File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\.env

```
# Environment Variables
PORT=3000
NODE_ENV=development
NODE_ENV=development

# Database Configuration - SSO
DB_SSO_HOST=127.0.0.1
DB_SSO_PORT=3306
DB_SSO_USER=root
DB_SSO_PASSWORD=
DB_SSO_NAME=u4714151_sso

# Database Configuration - WIS
DB_WIS_HOST=127.0.0.1
DB_WIS_PORT=3306
DB_WIS_USER=root
DB_WIS_PASSWORD=
DB_WIS_NAME=u4714151_wis

# Database Configuration - WISAKA
DB_WIS_HOST=127.0.0.1
DB_WIS_PORT=3306
DB_WIS_USER=root
DB_WIS_PASSWORD=
DB_WIS_NAME=u4714151_wisaka

# Database Configuration - WISMON
DB_WISMON_HOST=127.0.0.1
DB_WISMON_PORT=3306
DB_WISMON_USER=root
DB_WISMON_PASSWORD=
DB_WISMON_NAME=u4714151_wismon

# JWT Secret
JWT_SECRET=your-super-secret-jwt-key-u4714151-wismon-development

# CORS Settings - Allow Android emulator access
CORS_ORIGIN=*

# API Host (0.0.0.0 allows external connections for Android testing)
API_HOST=0.0.0.0
```

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\.gitignore

```
# Dependencies
node_modules/
npm-debug.log*
yarn-debug.log*
yarn-error.log*

# Runtime data
pids
*.pid
*.seed
*.pid.lock

# Build output
dist/
build/
*.tsbuildinfo

# Environment variables
```

```
.env
.env.local
.env.development.local
.env.test.local
.env.production.local

# IDE files
.vscode/
.idea/
*.swp
*.swo
*~

# OS generated files
.DS_Store
.DS_Store¿
._*
.Spotlight-V100
.Trashes
ehthumbs.db
Thumbs.db

# Logs
logs
*.log

# Coverage directory used by tools like istanbul
coverage/

# Temporary folders
tmp/
temp/
```

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\API_DOCUMENTATION.md

~~~
# Wismon Keuangan Backend API Documentation

## Overview

Backend API untuk aplikasi Wismon Keuangan - sistem pembayaran mahasiswa Universitas Wirahusada.

### Database Architecture

- **wirahusada_sso**: Authentication dan authorization (7 tables)
- **wirahusada_wis**: Student dan staff management (7 tables)
- **wirahusada_wismon**: Financial transactions dan payment processing (20 tables)

## Authentication Endpoints

### POST /api/auth/login

Login mahasiswa menggunakan nama/NIM dan NRM.

**Request Body:**

```json
{
  "namam_nim": "string", // Nama mahasiswa atau NIM
  "nrm": "string" // Nomor Registrasi Mahasiswa
}
```

**Response:**
~~~

```json
{
  "success": true,
  "message": "Login successful",
  "data": {
    "token": "jwt_token_here",
    "user": {
      "nrm": "student_nrm",
      "nim": "student_nim",
      "namam": "Student Full Name"
    }
  }
}
```

Pencarian Nama Aman:
Login mendukung pencarian nama yang case-insensitive dengan exact matching untuk keamanan:

- ¿ **Case-insensitive**: `Hanik Zaimatus Sholichah` = `hanik zaimatus sholichah`
- ¿ **Tanpa spasi**: `HanikZaimatusSholichah` akan cocok dengan `Hanik Zaimatus Sholichah`
- ¿ **Mixed case**: `hAnikzaimAtussholichAh` akan cocok dengan nama asli (tanpa spasi)
- ¿ **Partial match dihapus**: Untuk keamanan, tidak ada pencarian sebagian nama

Authentication Flow:

1. Coba cari berdasarkan NIM exact match
2. Jika tidak ditemukan, cari berdasarkan nama dengan exact match (case-insensitive)
3. Jika masih tidak ditemukan, coba cari dengan menghilangkan spasi (exact match, case-insensitive)

### GET /api/auth/profile

Mendapatkan profil mahasiswa yang sedang login.

