.org/packages/3f/2c/53169270309b7cd8e05504e07fe123bac053b89d00ac63617faacf0a2ec0/xxhash-3.8.1-cp313-cp313t-many
linux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:a6e088bd7870775624256a0d84c2a6714afd223b2eeb56b0ca58398e52a32fda"
, size = 249776, upload-time = "2026-07-06T10:46:00.977Z" },
    { url = "https://files.pythonhosted.org/packages/70/e0/5c551d8d592f944506f7c5185e210255c15e672a3c6008c156a1bd9b775e/xxhash-3.8.1-cp313-cp313t-many
linux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:72eb5ae575cc7ae2b23f6f8064a8b10f638c7149819ae9cc6d20ebd4d37a1629", s
ize = 274776, upload-time = "2026-07-06T10:46:02.869Z" },
    { url = "https://files.pythonhosted.org/packages/a0/2a/d3a762270cee2d7bcd0e25e28c623e5f3f5c0dc637b66e3e47dd5b0bb3f0/xxhash-3.8.1-cp313-cp313t-many
linux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:d0b48cdf690a64cedf7258c3dc9506cc41fc86edd7739c40e3098952265dc068"
, size = 252056, upload-time = "2026-07-06T10:46:04.688Z" },
    { url = "https://files.pythonhosted.org/packages/c1/8f/b78e4373b2cb6d1c42af60ea2d7e9146ad0710b239ac7f706d5d31d5bb98/xxhash-3.8.1-cp313-cp313t-many
linux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:fb9e256a357dfcede7818c6d34e70db2d6b664394803d1de4b6984d2de76c0f1", size
 = 482108, upload-time = "2026-07-06T10:46:06.498Z" },
    { url = "https://files.pythonhosted.org/packages/e6/0d/642d923336ea61a15f8ce64fc7e078729e6e06c3a026e517fa79b2c23b7a/xxhash-3.8.1-cp313-cp313t-many
linux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:51f71a6e2ad071e70c937e41fcb6c19f82c3f9f49831eba850ed4a106ffbb647", s
ize = 226739, upload-time = "2026-07-06T10:46:08.598Z" },
    { url = "https://files.pythonhosted.org/packages/a6/0a/a37d6da6427d45a8d23e3ee3a0ca9c9d4a90364849c6637fe2963a755f9b/xxhash-3.8.1-cp313-cp313t-many
linux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:e4a6443968c4e8dc69967e12776776a5952c119cc1bd94168ad1c5ad667c2be1", size = 319658, uploa
d-time = "2026-07-06T10:46:10.504Z" },
    { url = "https://files.pythonhosted.org/packages/4a/51/ebbd40da8a3f1bc53b4b7a9a87f8e28bd95c5f21bc14b8a57860cf367d1b/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_aarch64.whl", hash = "sha256:714503083a1f2065c9ad15340dd49ac8a8e948a505a705ffa1750cb951519113", size = 246059, upload-time = "2026-07-06T10:
46:12.634Z" },
    { url = "https://files.pythonhosted.org/packages/24/4c/d9014030147e1f0bb26e7da47aa240dd9ec61c763c573e558111d869f8e1/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_armv7l.whl", hash = "sha256:77f74e45a1e5574bbbf80181c8027b3a4c65c2248fffbd557bd596fff13102f9", size = 275535, upload-time = "2026-07-06T10:4
6:14.614Z" },
    { url = "https://files.pythonhosted.org/packages/84/86/caee2db41fadcd5a25aa4323213f9afec5a8586d4e419241e3d659362bd7/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_i686.whl", hash = "sha256:4e0e1b0fb0259c1b75d1251ac0bb4d7ab675d36f7a6bf4ba6aa630dae94f9ffa", size = 231292, upload-time = "2026-07-06T10:46:
16.452Z" },
    { url = "https://files.pythonhosted.org/packages/0b/60/f52f08bcdc904c4514ea5c25caa19e9f3214144434a6ff96dc82dc1cbddd/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_ppc64le.whl", hash = "sha256:10e4393ec33633c2f05ad01869e546ad080b1a18f2650503731f153774608b31", size = 250490, upload-time = "2026-07-06T10:
46:18.318Z" },
    { url = "https://files.pythonhosted.org/packages/24/a0/94dc7ae310838f250669c6ad7168e6d6fca17d49dac1053f06dc232c4a56/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_riscv64.whl", hash = "sha256:b3ba794c3d885803db6c3116686923f1ec13bc86e621e169a375282b63ea1cc6", size = 309861, upload-time = "2026-07-06T10:
46:20.503Z" },
    { url = "https://files.pythonhosted.org/packages/8b/f9/adeead7d0eb28cdfc2832544ea639ffbc6749ccde47a8e228d667459182e/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_s390x.whl", hash = "sha256:57189a69c0891e4818853feaa521c972d22c880a001453addea015f48e3c3398", size = 448739, upload-time = "2026-07-06T10:46
:22.79Z" },
    { url = "https://files.pythonhosted.org/packages/04/a4/22ec0e07db57d901c9298ae98aa3cf2be45bafded6f07c13131e85b89032/xxhash-3.8.1-cp313-cp313t-musl
linux_1_2_x86_64.whl", hash = "sha256:d59e71153fe9ff85648d00e18649b07e9b22c797291abb7e27274fa06df8b838", size = 223657, upload-time = "2026-07-06T10:4
6:24.831Z" },
    { url = "https://files.pythonhosted.org/packages/94/32/8a9531f37b59e5a013003db7cb7414baf4ce7e0e1268e0d5947cd3d6a2df/xxhash-3.8.1-cp313-cp313t-win3
2.whl", hash = "sha256:5b96f0024e9840f449bd91b2d005c921a4b666055a0d1b6492463799f32aae22", size = 32377, upload-time = "2026-07-06T10:46:26.86Z" },
    { url = "https://files.pythonhosted.org/packages/e7/ab/2ca45fd7f671de5f81fc297ef1c95080b40c86ec6be0cc6034b8f7707ac8/xxhash-3.8.1-cp313-cp313t-win_
amd64.whl", hash = "sha256:37d5a56c36dcc0b9a87b814cd992598d33863ff683749de6c86081f278d5e629", size = 33274, upload-time = "2026-07-06T10:46:28.39Z" },
    { url = "https://files.pythonhosted.org/packages/5a/54/20d7163463ddb6438b73a427d1655a77a502cf9b9b0c3ada3599629d9c0a/xxhash-3.8.1-cp313-cp313t-win_
arm64.whl", hash = "sha256:6696c8752aded28ff3b16f33ef28ce28fb5d209b80c206746f943199fcf5fd65", size = 29375, upload-time = "2026-07-06T10:46:29.962Z" }
,
    { url = "https://files.pythonhosted.org/packages/c2/8b/df2ba04f22a6cd6b39f96a6577329a8471a55c90ef8d8e2f7c102363613f/xxhash-3.8.1-cp314-cp314-andro
id_24_arm64_v8a.whl", hash = "sha256:9db455cb649dcfe4504d6d68a6d83a7315a99a3ca59871dc3ff840671f99adba", size = 38430, upload-time = "2026-07-06T10:46:
31.496Z" },
    { url = "https://files.pythonhosted.org/packages/b2/4f/6a059e8ad3ca8deedc91dfe335b21120490089515221 2c03ebbe721de68b/xxhash-3.8.1-cp314-cp314-andro
id_24_x86_64.whl", hash = "sha256:affb37f152e55b5e4494bb9d0107f7bb08515c6704fbed82d9f61214d74adc17", size = 36558, upload-time = "2026-07-06T10:46:33.
078Z" },
    { url = "https://files.pythonhosted.org/packages/cb/95/40be178205acce092ae418feb20ac737b32a02c7b864926ed0717354c9f8/xxhash-3.8.1-cp314-cp314-ios_1
3_0_arm64_iphoneos.whl", hash = "sha256:460261045936975193bfd20549a0de1cd52a33b405cbb972f0d80940c42266cd", size = 31181, upload-time = "2026-07-06T10:
46:34.793Z" },
    { url = "https://files.pythonhosted.org/packages/3f/89/2da4dbf051bafa156c0e3f12012db2b0ac3b84ff37ca1f021f6bfffcdfbb/xxhash-3.8.1-cp314-cp314-ios_1
3_0_arm64_iphonesimulator.whl", hash = "sha256:38c887aedb696ef8bca19983206d270848558cfae4a91afa6a2fb05dde58ffc5", size = 32192, upload-time = "2026-07
-06T10:46:36.393Z" },
    { url = "https://files.pythonhosted.org/packages/7c/4e/e000bbae3566bc8e0be771a8a0f294aa99075e3f0bc4ef43922ebffdebc8/xxhash-3.8.1-cp314-cp314-ios_1
3_0_x86_64_iphonesimulator.whl", hash = "sha256:594131ce1aad18db3689781f806db1b065cdaa04f4df36b4c038d2013aefd0bf", size = 34691, upload-time = "2026-0
7-06T10:46:38.1Z" },
    { url = "https://files.pythonhosted.org/packages/b4/4a/ea954aacc7d1c8711880ac2b55da94429a9b4296b151c4fc0966549ca1ee/xxhash-3.8.1-cp314-cp314-macos
x_10_15_x86_64.whl", hash = "sha256:78c794b643d214f1522e7a288bcf5a2de120d26cd170516749a4009dc92722c9", size = 34807, upload-time = "2026-07-06T10:46:3
9.647Z" },
    { url = "https://files.pythonhosted.org/packages/ca/29/df598e738ff37558ac627264deb2e560902d9bf7f46d3bd5175c9eee593e/xxhash-3.8.1-cp314-cp314-macos
x_11_0_arm64.whl", hash = "sha256:af0c9fedc4a2c24e8664953882fe8185f3790b8338c9c700f76f5ad660817711", size = 32410, upload-time = "2026-07-06T10:46:41.
359Z" },
    { url = "https://files.pythonhosted.org/packages/59/9c/81ab40e7d33ada0b3df5d1bc884894d15dbf4f805cd645b685e4606bb8e0/xxhash-3.8.1-cp314-cp314-manyl
inux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:115772daeb71b2f3b9381177017f53e6cf3f3439c840737fdabd21aba6e54920", size = 22056
4, upload-time = "2026-07-06T10:46:43.463Z" },
    { url = "https://files.pythonhosted.org/packages/fd/6f/62ae6f5c8606320a0e2a41c2dc8c6d91cc5d63d0f84dd9582e9543779dd8/xxhash-3.8.1-cp314-cp314-manyl
inux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:000435984a0469b0f822fe76f35bddea0f96a4d6521b3339a60a6428cdee1edc",
 size = 241462, upload-time = "2026-07-06T10:46:45.509Z" },
    { url = "https://files.pythonhosted.org/packages/15/a1/9c3a0ec6cb524396f551eddd102a76690a795494eb9784fc67542b0daa37/xxhash-3.8.1-cp314-cp314-manyl
inux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:2f1c68394818e0595569c2ff3cbc1e6d5a36a434e796f5c526b987b80c8a8c62", si
ze = 264491, upload-time = "2026-07-06T10:46:47.655Z" },
    { url = "https://files.pythonhosted.org/packages/64/f2/700a4674e4308eb59d2fdb973977e82eae231bea5044753fee5c9eec0e0c/xxhash-3.8.1-cp314-cp314-manyl
inux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:46b39976d008e2a845758650f0ff7136bca004f40da0c8798bd37ac37860154f",
 size = 242905, upload-time = "2026-07-06T10:46:49.857Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/f3/8a/72d9874375c8d4cbc64a8cd1d659d5695a8765c3db82efa82dc5bd9f14d0/xxhash-3.8.1-cp314-cp314-manyl
inux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:d5006c65ec507a333479e76e00e2c368781f16c24ededa764763956b32a0e93e", size
= 473873, upload-time = "2026-07-06T10:46:51.953Z" },
    { url = "https://files.pythonhosted.org/packages/03/f0/6db07590ed7e0a77f186ef0bcea8d52553bf1ba57833e09467a2411f0f2d/xxhash-3.8.1-cp314-cp314-manyl
inux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:c31a2649bcf1fe97cf11c79848d761df33ac46b3896942d31b640557b486ff6b", si
ze = 220765, upload-time = "2026-07-06T10:46:55.41Z" },
    { url = "https://files.pythonhosted.org/packages/8f/10/00d12d8b8beabbf49a8bbc626fb9f40445145a8887eb41a6acfb69149ac4/xxhash-3.8.1-cp314-cp314-manyl
inux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:8f759eed402448c2bdbb492e4fba1f20668ffe29688605ea61f0f67f9e4e386d", size = 310478, upload
-time = "2026-07-06T10:46:57.729Z" },
    { url = "https://files.pythonhosted.org/packages/1f/f9/12a82394eefb0f185d15a7f7b9f627c61c475a72dd83718436a5b84b42ac/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_aarch64.whl", hash = "sha256:7b5f97ecfede10d5b2870383620e2d25c8561e217c7bf9081073802b54248d2b", size = 238393, upload-time = "2026-07-06T10:4
6:59.87Z" },
    { url = "https://files.pythonhosted.org/packages/20/f3/53f963e320b9ce678337aa7273f39ce692ded8b99e3d22a866ec722159ab/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_armv7l.whl", hash = "sha256:1da930bbcac3e8fbe2191850e2abb57977a99348c12c4b385e1058ac1b0a9ecc", size = 268704, upload-time = "2026-07-06T10:47
:01.806Z" },
    { url = "https://files.pythonhosted.org/packages/0a/50/5b5badbd87c82d9f9b5f58ac74a3f29ef08f6fc387b324b8fd482450b862/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_i686.whl", hash = "sha256:747476436f6891b9773374ce8d48edcc8b12cb5b61b67c6fb6289633747d088f", size = 225015, upload-time = "2026-07-06T10:47:0
3.784Z" },
    { url = "https://files.pythonhosted.org/packages/30/93/3ca68265afe7b4e69435e08a7b6a1d9d0f2a071e889da1f8041ed00fe878/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_ppc64le.whl", hash = "sha256:4ef09bbc2519a93cd0f95f2ceb5f7b85919dffea643278e02362bf40e3c4bed1", size = 240951, upload-time = "2026-07-06T10:4
7:05.816Z" },
    { url = "https://files.pythonhosted.org/packages/dd/a6/27e19670c40f46b5e76e11f2f4713d21054804568425d870670e757172ad/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_riscv64.whl", hash = "sha256:a5eed9d41995a83f3332b4e3396abb7f433cac584222bd7e305b606d8353861e", size = 300751, upload-time = "2026-07-06T10:4
7:07.95Z" },
    { url = "https://files.pythonhosted.org/packages/bc/fb/b33e27689959fe7ed2ae0b830af41560d65213943983afa9db3a8d481bce/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_s390x.whl", hash = "sha256:53f3ed9118397074ff63a79b66b7fec1c84c782eecde35c5bc94e420a971c231", size = 443480, upload-time = "2026-07-06T10:47:
10Z" },
    { url = "https://files.pythonhosted.org/packages/26/60/0e0d973be5fe280753ef02fbc89349492ad6e903bf1dcb870b668f94b662/xxhash-3.8.1-cp314-cp314-musll
inux_1_2_x86_64.whl", hash = "sha256:d247b34bf433c92b41689318fd25d246313cab2275a6a47e2efac178b80d6efe", size = 217657, upload-time = "2026-07-06T10:47
:12.196Z" },
    { url = "https://files.pythonhosted.org/packages/ad/68/c9e3ecef4a9a417d464cb5bd200aa12f73192dee677901b9e08e0ad0d1bb/xxhash-3.8.1-cp314-cp314-win32
.whl", hash = "sha256:d58ce8b6cfa9c4d2f230557f69caf7c06369e318015d0b19485095bc2c5963ab", size = 32690, upload-time = "2026-07-06T10:47:14.204Z" },
    { url = "https://files.pythonhosted.org/packages/d7/99/e9e44588c0b62837bbec5ba7927816de0afa03406b1a0b6c7a7e1d1a30a0/xxhash-3.8.1-cp314-cp314-win_a
md64.whl", hash = "sha256:6cee733fe4ccb1737e0997135283c82341e5cfa9cf214b165f9087fb663aaf4f", size = 33460, upload-time = "2026-07-06T10:47:16.021Z" },
    { url = "https://files.pythonhosted.org/packages/45/2b/64f36d86380b3657ad9031967ab814f3ef31307174650853f69c18932ebc/xxhash-3.8.1-cp314-cp314-win_a
rm64.whl", hash = "sha256:58346024d47e84f7d8b3e7f5d6faa1d58acbbe49a8771497872059f58c1d8ea5", size = 30092, upload-time = "2026-07-06T10:47:17.81Z" },
    { url = "https://files.pythonhosted.org/packages/92/cb/18b64bff88c58a0ca209dc533e63cf02d7ae5aa6b1b9a9fd14e81b5dbd60/xxhash-3.8.1-cp314-cp314t-maco
sx_10_15_x86_64.whl", hash = "sha256:01cab782f8a0a05ecad2c63d7ef10f7ab475f660e0d6419d069418c14d88de7c", size = 35024, upload-time = "2026-07-06T10:47:
19.821Z" },
    { url = "https://files.pythonhosted.org/packages/af/1d/72d8a70520e5dcddb472ea0486d299da3240745a10658290cd7b5690ede2/xxhash-3.8.1-cp314-cp314t-maco
sx_11_0_arm64.whl", hash = "sha256:717b12fdc51819833704e85e6926d76981ffa3f780ef92e33ebb8b26d46bb230", size = 32697, upload-time = "2026-07-06T10:47:21
.649Z" },
    { url = "https://files.pythonhosted.org/packages/9c/b8/e041f555903c56db3d0a731b3d72a6575d75e0ed868b1bd2e5176111ca44/xxhash-3.8.1-cp314-cp314t-many
linux1_i686.manylinux_2_28_i686.manylinux_2_5_i686.whl", hash = "sha256:ec55d80e9b8a519d742669e0b49e8ce9e6747be42bf3c138158b6543a9c8e489", size = 2260
44, upload-time = "2026-07-06T10:47:23.612Z" },
    { url = "https://files.pythonhosted.org/packages/3a/7e/5cdcf06bf6ec4b5d2ac073feb23432ec1d603fd438864cbd2c09c7cb45e1/xxhash-3.8.1-cp314-cp314t-many
linux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:98d8ac1129b4dd39098cffed94d1284aceb61c3aa396757ccc736ac392e4cee5"
, size = 249899, upload-time = "2026-07-06T10:47:25.812Z" },
    { url = "https://files.pythonhosted.org/packages/c0/c0/eb7e059cb5e1dba11fd30d2fdf882f56e5a417a3eaa43669d43623767f45/xxhash-3.8.1-cp314-cp314t-many
linux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:3bc0fa90830df1e1277f33cc6e55de9990b83c0319fd8c7412866cfde38b025e", s
ize = 274892, upload-time = "2026-07-06T10:47:27.931Z" },
    { url = "https://files.pythonhosted.org/packages/66/74/a600aaf7cd39957fd1510adeedb1749c1e7eb82bd632a1153d9c664c3135/xxhash-3.8.1-cp314-cp314t-many
linux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:c73b6f652f0745425aa6378319c331293b5341756262e9408ed3d45f183375e6"
, size = 252243, upload-time = "2026-07-06T10:47:30.288Z" },
    { url = "https://files.pythonhosted.org/packages/ad/04/78d88fa75a6763e5d09bf1b947a392a27988903381b219006f92f3c68fc8/xxhash-3.8.1-cp314-cp314t-many
linux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:f6114692261eff4266386cdec0f7d87eee24e317ab397c218b7ae6a76b4c6339", size
 = 482191, upload-time = "2026-07-06T10:47:32.45Z" },
    { url = "https://files.pythonhosted.org/packages/7f/06/07a8aea1108d682de8791ce608cdf367d75ff4e7e57cd3c154bdc6f47b23/xxhash-3.8.1-cp314-cp314t-many
linux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:4df57c0b161ec1b3ed0526a67b0db0914b557e86ee8aae51887aec941b261542", s
ize = 226877, upload-time = "2026-07-06T10:47:34.705Z" },
    { url = "https://files.pythonhosted.org/packages/ed/b5/86bade5618a524d2c06c4041aa2fe8e5749ce16e88afba60d67c1684a21f/xxhash-3.8.1-cp314-cp314t-many
linux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:9043877a917be88ccf230aa5667c1bd059bce80f4c2727e4defa1b29b7f48b08", size = 319794, uploa
d-time = "2026-07-06T10:47:37.08Z" },
    { url = "https://files.pythonhosted.org/packages/23/69/9b1a2b89b1621bb740fbcb7beb512f60f99480c1bdc680c0c90e1f56ff75/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_aarch64.whl", hash = "sha256:559e3cabe522231909f9de98ef06929edbd53782046bd21aae0c72db6f2a0775", size = 246202, upload-time = "2026-07-06T10:
47:39.676Z" },
    { url = "https://files.pythonhosted.org/packages/08/ea/662ed6cb49f1d34078b6a3a3e0f3d29ff93fd7b5a03c0bc9ecfd9b2159c3/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_armv7l.whl", hash = "sha256:264710bd335016f303763ce1275c6486df30bb57c2245c91b224c983d7ac39b8", size = 275628, upload-time = "2026-07-06T10:4
7:41.99Z" },
    { url = "https://files.pythonhosted.org/packages/13/f5/49fc9e4c6728a5a3bd8fe639199d2fa67609b3a84f938aff6e8568dd3e4f/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_i686.whl", hash = "sha256:e14800b9b10bb39d7a60ad4a310e403164d7b8988a27ae933d4e40618a44088e", size = 231390, upload-time = "2026-07-06T10:47:
44.233Z" },
    { url = "https://files.pythonhosted.org/packages/64/9d/3acaf8f599c0e0b30e910a3a11ba32929da53c86dc73c7c55fe6a010b4e9/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_ppc64le.whl", hash = "sha256:ea6a3e734b0fd41b82784a400be946821900daebe610c050a5e0760838a34f99", size = 250600, upload-time = "2026-07-06T10:
47:47.611Z" },
    { url = "https://files.pythonhosted.org/packages/23/64/8acab4c5ec60dbe664b5b9858fd44c2413b07e535b09556a0a5022e78aa6/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_riscv64.whl", hash = "sha256:cf399fac542a1c7a4734a435b93df2c55e858c7d31abf6c1bdf46f9ae67fbfd0", size = 310032, upload-time = "2026-07-06T10:
47:49.88Z" },
    { url = "https://files.pythonhosted.org/packages/56/47/a0288d7329b1fe63e2734a32d19d444a96ae2b4810f545bc61e561224917/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_s390x.whl", hash = "sha256:44c89d915a75c11d2547eaee9098fcd80398987c4bff2974a0497a925bf92c07", size = 448882, upload-time = "2026-07-06T10:47
:52.631Z" },
    { url = "https://files.pythonhosted.org/packages/01/e7/3071dfd3beb5c38204ce1cf56bf7749fce08de900fa92714b81d1d8ca1f2/xxhash-3.8.1-cp314-cp314t-musl
linux_1_2_x86_64.whl", hash = "sha256:358650d5bda9c635da699c53adf4e8134af492ecc79c960f917eebf088bb6799", size = 223728, upload-time = "2026-07-06T10:4
7:55.093Z" },
    { url = "https://files.pythonhosted.org/packages/12/11/b99949f0ba2b07e9f9ffe83b9c86faa685f9080725dc21a916a607313be5/xxhash-3.8.1-cp314-cp314t-win3
2.whl", hash = "sha256:c240939e963653054fc7e4a17c382829cda4aa88a7daf0af841715dbded1b497", size = 33150, upload-time = "2026-07-06T10:47:57.274Z" },
    { url = "https://files.pythonhosted.org/packages/54/1c/09703eb341f8416e74e58d6c6732d4b5c46de59c942363203cb237cc95b0/xxhash-3.8.1-cp314-cp314t-win_
amd64.whl", hash = "sha256:7258ee276e8772599bc19e14b36f6260306e21b637190cd7cb489a2449d48684", size = 34005, upload-time = "2026-07-06T10:47:59.434Z" }
,
    { url = "https://files.pythonhosted.org/packages/d6/f9/6ed7251bb6a8af10ac73b1821c60583d2826e5b2064e45a979c935287c98/xxhash-3.8.1-cp314-cp314t-win_
arm64.whl", hash = "sha256:8f454166c2ffed45636c8d501741e649851ba2f346c4eb73a64c07ac00428f20", size = 30239, upload-time = "2026-07-06T10:48:01.874Z" }
,
]