Headers:

```
Authorization: Bearer <jwt_token>
```

Response:

```json
{
  "success": true,
  "data": {
    "nrm": "student_nrm",
    "nim": "student_nim",
    "namam": "Student Full Name",
    "tgdaftar": "registration_date",
    "tplahir": "birth_place"
  }
}
```

### POST /api/auth/verify

Verifikasi token JWT.

Request Body:

```json
{
  "token": "jwt_token_here"
}
```

Response:

```json
{
  "success": true,
  "data": {
    "valid": true,
    "user": {
      "nrm": "student_nrm",
      "nim": "student_nim",
      "namam": "Student Full Name"
    }
  }
}
```

## ¿ Login Page Implementation (page1login.txt)

### `POST /api/auth/login`

Purpose: Student login authentication
Database Access:

- Primary: `wirahusada_wis.mahasiswa` (student identification)
- Secondary: `wirahusada_sso.user` (future authentication enhancement)

Request Body:

```json
{
  "namam_nim": "string", // Student name OR NIM
  "nrm": "string" // Student registration number
}
```

Response Success (200):

```json
{
  "success": true,
  "message": "Login successful",
  "data": {
    "token": "jwt_token_here",
    "user": {
      "nrm": "student_nrm",
      "nim": "student_nim",
      "namam": "student_name"
    }
  }
}
```

Database Query Flow:

1. Search by NIM + NRM: `SELECT * FROM mahasiswa WHERE nim = ¿ AND nrm = ¿`
2. If not found, search by name: `SELECT * FROM mahasiswa WHERE namam LIKE ¿ AND nrm = ¿`

## ¿ Homepage Implementation (page2home.txt)

### `GET /api/auth/profile`

Purpose: Get student profile for homepage greeting ("hello, [student's namam]")

Database Access: `wirahusada_wis.mahasiswa`
Authentication: Required

Response Success (200):

```json
{
  "success": true,
  "message": "Profile retrieved successfully",
  "data": {
    "nrm": "student_nrm",
    "nim": "student_nim",
    "namam": "student_name", // Used for homepage greeting
    "tgdaftar": "2023-01-15",
    "tplahir": "2001-05-20"
  }
}
```

Database Query: `SELECT nrm, nim, namam, tgdaftar, tplahir, kdagama FROM mahasiswa WHERE nrm = ¿`

### `POST /api/auth/verify`

Purpose: Verify JWT token validity
Authentication: Required

## ¿ Wismon Payment Page Implementation (page3wismon.txt)

### `GET /api/payments/history`

Purpose: Get payment history with filtering (Riwayat Pembayaran)
Database Access:

- `wirahusada_wismon.transaksi` - Transaction records
- `wirahusada_wismon.t_pembayaranmahasiswa` - Student payment links
- `wirahusada_wismon.akun` - Account/payment method info

Query Parameters:

```
¿page=1&limit=20&startDate=2024-01-01&endDate=2024-12-31&type=Ijazah&sortBy=tanggal&sortOrder=desc
```

Response Success (200):

```json
{
  "success": true,
  "message": "Payment history retrieved successfully",
  "data": {
    "data": [
      {
        "id": "transaction_id",
        "tanggal": "24 Juni 2025",
        "tanggal_full": "24 Juni 2025, 10:30:15 WIB",
        "type": "KTI dan Wisuda",
        "jumlah": "Rp 1.750.000",
        "status": "LUNAS",
        "tx_id": "135842-AKD-20250624",
        "method": "101 - Tunai - Kas Akademi",
        "method_code": "101"
      }
    ],
    "total": 25,
```

~~~
    "page": 1,
    "limit": 20
  }
}
```

**Database Query**:

```sql
SELECT
  t.id, t.kode_transaksi as tx_id,
  DATE_FORMAT(t.tanggal, '%d %M %Y') as tanggal,
  DATE_FORMAT(t.tanggal, '%d %M %Y, %H:%i:%s WIB') as tanggal_full,
  a.nama_akun as type,
  CONCAT('Rp ', FORMAT(t.total, 0)) as jumlah,
  'LUNAS' as status,
  CONCAT(a.kode, ' - ', a.nama_akun) as method,
  a.kode as method_code
FROM transaksi t
INNER JOIN t_pembayaranmahasiswa tpm ON t.id = tpm.no
INNER JOIN akun a ON t.kodeakun = a.kode
WHERE tpm.nrm = ¿
ORDER BY t.tanggal DESC
LIMIT ¿ OFFSET ¿
```

### `GET /api/payments/summary`

**Purpose**: Get payment recapitulation (Rekapitulasi Pembayaran)
**Database Access**: `wirahusada_wismon` (aggregated transaction data)

**Response Success (200)**:

```json
{
  "success": true,
  "message": "Payment summary retrieved successfully",
  "data": {
    "total_pembayaran": "Rp 2.300.000",
    "breakdown": {
      "Ijazah": "Rp 400.000",
      "KTI dan Wisuda": "Rp 1.750.000",
      "Lain-lain": "Rp 150.000"
    }
  }
}
```

**Database Query**:

```sql
SELECT
  a.nama_akun as payment_type,
  SUM(t.total) as total_amount
FROM transaksi t
INNER JOIN t_pembayaranmahasiswa tpm ON t.id = tpm.no
INNER JOIN akun a ON t.kodeakun = a.kode
WHERE tpm.nrm = ¿
GROUP BY a.nama_akun, a.kode
ORDER BY total_amount DESC
```

### `GET /api/payments/detail/:id`

**Purpose**: Get detailed transaction information (Detail Transaksi)
**Database Access**:
~~~

- `wirahusada_wismon` - Transaction details
- `wirahusada_wis.mahasiswa` - Student information

Response Success (200):

```json
{
  "success": true,
  "message": "Transaction detail retrieved successfully",
  "data": {
    "id": "transaction_id",
    "tanggal": "24 Juni 2025",
    "tanggal_full": "24 Juni 2025, 10:30:15 WIB",
    "type": "KTI dan Wisuda",
    "jumlah": "Rp 1.750.000",
    "status": "LUNAS",
    "tx_id": "135842-AKD-20250624", // Copyable transaction ID
    "method": "101 - Tunai - Kas Akademi",
    "student_name": "Budi Santoso",
    "student_nim": "202201001",
    "student_prodi": "D3 Keperawatan",
    "payment_breakdown": {
      "KTI dan Wisuda": "Rp 1.750.000"
    }
  }
}
```

### `POST /api/payments/refresh`

Purpose: Refresh payment data (Refresh button)

Response Success (200):

```json
{
  "success": true,
  "message": "Payment data refreshed successfully",
  "data": { "refreshed": true }
}
```

### `GET /api/payments/types`

Purpose: Get available payment types for filtering

Response Success (200):

```json
{
  "success": true,
  "message": "Payment types retrieved successfully",
  "data": [
    "Tunai - Kas Akademi",
    "Cek - Bank BNI",
    "Ijazah",
    "KTI dan Wisuda",
    "Lain-lain"
  ]
}
```

## ¿ System Endpoints

### `GET /health`

Purpose: Health check
Authentication: Not required

### `GET /`

Purpose: API information and available endpoints
Authentication: Not required

## ¿¿ Database Schema Relationships

### Key Foreign Key Relationships

- `t_pembayaranmahasiswa.nrm` ¿ `wirahusada_wis.mahasiswa.nrm`
- `t_pembayaranmahasiswa.no` ¿ `transaksi.id`
- `transaksi.kodeakun` ¿ `akun.kode`

### Data Flow Patterns

1. **Authentication**: `wirahusada_wis.mahasiswa` for student validation
2. **Homepage**: `wirahusada_wis.mahasiswa` for greeting
3. **Payment History**: `wirahusada_wismon` tables joined for transaction data
4. **Payment Detail**: Cross-database join between `wirahusada_wismon` and `wirahusada_wis`

## ¿ Frontend Integration Examples

### Login Implementation