[[package]]
name = "yarl"
version = "1.24.5"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "idna" },
    { name = "multidict" },
    { name = "propcache" },
]
sdist = { url = "https://files.pythonhosted.org/packages/31/33/ebe9e3d1f86c7a0b51094c0a146392045ca1631d2664889539dec8088a33/yarl-1.24.5.tar.gz", hash
= "sha256:e81b83143bee16329c23db3c1b2d82b29892fcbcb849186d2f6e98a5abe9a57f", size = 228679, upload-time = "2026-07-20T02:07:45.435Z" }
```

```
wheels = [
    { url = "https://files.pythonhosted.org/packages/1b/84/71d051c850b5af41d168c679d9eb67eb7c55283ac4ee131673edf134bc4e/yarl-1.24.5-cp312-cp312-macosx_10_13_universal2.whl", hash = "sha256:d693396e5aea78db03decd60aec9ece16c9b40ba00a587f089615ff4e718a81d", size = 136035, upload-time = "2026-07-20T02:05:25.489Z" },
    { url = "https://files.pythonhosted.org/packages/03/4d/8ad27f9a1b7e69313cca5d695b925b48efe51208d3490e0844bae97cabc0/yarl-1.24.5-cp312-cp312-macosx_10_13_x86_64.whl", hash = "sha256:3363fcc96e665878946ad7a106b9a13eac0541766a690ef287c0232ac768b6ec", size = 97642, upload-time = "2026-07-20T02:05:27.429Z" },
    { url = "https://files.pythonhosted.org/packages/ea/b4/05b4131c407006cd1e410e9c6539f16a0945724677e5364447313c15ea3e/yarl-1.24.5-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:9d399bdcfb4a0f659b9b3788bbc89babe63d9a6a65aacdf4d4e7065ff2e6316c", size = 97323, upload-time = "2026-07-20T02:05:29.441Z" },
    { url = "https://files.pythonhosted.org/packages/20/16/e618c875c73e0e39611f20a581b3d5e8d59b8857bf001bee3263044c6deb/yarl-1.24.5-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:90333fd89b43c0d08ac85f3f1447593fc2c66de18c3d6378d7125ea118dc7a54", size = 107741, upload-time = "2026-07-20T02:05:31.367Z" },
    { url = "https://files.pythonhosted.org/packages/d9/9a/c4defeaf3ed33fcb346aacf9c6e971a8d4e2bde04a0310e79abb208e7965/yarl-1.24.5-cp312-cp312-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:665b0a2c463cc9423dd647e0bfd9f4ccc9b50f768c55304d5e9f80b177c1de12", size = 103570, upload-time = "2026-07-20T02:05:33.303Z" },
    { url = "https://files.pythonhosted.org/packages/5f/e7/0e0e0de5865ebd5914537ef486f36c727a59865c3ac0cf5ff1b32aececbf/yarl-1.24.5-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:e006d3a974c4ee19512e5f058abedb6eef36a5e553c14812bdeba1758d812e6d", size = 115815, upload-time = "2026-07-20T02:05:35.292Z" },
    { url = "https://files.pythonhosted.org/packages/2b/27/ca56b700cb170aba25a3893b75355b213935657dc5714d2383354a270e62/yarl-1.24.5-cp312-cp312-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:e7d42c53124345 0ef0d4d9c172e7ed6ef052640f195629065041b5add4e058d1", size = 116025, upload-time = "2026-07-20T02:05:37.503Z" },
    { url = "https://files.pythonhosted.org/packages/d6/d0/d56c859b8222116f5d68459199f48359e0bf121b6f65a69bf329b3602ba0/yarl-1.24.5-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:f08c7513ecef5aad65687bfdf6bc601ae9fccd04a42904501f8f7141abad9eb9", size = 109835, upload-time = "2026-07-20T02:05:39.506Z" },
    { url = "https://files.pythonhosted.org/packages/70/a2/3a35557e4d1a79425040eba202ccaf08bdc8717680fc77e2498a1ad2e0a5/yarl-1.24.5-cp312-cp312-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:6c95b17fe34ed802f17e205112e6e10db92275c34fee290aa9bdc55a9c724027", size = 108884, upload-time = "2026-07-20T02:05:41.584Z" },
    { url = "https://files.pythonhosted.org/packages/e4/35/ef4c26356b7913c68983bac2d72a4212b3347af551cb8d250b99b5ed7b7f/yarl-1.24.5-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:56b149b22de33b23b0c6077ab9518c6dcb538ad462e1830e68d06591ccf6e38b", size = 107308, upload-time = "2026-07-20T02:05:43.697Z" },
    { url = "https://files.pythonhosted.org/packages/d5/91/ff0dc66c2ccf3e0153ab97ff61eabab4400e6a5264af427ab30cd69f1857/yarl-1.24.5-cp312-cp312-musllinux_1_2_armv7l.whl", hash = "sha256:a8fe66b8f300da93798025a785a5b90b42f3810dc2b72283ff84a41aaaebc293", size = 103646, upload-time = "2026-07-20T02:05:45.895Z" },
    { url = "https://files.pythonhosted.org/packages/74/f0/33b9271c7f881766359d58266fa0811d2e5210ed860e28da7dc6d7786344/yarl-1.24.5-cp312-cp312-musllinux_1_2_ppc64le.whl", hash = "sha256:377fe3732edbaf78ee74efdf2c9f49f6e99f20e7f9d2649fda3eb4badd77d76e", size = 115305, upload-time = "2026-07-20T02:05:47.832Z" },
    { url = "https://files.pythonhosted.org/packages/ef/65/fd79fb1868c4a80db8661091de525bf430f63c3bea1b20e8b6a84fc7d359/yarl-1.24.5-cp312-cp312-musllinux_1_2_riscv64.whl", hash = "sha256:e8ffa78582120024f476a611d7befc123cee59e47e8309d470cf667d806e613b", size = 108404, upload-time = "2026-07-20T02:05:49.604Z" },
    { url = "https://files.pythonhosted.org/packages/ff/ba/dbabe6b262f17a816c70cfc09558dbf03ece3ec76684d02f911a3d3a189c/yarl-1.24.5-cp312-cp312-musllinux_1_2_s390x.whl", hash = "sha256:daba5e594f06114e37db186efd2dd916609071e59daca901a0a2e71f02b142ce", size = 115940, upload-time = "2026-07-20T02:05:51.741Z" },
    { url = "https://files.pythonhosted.org/packages/a5/43/fab2d1dad9d340a268cdde63756a123d069723efff6a372d123fa74a9517/yarl-1.24.5-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:65be18ec59496c13908f02a2472751d9ef840b4f3fb5726f129306bf6a2a7bba", size = 110006, upload-time = "2026-07-20T02:05:53.554Z" },
    { url = "https://files.pythonhosted.org/packages/c4/27/41eb51bbd1b8d89546b83897cfb0164f1e109304fd408dbb151b639eec0f/yarl-1.24.5-cp312-cp312-win_amd64.whl", hash = "sha256:a929d878fec099030c292803b31e5d5540a7b6a31e6a3cc76cb4685fc2a2f51b", size = 97618, upload-time = "2026-07-20T02:05:55.57Z" },
    { url = "https://files.pythonhosted.org/packages/3c/25/b2553764b3d65db711d8f45416351ec4f420847558eb669edcbcaadf5780/yarl-1.24.5-cp312-cp312-win_arm64.whl", hash = "sha256:7ce27823052e2013b597e0c738b13e7e36b8ccb9400df8959417b052ab0fd92c", size = 93018, upload-time = "2026-07-20T02:05:57.554Z" },
    { url = "https://files.pythonhosted.org/packages/e1/63/64ef361967cc983573149dc1515d531db5da8a4c92d22bb833d59e01b313/yarl-1.24.5-cp313-cp313-macosx_10_13_universal2.whl", hash = "sha256:79af890482fc94648e8cde4c68620378f7fef60932710fa17a66abc039244da2", size = 135075, upload-time = "2026-07-20T02:05:59.671Z" },
    { url = "https://files.pythonhosted.org/packages/bb/89/55920fd853ce43e608adbc3962456f0d649d6bb15250dc2988321da0fe1c/yarl-1.24.5-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:46c2f213e23a04b93a392942d782eb9e413e6ef6bf7c8c53884e599a5c174dcb", size = 97225, upload-time = "2026-07-20T02:06:01.769Z" },
    { url = "https://files.pythonhosted.org/packages/15/f0/7688d3f2cfff7590df2af38ec46d969f4281a4dddb08a9ad2eafbcdddf98/yarl-1.24.5-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:92ab3e11448f2ff7bf53c5a26eff0edc086898ec8b21fb154b85839ce1d88075", size = 96751, upload-time = "2026-07-20T02:06:03.676Z" },
    { url = "https://files.pythonhosted.org/packages/05/1a/a851a0f94aaaf379dd4f901bfc80f634280bec51eb260b47363e2a4cd62e/yarl-1.24.5-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:ebb0ec7f17803063d5aeb982f3b1bd2b2f4e4fae6751226cbd6ba1fcfe9e63ff", size = 107960, upload-time = "2026-07-20T02:06:05.699Z" },
    { url = "https://files.pythonhosted.org/packages/6c/a8/faea066c12f9c77ca0de90641f1655f9dd7b412477bf28c76d692f3aecff/yarl-1.24.5-cp313-cp313-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:82632daed195dcc8ea664e8556dc9bdbd671960fb3776bd92806ce05792c2448", size = 103500, upload-time = "2026-07-20T02:06:07.556Z" },
    { url = "https://files.pythonhosted.org/packages/fb/9c/1e67084c2a6e2f2db0e3be798328cb3be42c0119b621d25461479a224d21/yarl-1.24.5-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:53e549287ef628fecba270045c9701b0c564563a9b0577d24a4ec75b8ab8040f", size = 115780, upload-time = "2026-07-20T02:06:09.599Z" },
    { url = "https://files.pythonhosted.org/packages/58/86/1f94664e147474337e3359f52012cf3d02f825f694317b178bfba1078c62/yarl-1.24.5-cp313-cp313-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:fcd3b77e2f17bbe4ca56ec7bcb07992647d19d0b9c05d84886dcd6f9eb810afd", size = 115308, upload-time = "2026-07-20T02:06:11.352Z" },
    { url = "https://files.pythonhosted.org/packages/0a/43/8e55ae7538ba5f28ccb3c845c6dd4549cf7016d5992e5326512519107cdd/yarl-1.24.5-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:d46b86567dd4e248c6c159fcbcdcce01e0a5c8a7cd2334a0fff759d0fa075b16", size = 110574, upload-time = "2026-07-20T02:06:13.129Z" },
    { url = "https://files.pythonhosted.org/packages/ce/ba/a889ec8765cedcf2ac44dcb02d6a21e4861399b243b263c5f2dde27ee740/yarl-1.24.5-cp313-cp313-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:7f72c74aa99359e27a2ee8d6613fefa28b5f76a983c083074dfc2aaa4ab46213", size = 109914, upload-time = "2026-07-20T02:06:15.243Z" },
    { url = "https://files.pythonhosted.org/packages/9c/c3/e45f821af67b791c2dbbe4a9f4137a1d33f8d386654a05a0c3f47bdfa25d/yarl-1.24.5-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:3f45789ce415a7ec0820dc4f82925f9b5f7732070be1dec1f5f23ec381435a24", size = 107712, upload-time = "2026-07-20T02:06:17.443Z" },
    { url = "https://files.pythonhosted.org/packages/02/00/2ab0f42c9857fcb490bfaa6647b14540b53d241ab209f23220b958cc5832/yarl-1.24.5-cp313-cp313-musllinux_1_2_armv7l.whl", hash = "sha256:6e73e7fe93f17a7b191f52ec9da9dd8c06a8fe735a1ecbd13b97d1c723bff385", size = 104251, upload-time = "2026-07-20T02:06:19.259Z" },
    { url = "https://files.pythonhosted.org/packages/7a/70/709d9a286e98af2c7fd8e4e6cada658b5c0e30d87dd7e2a63c2fb5767217/yarl-1.24.5-cp313-cp313-musllinux_1_2_ppc64le.whl", hash = "sha256:4a36f9becdd4c5c52a20c3e9484128b070b1dcfc8944c006f3a528295a359a9c", size = 115319, upload-time = "2026-07-20T02:06:21.207Z" },
    { url = "https://files.pythonhosted.org/packages/5c/6c/3eaa515142991fe84cfc483ff986492211f1978f90161ccefdbec919d09b/yarl-1.24.5-cp313-cp313-musllinux_1_2_riscv64.whl", hash = "sha256:7bcbe0fcf850eae67b6b01749815a4f7161c560a844c769ad7b48fcd99f791c4", size = 109163, upload-time = "2026-07-20T02:06:23.006Z" },
    { url = "https://files.pythonhosted.org/packages/bb/64/711dafce66c323a3144d470547a71c5384c57623308ac8bb5e4b903ac148/yarl-1.24.5-cp313-cp313-musllinux_1_2_s390x.whl", hash = "sha256:24e861e9630e0daddcb9191fb187f60f034e17a4426f8101279f0c475cd74144", size = 115435, upload-time = "2026-07-20T02:06:24.923Z" },
    { url = "https://files.pythonhosted.org/packages/cf/f3/9b9d0e6d84bea851eb1ba99e4bdc755b86fd813e49ec86dfe42f26befdef/yarl-1.24.5-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:9335a099ad87287c37fe5d1a982ff392fa5efe5d14b40a730b1ec1d6a41382b4", size = 110691, upload-time = "2026-07-20T02:06:26.973Z" },
    { url = "https://files.pythonhosted.org/packages/86/e4/62a06b7e87c4246ac76b7c2da136f972eb4a3a1fc94abb07e7022d6fdb0a/yarl-1.24.5-cp313-cp313-win_amd64.whl", hash = "sha256:2dbe06fc16bc91502bca713704022182e5729861ae00277c3a23354b40929740", size = 97454, upload-time = "2026-07-20T02:06:29.163Z" },
    { url = "https://files.pythonhosted.org/packages/9e/c9/5fc8025b318ab10db413b61056bd0d95c557a70e8df4210c7511f866329c/yarl-1.24.5-cp313-cp313-win_arm64.whl", hash = "sha256:6b8536851f9f65e7f00c7a1d49ba7f2be0ffe2c11555367fc9f50d9f842410a1", size = 92813, upload-time = "2026-07-20T02:06:31.113Z" },
    { url = "https://files.pythonhosted.org/packages/a9/08/5f3085fef9564217074db9dd8573de1795bc82cde61a7ad10b6a7234a569/yarl-1.24.5-cp314-cp314-macosx_10_15_universal2.whl", hash = "sha256:2729fcfc4f6a596fb0c50f32090400aa9367774ac296a00387e65098c0befa76", size = 135680, upload-time = "2026-07-20T02:06:33.273Z" },
    { url = "https://files.pythonhosted.org/packages/98/35/ba9436e579bd48a8801f2021d842d9ab4994c26e4c7dd3a4c1f1bcb57a9e/yarl-1.24.5-cp314-cp314-macosx_10_15_x86_64.whl", hash = "sha256:ff330d3c30db4eb6b01d79e29d2d0b407a7ecad39cfd9ec993ece57396a2ec0d", size = 97395, upload-time = "2026-07-20T02:06:35.259Z" },
```