```javascript
const login = async (namam_nim, nrm) => {
  const response = await fetch("/api/auth/login", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ namam_nim, nrm }),
  });

  const result = await response.json();
  if (result.success) {
    localStorage.setItem("token", result.data.token);
    window.location.href = "/homepage";
  }
};
```

### Homepage Greeting

```javascript
const loadHomepage = async () => {
  const response = await fetch("/api/auth/profile", {
    headers: { Authorization: `Bearer ${localStorage.getItem("token")}` },
  });

  const result = await response.json();
  if (result.success) {
    document.getElementById(
      "greeting"
    ).textContent = `hello, ${result.data.namam}`;
  }
};
```

### Payment History with Filters

```javascript
const loadPaymentHistory = async (filters = {}) => {
  const params = new URLSearchParams(filters);
  const response = await fetch(`/api/payments/history¿${params}`, {
    headers: { Authorization: `Bearer ${localStorage.getItem("token")}` },
  });

  const result = await response.json();
  if (result.success) {
    renderPaymentHistory(result.data.data);
  }
};

// Example usage with filters
loadPaymentHistory({
  page: 1,
  limit: 20,
  type: "Ijazah",
  sortBy: "tanggal",
  sortOrder: "desc",
});
```

### Transaction Detail

```javascript
const showTransactionDetail = async (transactionId) => {
  const response = await fetch(`/api/payments/detail/${transactionId}`, {
    headers: { Authorization: `Bearer ${localStorage.getItem("token")}` },
  });

  const result = await response.json();
  if (result.success) {
    populateDetailPage(result.data);
  }
};
```

## ¿ Security Features

- **JWT Authentication**: 24-hour token expiration
- **CORS Protection**: Configured for specific origins
- **Input Validation**: All inputs validated and sanitized
- **SQL Injection Prevention**: Parameterized queries
- **Authorization Checks**: User can only access their own data

---

## ¿ Performance Optimizations

- **Connection Pooling**: MySQL connection pools (limit: 10)
- **Pagination**: Default 20 items per page for history
- **Indexed Queries**: Database indexes on key columns
- **Request Logging**: For monitoring and debugging

## ¿ Error Handling

All endpoints return consistent error format:

~~~
```json
{
  "success": false,
  "message": "Error description",
  "errors": ["detailed_error_1", "detailed_error_2"]
}
```

**Common HTTP Status Codes**:

- `200`: Success
- `400`: Bad Request (validation errors)
- `401`: Unauthorized (missing/invalid token)
- `403`: Forbidden (token expired)
- `404`: Not Found (resource not found)
- `500`: Internal Server Error

**Example Error Responses**:

```json
// Login failed
{
  "success": false,
  "message": "Student not found or invalid credentials",
  "errors": ["Student not found or invalid credentials"]
}

// Missing authentication
{
  "success": false,
  "message": "Access token required",
  "errors": ["Access token required"]
}

// Invalid transaction ID
{
  "success": false,
  "message": "Transaction not found",
  "errors": ["Transaction not found or does not belong to this user"]
}
```
~~~

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\package-lock.json