```
    { url = "https://files.pythonhosted.org/packages/18/a9/a07f76f3c44e02b25cc743af5ef93eef27f7013eadca770451b6a6ccb5db/yarl-1.24.5-cp314-cp314-macosx_11_0_arm64.whl", hash = "sha256:e42d75862735da90e7fc5a7b23db0c976f737113a54b3c9777a9b665e9cbff75", size = 97223, upload-time = "2026-07-20T02:06:37.216Z" },
    { url = "https://files.pythonhosted.org/packages/77/f7/a9a1d6fa7dd9e388f95b30f6ad3ec4e285f6c8f61f44ce16070c3fcfe414/yarl-1.24.5-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:a3732e66413163e72508da9eff9ce9d2846fde51fae45d3605393d3e6cd303e9", size = 108777, upload-time = "2026-07-20T02:06:39.292Z" },
    { url = "https://files.pythonhosted.org/packages/2f/44/e0b86c302471fabd6f02808ecf2ac52b8412b624787849d4bf2cdb466f6f/yarl-1.24.5-cp314-cp314-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:5b8ee53be440a0cffc991a27be3057e0530122548dbe7c0892df08822fce5ede", size = 103119, upload-time = "2026-07-20T02:06:41.456Z" },
    { url = "https://files.pythonhosted.org/packages/d1/16/9c16d180bf8faaf223225eb50e1245870ff1ae0e302a27153988e65c51fd/yarl-1.24.5-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:af3aefa655adb5869491fa907e652290386800ae99cc50095cba71e2c6aefdca", size = 116471, upload-time = "2026-07-20T02:06:43.696Z" },
    { url = "https://files.pythonhosted.org/packages/d2/8d/b219b9df28a02ce95cfbdd41d2f7caa5669d0ff979c1c9975697145e33c5/yarl-1.24.5-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:2120b96872df4a117cde97d270bac96aea7cc52205d305cf4611df694a487027", size = 115974, upload-time = "2026-07-20T02:06:45.874Z" },
    { url = "https://files.pythonhosted.org/packages/9b/e8/f20557aca240d88e69850ad1ee91756821d094bb1310565c04d25c6682a2/yarl-1.24.5-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:66410eb6345d467151934b49bfa70fb32f5b35a6140baa40ad97d6436abea2e9", size = 110830, upload-time = "2026-07-20T02:06:47.852Z" },
    { url = "https://files.pythonhosted.org/packages/db/18/199b85109a53eeca64ee19c9cca228287e8e4ab0cc1a09b28f530e65cce0/yarl-1.24.5-cp314-cp314-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:4af7b7e1be0a69bee8210735fe6dcfc38879adfac6d62e789d53ba432d1ffa41", size = 110054, upload-time = "2026-07-20T02:06:49.84Z" },
    { url = "https://files.pythonhosted.org/packages/aa/2f/ed28147f8cd7f48c49367c90713b30a555284b6105a6a56f3a05568da795/yarl-1.24.5-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:fa139875ff98ab97da323cfadfaff08900d1ad42f1b5087b0b812a55c5a06373", size = 108312, upload-time = "2026-07-20T02:06:51.835Z" },
    { url = "https://files.pythonhosted.org/packages/c5/c5/55e16ae0a5c227cea8df1c6871ba57d614a34243146c05729caf2a1bd9c5/yarl-1.24.5-cp314-cp314-musllinux_1_2_armv7l.whl", hash = "sha256:0055afc45e864b92729ac7600e2d102c17bef060647e74bca75fa84d66b9ff36", size = 103662, upload-time = "2026-07-20T02:06:54.061Z" },
    { url = "https://files.pythonhosted.org/packages/8d/ea/dbd7c2caec459c9a426f18b02688ecbfb58620d0f6a3422d24769fbaf8ab/yarl-1.24.5-cp314-cp314-musllinux_1_2_ppc64le.whl", hash = "sha256:f0e466ed7511fe9d459a819edbc6c2585c0b6eabde9fa8a8947552468a7a6ef0", size = 116090, upload-time = "2026-07-20T02:06:56.015Z" },
    { url = "https://files.pythonhosted.org/packages/06/84/39ce4ce3059e07fece5fbdbee8c4053406af9aca911ce9fa5f8548aab6af/yarl-1.24.5-cp314-cp314-musllinux_1_2_riscv64.whl", hash = "sha256:f141474e85b7e54998ec5180530a7cda99ab29e282fa50e0756d89981a9b43c5", size = 109523, upload-time = "2026-07-20T02:06:57.926Z" },
    { url = "https://files.pythonhosted.org/packages/a9/8b/71ff44137b405c64a7788075669c24010019f57a7464b78c3a6cbee539d9/yarl-1.24.5-cp314-cp314-musllinux_1_2_s390x.whl", hash = "sha256:e2935f8c39e3b03e83519292d78f075189978f3f4adc15a78144c7c8e2a1cba5", size = 116084, upload-time = "2026-07-20T02:06:59.868Z" },
    { url = "https://files.pythonhosted.org/packages/62/c0/423078fdd4042e1862c11f0ffd977a0ffa393783c12bee94685923bc189e/yarl-1.24.5-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:9d1216a7f6f77836617dba35687c5b78a4170afc3c3f18fc788f785ba26565c4", size = 111006, upload-time = "2026-07-20T02:07:01.907Z" },
    { url = "https://files.pythonhosted.org/packages/cf/52/6daa2ee9d95e5c98b8128f8df91eb692eb423ab274b8cf08db52152fad26/yarl-1.24.5-cp314-cp314-win_amd64.whl", hash = "sha256:5ba4f78df2bcc19f764a4b26a8a4f5049c110090ad5825993aacb052bf8003ad", size = 99215, upload-time = "2026-07-20T02:07:03.852Z" },
    { url = "https://files.pythonhosted.org/packages/ec/0e/464a847d7359e0da75dd9fc5c1d1aa35d0159ea31e5f8e66a3c1c29ff3d0/yarl-1.24.5-cp314-cp314-win_arm64.whl", hash = "sha256:9e4e16c73d717c5cf27626c524d0a2e261ad20e46932b2670f64ad5dde23e26f", size = 94566, upload-time = "2026-07-20T02:07:06.074Z" },
    { url = "https://files.pythonhosted.org/packages/e2/55/e03acc4446772660bc335e86e41ef31e4d0d838fd641531a11a5ee33b493/yarl-1.24.5-cp314-cp314t-macosx_10_15_universal2.whl", hash = "sha256:e1ae548a9d901adca07899a4147a7c826bbcc06239d3ce9a59f57886a28a4c88", size = 142533, upload-time = "2026-07-20T02:07:08.284Z" },
    { url = "https://files.pythonhosted.org/packages/ae/71/4acd3a1fc7cf14345cdb302665ecd2097f62c365b4f14ca17d4f37775cf9/yarl-1.24.5-cp314-cp314t-macosx_10_15_x86_64.whl", hash = "sha256:ff405d91509d88e8d44129cd87b18d70acd1f0c1aeabd7bc3c46792b1fe2acba", size = 100776, upload-time = "2026-07-20T02:07:10.197Z" },
    { url = "https://files.pythonhosted.org/packages/ff/0b/cfb76b7fe99686db264bff829779a539d923e7564ffd7ef18da6c54c3774/yarl-1.24.5-cp314-cp314t-macosx_11_0_arm64.whl", hash = "sha256:47e98aab9d8d82ff682e7b0b5dded33bf138a32b817fcf7fa3b27b2d7c412928", size = 100913, upload-time = "2026-07-20T02:07:12.357Z" },
    { url = "https://files.pythonhosted.org/packages/8b/3f/7116e782992abbd4fb6948488aec72078895e929a23078290739e8396fce/yarl-1.24.5-cp314-cp314t-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:f0a658a6d3fafee5c6f63c58f3e785c8c43c93fbc02bf9f2b6663f8185e0971f", size = 106507, upload-time = "2026-07-20T02:07:14.173Z" },
    { url = "https://files.pythonhosted.org/packages/33/90/d4d2d73ee78229cc889872eb8e085d8f5c6f51abdb178409fd9b23cf74fd/yarl-1.24.5-cp314-cp314t-manylinux2014_armv7l.manylinux_2_17_armv7l.manylinux_2_31_armv7l.whl", hash = "sha256:4377407001ca3c057773f44d8ddd6358fa5f691407c1ba92210bd3cf8d9e4c95", size = 99219, upload-time = "2026-07-20T02:07:16.019Z" },
    { url = "https://files.pythonhosted.org/packages/3e/fa/a6df1a9bccd644eec00abee0dff4277416222cec435330fd1f2858523ec1/yarl-1.24.5-cp314-cp314t-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:7c0494a31a1ac5461a226e7947a9c9b78c44e1dc7185164fa7e9651557a5d9bc", size = 111804, upload-time = "2026-07-20T02:07:18.141Z" },
    { url = "https://files.pythonhosted.org/packages/8a/9e/7b2a1f4bcc20e9447156dd2b1c4d01f70d9df0759025ee7d09a84ffae134/yarl-1.24.5-cp314-cp314t-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:a7cff474ab7cd149765bb784cf6d78b32e18e20473fb7bda860bce98ab58e9da", size = 110943, upload-time = "2026-07-20T02:07:20.06Z" },
    { url = "https://files.pythonhosted.org/packages/08/ff/22c92affb0f9b623ca753d27d968b5625b868f12c6378d049d55ae247643/yarl-1.24.5-cp314-cp314t-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:cbb833ccacdb5519eff9b8b71ee618cc2801c878e77e288775d77c3a2ced858a", size = 108251, upload-time = "2026-07-20T02:07:22.217Z" },
    { url = "https://files.pythonhosted.org/packages/45/44/5769b96298c1e195fb412997b6090af2a84105cf59c17613558a2d011d1f/yarl-1.24.5-cp314-cp314t-manylinux_2_31_riscv64.manylinux_2_39_riscv64.whl", hash = "sha256:82f75e05912e84b7a0fe57075d9c59de3cb352b928330f2eb69b2e1f54c3e1f0", size = 106025, upload-time = "2026-07-20T02:07:24.083Z" },
    { url = "https://files.pythonhosted.org/packages/4c/40/009e8e791fd9762c0e1567e69248acb4f49064597e1680874c16dd8bb798/yarl-1.24.5-cp314-cp314t-musllinux_1_2_aarch64.whl", hash = "sha256:16a2f5010280020e90f5330257e6944bc33e73593b136cc5a241e6c1dc292498", size = 106573, upload-time = "2026-07-20T02:07:26.248Z" },
    { url = "https://files.pythonhosted.org/packages/20/c6/b7480578f8a0a80946f36ad6df547ecec704f9ba69d2de60f8aa6f1c1cbf/yarl-1.24.5-cp314-cp314t-musllinux_1_2_armv7l.whl", hash = "sha256:ffcd54362564dc1a30fb74d8b8a6e5a6b11ebd5e27266adc3b7427a21a6c9104", size = 100751, upload-time = "2026-07-20T02:07:28.098Z" },
    { url = "https://files.pythonhosted.org/packages/d4/27/4476f3360b91a48c5cf125e91f59a3bd35299d84a431a258d57f5977bb11/yarl-1.24.5-cp314-cp314t-musllinux_1_2_ppc64le.whl", hash = "sha256:0465ec8cedc2349b97a6b595ace64084a50c6e839eca40aa0626f38b8350e331", size = 111643, upload-time = "2026-07-20T02:07:30.88Z" },
    { url = "https://files.pythonhosted.org/packages/4c/4b/5cdd3e5ee944e8af31e52f6cd3d3af5fd7b937e036ccbbba2c9ffebede95/yarl-1.24.5-cp314-cp314t-musllinux_1_2_riscv64.whl", hash = "sha256:4db9aecb141cb7a5447171b57aa1ed3a8fee06af40b992ffc31206c0b0121550", size = 106312, upload-time = "2026-07-20T02:07:33.06Z" },
    { url = "https://files.pythonhosted.org/packages/18/86/f406b0c2a6f99575de2da671ef47aa06f89a5be83a27a46971c3b86cecdb/yarl-1.24.5-cp314-cp314t-musllinux_1_2_s390x.whl", hash = "sha256:f540c013589084679a6c7fac07096b10159737918174f5dfc5e11bf5bca4dfe6", size = 110379, upload-time = "2026-07-20T02:07:35.155Z" },
    { url = "https://files.pythonhosted.org/packages/f0/6c/9f3adfbd3b30b4fa0f7ccb3a83eba2c1152d3fff554d535e640ba0f7ba2b/yarl-1.24.5-cp314-cp314t-musllinux_1_2_x86_64.whl", hash = "sha256:a61834fb15d81322d872eaafd333838ae7c9cea84067f232656f75965933d047", size = 108497, upload-time = "2026-07-20T02:07:37.35Z" },
    { url = "https://files.pythonhosted.org/packages/dd/37/91eb2e5ca883a529c1b390348a74cd9fc0512171727f547ce70bfe02be5c/yarl-1.24.5-cp314-cp314t-win_amd64.whl", hash = "sha256:5c88e5815a49d289e599f3513aa7fde0bc2092ff188f99c940f007f90f53d104", size = 102450, upload-time = "2026-07-20T02:07:39.578Z" },
    { url = "https://files.pythonhosted.org/packages/bf/f4/ed5c402ac8fde4403ed3366c2716bfddc8a6677ebd59f3d62772cc7fe468/yarl-1.24.5-cp314-cp314t-win_arm64.whl", hash = "sha256:cf139c02f5f23ef6532040a30ff662c00a318c952334f211046b8e60b7f17688", size = 97222, upload-time = "2026-07-20T02:07:41.55Z" },
    { url = "https://files.pythonhosted.org/packages/61/02/962c1cbfc401a30c1d034dc67ff395f64b52302c6d62de556c1fca99acc0/yarl-1.24.5-py3-none-any.whl", hash = "sha256:a33700d13d9b7d84fd10947b09ff69fb9a792e519c8cb9764a3ca70baa6c23a7", size = 58612, upload-time = "2026-07-20T02:07:43.461Z" },
]

[[package]]
name = "zstandard"
version = "0.25.0"
source = { registry = "https://pypi.org/simple" }
sdist = { url = "https://files.pythonhosted.org/packages/fd/aa/3e0508d5a5dd96529cdc5a97011299056e14c6505b678fd58938792794b1/zstandard-0.25.0.tar.gz", hash = "sha256:7713e1179d162cf5c7906da876ec2ccb9c3a9dcbdffef0cc7f70c3667a205f0b", size = 711513, upload-time = "2025-09-14T22:15:54.002Z" }
wheels = [
    { url = "https://files.pythonhosted.org/packages/82/fc/f26eb6ef91ae723a03e16eddb198abcfce2bc5a42e224d44cc8b6765e57e/zstandard-0.25.0-cp312-cp312-m
```

acosx_10_13_x86_64.whl", hash = "sha256:7b3c3a3ab9daa3eed242d6ecceead93aebbb8f5f84318d82cee643e019c4b73b", size = 795738, upload-time = "2025-09-14T22:16:56.237Z" },
    { url = "https://files.pythonhosted.org/packages/aa/1c/d920d64b22f8dd028a8b90e2d756e431a5d86194caa78e3819c7bf53b4b3/zstandard-0.25.0-cp312-cp312-macosx_11_0_arm64.whl", hash = "sha256:913cbd31a400febff93b564a23e17c3ed2d56c064006f54efec210d586171c00", size = 640436, upload-time = "2025-09-14T22:16:57.774Z" },
    { url = "https://files.pythonhosted.org/packages/53/6c/288c3f0bd9fcfe9ca41e2c2fbfd17b2097f6af57b62a81161941f09afa76/zstandard-0.25.0-cp312-cp312-manylinux2010_i686.manylinux2014_i686.manylinux_2_12_i686.manylinux_2_17_i686.whl", hash = "sha256:011d388c76b11a0c165374ce660ce2c8efa8e5d87f34996aa80f9c0816698b64", size = 5343019, upload-time = "2025-09-14T22:16:59.302Z" },
    { url = "https://files.pythonhosted.org/packages/1e/15/efef5a2f204a64bdb5571e6161d49f7ef0fffdbca953a615efbec045f60f/zstandard-0.25.0-cp312-cp312-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:6dffecc361d079bb48d7caef5d673c88c8988d3d33fb74ab95b7ee6da42652ea", size = 5063012, upload-time = "2025-09-14T22:17:01.156Z" },
    { url = "https://files.pythonhosted.org/packages/b7/37/a6ce629ffdb43959e92e87ebdaeebb5ac81c944b6a75c9c47e300f85abdf/zstandard-0.25.0-cp312-cp312-manylinux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:7149623bba7fdf7e7f24312953bcf73cae103db8cae49f8154dd1eadc8a29ecb", size = 5394148, upload-time = "2025-09-14T22:17:03.091Z" },
    { url = "https://files.pythonhosted.org/packages/e3/79/2bf870b3abeb5c070fe2d670a5a8d1057a8270f125ef7676d29ea900f496/zstandard-0.25.0-cp312-cp312-manylinux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:6a573a35693e03cf1d67799fd01b50ff578515a8aeadd4595d2a7fa9f3ec002a", size = 5451652, upload-time = "2025-09-14T22:17:04.979Z" },
    { url = "https://files.pythonhosted.org/packages/53/60/7be26e610767316c028a2cbedb9a3beabdbe33e2182c373f71a1c0b88f36/zstandard-0.25.0-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:5a56ba0db2d244117ed744dfa8f6f5b366e14148e00de44723413b2f3938a902", size = 5546993, upload-time = "2025-09-14T22:17:06.781Z" },
    { url = "https://files.pythonhosted.org/packages/85/c7/3483ad9ff0662623f3648479b0380d2de5510abf00990468c286c6b04017/zstandard-0.25.0-cp312-cp312-musllinux_1_1_aarch64.whl", hash = "sha256:10ef2a79ab8e2974e2075fb984e5b9806c64134810fac21576f0668e7ea19f8f", size = 5046806, upload-time = "2025-09-14T22:17:08.415Z" },
    { url = "https://files.pythonhosted.org/packages/08/b3/206883dd25b8d1591a1caa44b54c2aad84badccf2f1de9e2d60a446f9a25/zstandard-0.25.0-cp312-cp312-musllinux_1_1_x86_64.whl", hash = "sha256:aaf21ba8fb76d102b696781bddaa0954b782536446083ae3fdaa6f16b25a1c4b", size = 5576659, upload-time = "2025-09-14T22:17:10.164Z" },
    { url = "https://files.pythonhosted.org/packages/9d/31/76c0779101453e6c117b0ff22565865c54f48f8bd807df2b00c2c404b8e0/zstandard-0.25.0-cp312-cp312-musllinux_1_2_aarch64.whl", hash = "sha256:1869da9571d5e94a85a5e8d57e4e8807b175c9e4a6294e3b66fa4efb074d90f6", size = 4953933, upload-time = "2025-09-14T22:17:11.857Z" },
    { url = "https://files.pythonhosted.org/packages/18/e1/97680c664a1bf9a247a280a053d98e251424af51f1b196c6d52f117c9720/zstandard-0.25.0-cp312-cp312-musllinux_1_2_i686.whl", hash = "sha256:809c5bcb2c67cd0ed81e9229d227d4ca28f82d0f778fc5fea624a9def3963f91", size = 5268008, upload-time = "2025-09-14T22:17:13.627Z" },
    { url = "https://files.pythonhosted.org/packages/1e/73/316e4010de585ac798e154e88fd81bb16afc5c5cb1a72eeb16dd37e8024a/zstandard-0.25.0-cp312-cp312-musllinux_1_2_ppc64le.whl", hash = "sha256:f27662e4f7dbf9f9c12391cb37b4c4c3cb90ffbd3b1fb9284dadbbb8935fa708", size = 5433517, upload-time = "2025-09-14T22:17:16.103Z" },
    { url = "https://files.pythonhosted.org/packages/5b/60/dd0f8cfa8129c5a0ce3ea6b7f70be5b33d2618013a161e1ff26c2b39787c/zstandard-0.25.0-cp312-cp312-musllinux_1_2_s390x.whl", hash = "sha256:99c0c846e6e61718715a3c9437ccc625de26593fea60189567f0118dc9db7512", size = 5814292, upload-time = "2025-09-14T22:17:17.827Z" },
    { url = "https://files.pythonhosted.org/packages/fc/5f/75aafd4b9d11b5407b641b8e41a57864097663699f23e9ad4dbb91dc6bfe/zstandard-0.25.0-cp312-cp312-musllinux_1_2_x86_64.whl", hash = "sha256:474d2596a2dbc241a556e965fb76002c1ce655445e4e3bf38e5477d413165ffa", size = 5360237, upload-time = "2025-09-14T22:17:19.954Z" },
    { url = "https://files.pythonhosted.org/packages/ff/8d/0309daffea4fcac7981021dbf21cdb2e3427a9e76bafbcdbdf5392ff99a4/zstandard-0.25.0-cp312-cp312-win32.whl", hash = "sha256:23ebc8f17a03133b4426bcc04aabd68f8236eb78c3760f12783385171b0fd8bd", size = 436922, upload-time = "2025-09-14T22:17:24.398Z" },
    { url = "https://files.pythonhosted.org/packages/79/3b/fa54d9015f945330510cb5d0b0501e8253c127cca7ebe8ba46a965df18c5/zstandard-0.25.0-cp312-cp312-win_amd64.whl", hash = "sha256:ffef5a74088f1e09947aecf91011136665152e0b4b359c42be3373897fb39b01", size = 506276, upload-time = "2025-09-14T22:17:21.429Z" },
    { url = "https://files.pythonhosted.org/packages/ea/6b/8b51697e5319b1f9ac71087b0af9a40d8a6288ff8025c36486e0c12abcc4/zstandard-0.25.0-cp312-cp312-win_arm64.whl", hash = "sha256:181eb40e0b6a29b3cd2849f825e0fa34397f649170673d385f3598ae17cca2e9", size = 462679, upload-time = "2025-09-14T22:17:23.147Z" },
    { url = "https://files.pythonhosted.org/packages/35/0b/8df9c4ad06af91d39e94fa96cc010a24ac4ef1378d3efab9223cc8593d40/zstandard-0.25.0-cp313-cp313-macosx_10_13_x86_64.whl", hash = "sha256:ec996f12524f88e151c339688c3897194821d7f03081ab35d31d1e12ec975e94", size = 795735, upload-time = "2025-09-14T22:17:26.042Z" },
    { url = "https://files.pythonhosted.org/packages/3f/06/9ae96a3e5dcfd119377ba33d4c42a7d89da1efabd5cb3e366b156c45ff4d/zstandard-0.25.0-cp313-cp313-macosx_11_0_arm64.whl", hash = "sha256:a1a4ae2dec3993a32247995bdfe367fc3266da832d82f8438c8570f989753de1", size = 640440, upload-time = "2025-09-14T22:17:27.366Z" },
    { url = "https://files.pythonhosted.org/packages/d9/14/933d27204c2bd404229c69f445862454dcc101cd69ef8c6068f15aaec12c/zstandard-0.25.0-cp313-cp313-manylinux2010_i686.manylinux2014_i686.manylinux_2_12_i686.manylinux_2_17_i686.whl", hash = "sha256:e96594a5537722fdfb79951672a2a63aec5ebfb823e7560586f7484819f2a08f", size = 5343070, upload-time = "2025-09-14T22:17:28.896Z" },
    { url = "https://files.pythonhosted.org/packages/6d/db/ddb11011826ed7db9d0e485d13df79b58586bfdec56e5c84a928a9a78c1c/zstandard-0.25.0-cp313-cp313-manylinux2014_aarch64.manylinux_2_17_aarch64.whl", hash = "sha256:bfc4e20784722098822e3eee42b8e576b379ed72cca4a7cb856ae733e62192ea", size = 5063001, upload-time = "2025-09-14T22:17:31.044Z" },
    { url = "https://files.pythonhosted.org/packages/db/00/87466ea3f99599d02a5238498b87bf84a6348290c19571051839ca943777/zstandard-0.25.0-cp313-cp313-manylinux2014_ppc64le.manylinux_2_17_ppc64le.whl", hash = "sha256:457ed498fc58cdc12fc48f7950e02740d4f7ae9493dd4ab2168a47c93c31298e", size = 5394120, upload-time = "2025-09-14T22:17:32.711Z" },
    { url = "https://files.pythonhosted.org/packages/2b/95/fc5531d9c618a679a20ff6c29e2b3ef1d1f4ad66c5e161ae6ff847d102a9/zstandard-0.25.0-cp313-cp313-manylinux2014_s390x.manylinux_2_17_s390x.whl", hash = "sha256:fd7a5004eb1980d3cefe26b2685bcb0b17989901a70a1040d1ac86f1d898c551", size = 5451230, upload-time = "2025-09-14T22:17:34.41Z" },
    { url = "https://files.pythonhosted.org/packages/63/4b/e3678b4e776db00f9f7b2fe58e547e8928ef32727d7a1ff01dea010f3f13/zstandard-0.25.0-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.whl", hash = "sha256:8e735494da3db08694d26480f1493ad2cf86e99bdd53e8e9771b2752a5c0246a", size = 5547173, upload-time = "2025-09-14T22:17:36.084Z" },