```
{
  "name": "be_wismon_keuangan",
  "version": "1.0.0",
  "lockfileVersion": 3,
  "requires": true,
  "packages": {
    "": {
      "name": "be_wismon_keuangan",
      "version": "1.0.0",
      "license": "ISC",
      "dependencies": {
        "@types/bcryptjs": "^2.4.6",
        "@types/cors": "^2.8.19",
        "@types/jsonwebtoken": "^9.0.10",
        "bcryptjs": "^3.0.2",
        "cors": "^2.8.5",
        "dotenv": "^16.4.7",
        "express": "^4.21.2",
        "express-rate-limit": "^7.5.1",
        "helmet": "^8.1.0",
```

```
        "jsonwebtoken": "^9.0.2",
        "mysql2": "^3.11.5"
      },
      "devDependencies": {
        "@types/express": "^5.0.0",
        "@types/node": "^22.10.2",
        "nodemon": "^3.1.9",
        "ts-node": "^10.9.2",
        "typescript": "^5.7.2"
      }
    },
    "node_modules/@cspotcode/source-map-support": {
      "version": "0.8.1",
      "resolved": "https://registry.npmjs.org/@cspotcode/source-map-support/-/source-map-support-0.8.1.tgz",
                                                                                                     "integrity":
"sha512-IchNf6dN4tHoMFIn/7OE8LWZ19Y6q/67Bmf6vnGREv8RSbBVb9LPJxEcnwrcwX6ixSvaiGoomAUvu4YSxXrVgw==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@jridgewell/trace-mapping": "0.3.9"
      },
      "engines": {
        "node": ">=12"
      }
    },
    "node_modules/@jridgewell/resolve-uri": {
      "version": "3.1.2",
      "resolved": "https://registry.npmjs.org/@jridgewell/resolve-uri/-/resolve-uri-3.1.2.tgz",
                                                                                                     "integrity":
"sha512-bRISgCIjP20/tbWSPWMEi54QVPRZExkuD9lJL+UIxUKtwVJA8wW1Trb1jMs1RFXo1CBTNZ/5hpC9QvmKWdopKw==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=6.0.0"
      }
    },
    "node_modules/@jridgewell/sourcemap-codec": {
      "version": "1.5.0",
      "resolved": "https://registry.npmjs.org/@jridgewell/sourcemap-codec/-/sourcemap-codec-1.5.0.tgz",
                                                                                                     "integrity":
"sha512-gv3ZRaISU3fjPAgNsriBRqGWQL6quFx04YMPW/zD8XMLsU32mhCCbfbO6KZFLjvYpCZ8zyDEgqsgf+PwPaM7GQ==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/@jridgewell/trace-mapping": {
      "version": "0.3.9",
      "resolved": "https://registry.npmjs.org/@jridgewell/trace-mapping/-/trace-mapping-0.3.9.tgz",
                                                                                                     "integrity":
"sha512-3Belt6tdc8bPgAtbcmdtNJlirVoTmEb5e2gC94PnkwEW9jI6CAHUeoG85tjWP5WquqfavoMtMwiG4P926ZKKuQ==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@jridgewell/resolve-uri": "^3.0.3",
        "@jridgewell/sourcemap-codec": "^1.4.10"
      }
    },
    "node_modules/@tsconfig/node10": {
      "version": "1.0.11",
      "resolved": "https://registry.npmjs.org/@tsconfig/node10/-/node10-1.0.11.tgz",
                                                                                                     "integrity":
"sha512-DcRjDCujK/kCk/cUe8Xz8ZSpm8mS3mNNpta+jGCA6USEDfktlNvm1+IuZ9eTcDbNk41BHwpHHeW+N1lKCz4zOw==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/@tsconfig/node12": {
      "version": "1.0.11",
      "resolved": "https://registry.npmjs.org/@tsconfig/node12/-/node12-1.0.11.tgz",
                                                                                                    "integrity":
"sha512-cqefuRsh12pWyGsIoBKJA9luFu3mRxCA+ORZvA4ktLSzIuCUtWVxGIuXigEwO5/ywWFMZ2QEGKWvkZG1zDMTag==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/@tsconfig/node14": {
      "version": "1.0.3",