    { url = "https://files.pythonhosted.org/packages/4e/d5/ba05ed95c6b8ec30bd468dfeab20589f2cf709b5c940483e31d991f2ca58/zstandard-0.25.0-cp313-cp313-musllinux_1_1_aarch64.whl", hash = "sha256:3a39c94ad7866160a4a46d772e43311a743c316942037671beb264e395bdd611", size = 5046736, upload-time = "2025-09-14T22:17:37.891Z" },
    { url = "https://files.pythonhosted.org/packages/50/d5/870aa06b3a76c73eced65c044b92286a3c4e00554005ff51962deef28e28/zstandard-0.25.0-cp313-cp313-musllinux_1_1_x86_64.whl", hash = "sha256:172de1f06947577d3a3005416977cce6168f2261284c02080e7ad0185faeced3", size = 5576368, upload-time = "2025-09-14T22:17:40.206Z" },
    { url = "https://files.pythonhosted.org/packages/5d/35/398dc2ffc89d304d59bc12f0fdd931b4ce455bddf7038a0a67733a25f550/zstandard-0.25.0-cp313-cp313-musllinux_1_2_aarch64.whl", hash = "sha256:3c83b0188c852a47cd13ef3bf9209fb0a77fa5374958b8c53aaa699398c6bd7b", size = 4954022, upload-time = "2025-09-14T22:17:41.879Z" },
    { url = "https://files.pythonhosted.org/packages/9a/5c/36ba1e5507d56d2213202ec2b05e8541734af5f2ce378c5d1ceaf4d88dc4/zstandard-0.25.0-cp313-cp313-musllinux_1_2_i686.whl", hash = "sha256:1673b7199bbe763365b81a4f3252b8e80f44c9e323fc42940dc8843bfeaf9851", size = 5267889, upload-time = "2025-09-14T22:17:43.577Z" },
    { url = "https://files.pythonhosted.org/packages/70/e8/2ec6b6fb7358b2ec0113ae202647ca7c0e9d15b61c005ae5225ad0995df5/zstandard-0.25.0-cp313-cp313-musllinux_1_2_ppc64le.whl", hash = "sha256:0be7622c37c183406f3dbf0cba104118eb16a4ea7359eeb5752f0794882fc250", size = 5433952, upload-time = "2025-09-14T22:17:45.271Z" },
    { url = "https://files.pythonhosted.org/packages/7b/01/b5f4d4dbc59ef193e870495c6f1275f5b2928e01ff5a81fecb22a06e22fb/zstandard-0.25.0-cp313-cp313-musllinux_1_2_s390x.whl", hash = "sha256:5f5e4c2a23ca271c218ac025bd7d635597048b366d6f31f420aaeb715239fc98", size = 5814054, upload-time = "2025-09-14T22:17:47.08Z" },
    { url = "https://files.pythonhosted.org/packages/b2/e5/fbd822d5c6f427cf158316d012c5a12f233473c2f9c5fe5ab1ae5d21f3d8/zstandard-0.25.0-cp313-cp313-musllinux_1_2_x86_64.whl", hash = "sha256:4f187a0bb61b35119d1926aee039524d1f93aaf38a9916b8c4b78ac8514a0aaf", size = 5360113, upload-time = "2025-09-14T22:17:48.893Z" },
    { url = "https://files.pythonhosted.org/packages/8e/e0/69a553d2047f9a2c7347caa225bb3a63b6d7704ad74610cb7823baa08ed7/zstandard-0.25.0-cp313-cp313-win32.whl", hash = "sha256:7030defa83eef3e51ff26f0b7bfb229f0204b66fe18e04359ce3474ac33cbc09", size = 436936, upload-time = "2025-09-14T22:17:52.658Z" },
    { url = "https://files.pythonhosted.org/packages/d9/82/b9c06c870f3bd8767c201f1edbdf9e8dc34be5b0fbc5682c4f80fe948475/zstandard-0.25.0-cp313-cp313-win_amd64.whl", hash = "sha256:1f830a0dac88719af0ae43b8b2d6aef487d437036468ef3c2ea59c51f9d55fd5", size = 506232, upload-time = "2025-09-14T22:17:50.402Z" },
    { url = "https://files.pythonhosted.org/packages/d4/57/60c3c01243bb81d381c9916e2a6d9e149ab8627c0c7d7abb2d73384b3c0c/zstandard-0.25.0-cp313-cp313-win_arm64.whl", hash = "sha256:85304a43f4d513f5464ceb938aa02c1e78c2943b29f44a750b48b25ac999a049", size = 462671, upload-time = "2025-09-14T22:17:51.533Z" },
    { url = "https://files.pythonhosted.org/packages/3d/5c/f8923b595b55fe49e30612987ad8bf053aef555c14f05bb659dd5dbe3e8a/zstandard-0.25.0-cp314-cp314-macosx_10_13_x86_64.whl", hash = "sha256:e29f0cf06974c899b2c188ef7f783607dbef36da4c242eb6c82dcd8b512855e3", size = 795887, upload-time = "2025-09-14T22:17:54.198Z" },
    { url = "https://files.pythonhosted.org/packages/8d/09/d0a2a14fc3439c5f874042dca72a79c70a532090b7ba0003be73fee37ae2/zstandard-0.25.0-cp314-cp314-m

acosx_11_0_arm64.whl", hash = "sha256:05df5136bc5a011f33cd25bc9f506e7426c0c9b3f9954f056831ce68f3b6689f", size = 640658, upload-time = "2025-09-14T22:17:55.423Z" },
    { url = "https://files.pythonhosted.org/packages/5d/7c/8b6b71b1ddd517f68ffb55e10834388d4f793c49c6b83effaaa05785b0b4/zstandard-0.25.0-cp314-cp314-manylinux2010_i686.manylinux_2_12_i686.manylinux_2_28_i686.whl", hash = "sha256:f604efd28f239cc21b3adb53eb061e2a205dc164be408e553b41ba2ffe0ca15c", size = 5379849, upload-time = "2025-09-14T22:17:57.372Z" },
    { url = "https://files.pythonhosted.org/packages/a4/86/a48e56320d0a17189ab7a42645387334fba2200e904ee47fc5a26c1fd8ca/zstandard-0.25.0-cp314-cp314-manylinux2014_aarch64.manylinux_2_17_aarch64.manylinux_2_28_aarch64.whl", hash = "sha256:223415140608d0f0da010499eaa8ccdb9af210a543fac54bce15babbcfc78439", size = 5058095, upload-time = "2025-09-14T22:17:59.498Z" },
    { url = "https://files.pythonhosted.org/packages/f8/ad/eb659984ee2c0a779f9d06dbfe45e2dc39d99ff40a319895df2d3d9a48e5/zstandard-0.25.0-cp314-cp314-manylinux2014_ppc64le.manylinux_2_17_ppc64le.manylinux_2_28_ppc64le.whl", hash = "sha256:2e54296a283f3ab5a26fc9b8b5d4978ea0532f37b231644f367aa588930aa043", size = 5551751, upload-time = "2025-09-14T22:18:01.618Z" },
    { url = "https://files.pythonhosted.org/packages/61/b3/b637faea43677eb7bd42ab204dfb7053bd5c4582bfe6b1baefa80ac0c47b/zstandard-0.25.0-cp314-cp314-manylinux2014_s390x.manylinux_2_17_s390x.manylinux_2_28_s390x.whl", hash = "sha256:ca54090275939dc8ec5dea2d2afb400e0f83444b2fc24e07df7fdef677110859", size = 6364818, upload-time = "2025-09-14T22:18:03.769Z" },
    { url = "https://files.pythonhosted.org/packages/31/dc/cc50210e11e465c975462439a492516a73300ab8caa8f5e0902544fd748b/zstandard-0.25.0-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl", hash = "sha256:e09bb6252b6476d8d56100e8147b803befa9a12cea144bbe629dd508800d1ad0", size = 5560402, upload-time = "2025-09-14T22:18:05.954Z" },
    { url = "https://files.pythonhosted.org/packages/c9/ae/56523ae9c142f0c08efd5e868a6da613ae76614eca1305259c3bf6a0ed43/zstandard-0.25.0-cp314-cp314-musllinux_1_2_aarch64.whl", hash = "sha256:a9ec8c642d1ec73287ae3e726792dd86c96f5681eb8df274a757bf62b750eae7", size = 4955108, upload-time = "2025-09-14T22:18:07.68Z" },
    { url = "https://files.pythonhosted.org/packages/98/cf/c899f2d6df0840d5e384cf4c4121458c72802e8bda19691f3b16619f51e9/zstandard-0.25.0-cp314-cp314-musllinux_1_2_i686.whl", hash = "sha256:a4089a10e598eae6393756b036e0f419e8c1d60f44a831520f9af41c14216cf2", size = 5269248, upload-time = "2025-09-14T22:18:09.753Z" },
    { url = "https://files.pythonhosted.org/packages/1b/c0/59e912a531d91e1c192d3085fc0f6fb2852753c301a812d856d857ea03c6/zstandard-0.25.0-cp314-cp314-musllinux_1_2_ppc64le.whl", hash = "sha256:f67e8f1a324a900e75b5e28ffb152bcac9fbed1cc7b43f99cd90f395c4375344", size = 5430330, upload-time = "2025-09-14T22:18:11.966Z" },
    { url = "https://files.pythonhosted.org/packages/a0/1d/7e31db1240de2df22a58e2ea9a93fc6e38cc29353e660c0272b6735d6669/zstandard-0.25.0-cp314-cp314-musllinux_1_2_s390x.whl", hash = "sha256:9654dbc012d8b06fc3d19cc825af3f7bf8ae242226df5f83936cb39f5fdc846c", size = 5811123, upload-time = "2025-09-14T22:18:13.907Z" },
    { url = "https://files.pythonhosted.org/packages/f6/49/fac46df5ad353d50535e118d6983069df68ca5908d4d65b8c466150a4ff1/zstandard-0.25.0-cp314-cp314-musllinux_1_2_x86_64.whl", hash = "sha256:4203ce3b31aec23012d3a4cf4a2ed64d12fea5269c49aed5e4c3611b938e4088", size = 5359591, upload-time = "2025-09-14T22:18:16.465Z" },
    { url = "https://files.pythonhosted.org/packages/c2/38/f249a2050ad1eea0bb364046153942e34abba95dd5520af199aed86fbb49/zstandard-0.25.0-cp314-cp314-win32.whl", hash = "sha256:da469dc041701583e34de852d8634703550348d5822e66a0c827d39b05365b12", size = 444513, upload-time = "2025-09-14T22:18:20.61Z" },
    { url = "https://files.pythonhosted.org/packages/3a/43/241f9615bcf8ba8903b3f0432da069e857fc4fd1783bd26183db53c4804b/zstandard-0.25.0-cp314-cp314-win_amd64.whl", hash = "sha256:c19bcdd826e95671065f8692b5a4aa95c52dc7a02a4c5a0cac46deb879a017a2", size = 516118, upload-time = "2025-09-14T22:18:17.849Z" },
    { url = "https://files.pythonhosted.org/packages/f0/ef/da163ce2450ed4febf6467d77ccb4cd52c4c30ab45624bad26ca0a27260c/zstandard-0.25.0-cp314-cp314-win_arm64.whl", hash = "sha256:d7541afd73985c630bafcd6338d2518ae96060075f9463d7dc14cfb33514383d", size = 476940, upload-time = "2025-09-14T22:18:19.088Z" },
]

VERIFICATION.md

97 lines · 5.0 KB

```
# GenBI — Verification Checklist (Phase 7 Tasks 1–3)

> Run this on a machine with Docker, `uv`, and `pnpm` installed.
> These tasks convert the Phase 5b/6 work from "written + statically checked"
> to "verified working." Report failures back so they can be fixed.
>
> Expected total time: ~20–40 min (dominated by the AGE image build).

## Task 2 — Build the stack (do this FIRST)

```bash
make setup
```

What it does: prereq check → gen `.env` → install deps → build images →
start infra → migrate → start full stack → verify.

Watch for

#	Risk	Symptom	Fix
1	**AGE PG16 build fails** (highest risk)	`make setup` fails during `docker compose build` with a compile error in the AGE step	Pin an AGE release tag in `infra/postgres/Dockerfile` (`--branch PG16` → `--branch v1.5.0` or similar). May also need to match `llvm-dev`/`clang` major version to PG16's bundled LLVM.
2	`CREATE EXTENSION age` denied	`init.sql` logs "AGE not available" despite image building	Confirm the `.so` landed: `docker run --rm genbi/postgres-pgvector-age:pg16 pg_config --pkglibdir` then check `age.so` exists there.
3	`gen-env.sh` Fernet fallback	Warning about cryptography, or key looks wrong	Ensure `cryptography` is installed on host (`pip install cryptography`), or accept the base64 fallback (works, just less standard).
4	uv.lock / pnpm-lock.yaml missing	`setup.sh` prints "Generated ... — commit it"	Commit the lockfiles after generation so `--frozen` builds work next time.
5	Alembic stamp fails	"alembic not yet available" warning	Confirm `backend/alembic.ini` exists and `script_location = app/db/migrations` resolves from `WORKDIR /app`.

Done when
- `docker compose -f infra/docker-compose.dev.yml ps` shows every service.
- The AGE image built successfully (or AGE is gracefully disabled with a logged notice).

Task 1 — Run the test suite

```bash
# Inside the running backend container (preferred — matches the runtime env):
docker compose -f infra/docker-compose.dev.yml exec backend \
  uv run pytest tests/ -v -m "not e2e" --tb=short

# The golden eval gate specifically:
docker compose -f infra/docker-compose.dev.yml exec backend \
  uv run pytest tests/evals/ -v

# Or locally without Docker (faster iteration, no DB/Redis needed for most tests):
cd backend && uv sync --dev
uv run pytest tests/ -v -m "not e2e" --tb=short
```

Watch for

#	Risk	Symptom	Fix
1	**pytest-asyncio doesn't collect async class methods**	`TestEvalSuite.test_full_suite_accuracy` shows as skipped or errors with "coroutine never awaited"	Add `@pytest.mark.asyncio` to the class, OR convert the two methods to module-level async functions. Verify first — `asyncio_mode = "auto"` may already handle it.
2	`langchain_anthropic` import error at collection	ModuleNotFoundError during test collection	Confirm `uv sync --dev` installed it (it's in base deps, should be fine).
3	`load_prompt` returns empty in container	Tests pass but NL2SQL system prompt is empty	Already fixed in commit `c68b888` — verify `/app/.claude/prompts/nl2sql-system.md` exists inside the container.
4	AGE-dependent tests fail in CI	CI uses stock `pgvector/pgvector:pg16` (no AGE)	Mark AGE tests `@pytest.mark.age` and skip in CI, OR build the custom image in CI (Phase 7 Task 4).
5	Cache round-trip tests fail without Redis	`cache.set` then `cache.get` returns None	Expected — cache degrades to None without Redis. These tests need Redis or should be skipped.

Done when
- `pytest tests/ -v -m "not e2e"` exits 0.
- `pytest tests/evals/` exits 0 (the rewritten golden gate passes).
- `test_suite_detects_corrupted_mock` passes (proves the gate can fail).

Task 3 — Smoke test

```bash
make verify
```

Watch for

#	Risk	Symptom	Fix
1	Backend readiness returns 503	`/health/ready` reports database or redis unreachable	Backend likely booted before migrations finished — wait 30s and re-run. If persistent, check `make logs backend`.
2	Cube healthcheck fails	`wget: command not found` in cube container	Switch the cube healthcheck to `curl` or a TCP check in `docker-compose.dev.yml`.
3	Frontend not served	HTTP 000 on `:3000`	Frontend takes longer to start; check `make logs frontend`.

Done when
`make verify` prints all   and exits 0.

After verification
```

```

- Report which tasks passed/failed, with any error output.
- If something failed and reveals a design issue (not a quick fix), add it as
  a new Phase 7 task in `todo.md` rather than expanding scope.
- Once Tasks 1–3 are green, the final step is Phase 7 Task 5: flip the
  `   STATUS: UNVERIFIED` banners in Phase 5b/6 headers to `  STATUS: VERIFIED`.

## Branches
- `main` — all Phase 5b/6 work committed
- `fix/load-prompt-container-path` — the `load_prompt` fix (Task 1 risk #3,
  pre-resolved); merge after verification or keep if it helps
```