      "resolved": "https://registry.npmjs.org/@tsconfig/node14/-/node14-1.0.3.tgz",
                                                                                                    "integrity":
"sha512-ysT8mhdixWK6Hw3i1V2AeRqZ5WfXg1G43mqoYlM2nc6388Fq5jcXyr5mRsqViLx/GJYdoL0bfXD8nmF+Zn/Iow==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/@tsconfig/node16": {
      "version": "1.0.4",
      "resolved": "https://registry.npmjs.org/@tsconfig/node16/-/node16-1.0.4.tgz",
                                                                                                    "integrity":
"sha512-vxhUy4J8lyeyinH7Azl1pdd43GJhZH/tP2weN8TntQblOY+A0XbT8DJk1/oCPuOOyg/Ja757rG0CgHcWC8OfMA==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/@types/bcryptjs": {
      "version": "2.4.6",
      "resolved": "https://registry.npmjs.org/@types/bcryptjs/-/bcryptjs-2.4.6.tgz",
                                                                                                    "integrity":
"sha512-9xlo6R2qDs5uixm0bcIqCeMCE6HiQsIyel9KQySStiyqNl2tnj2mP3DX1Nf56MD6KMenNNlBBsy3LJ7gUEQPXQ==",
      "license": "MIT"
    },
    "node_modules/@types/body-parser": {
      "version": "1.19.6",
      "resolved": "https://registry.npmjs.org/@types/body-parser/-/body-parser-1.19.6.tgz",
                                                                                                    "integrity":
"sha512-HLFeCYgz89uk22N5Qg3dvGvsv46B8GLvKKo1zKG4NybA8U2DiEO3w9lqGg29t/tfLRJpJ6iQxnVw4OnB7MoM9g==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@types/connect": "*",
        "@types/node": "*"
      }
    },
    "node_modules/@types/connect": {
      "version": "3.4.38",
      "resolved": "https://registry.npmjs.org/@types/connect/-/connect-3.4.38.tgz",
                                                                                                    "integrity":
"sha512-K6uROf1LD88uDQqJCktA4yzL1YYAK6NgfsI0v/mTgyPKWsX1CnJ0XPSDhViejru1GcRkLWb8RlzFYJRqGUbaug==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@types/node": "*"
      }
    },
    "node_modules/@types/cors": {
      "version": "2.8.19",
      "resolved": "https://registry.npmjs.org/@types/cors/-/cors-2.8.19.tgz",
                                                                                                    "integrity":
"sha512-mFNylyeyqN93lfe/9CSxOGREz8cpzAhH+E93xJ4xWQf62V8sQ/24reV2nyzUWM6H6Xji+GGHpkbLe7pVoUEskg==",
      "license": "MIT",
      "dependencies": {
        "@types/node": "*"
      }
    },
    "node_modules/@types/express": {
      "version": "5.0.3",
      "resolved": "https://registry.npmjs.org/@types/express/-/express-5.0.3.tgz",
                                                                                                    "integrity":
```

```
"sha512-wGA0NX93b19/dZC1J18tKWVIYWyyF2ZjT9vin/NRu0qzzvfVzWjs04iq2rQ3H65vCTQYlRqs3YHfY7zjdV+9Kw==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@types/body-parser": "*",
        "@types/express-serve-static-core": "^5.0.0",
        "@types/serve-static": "*"
      }
    },
    "node_modules/@types/express-serve-static-core": {
      "version": "5.0.6",
                                                                                                   "resolved":
"https://registry.npmjs.org/@types/express-serve-static-core/-/express-serve-static-core-5.0.6.tgz",
                                                                                                  "integrity":
"sha512-3xhRnjJPkULekpSzgtoNYYcTWgEZkp4myc+Saevii5JPnHNvHMRlBSHDbs7Bh1iPPoVTERHEZXyhyLbMEsExsA==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@types/node": "*",
        "@types/qs": "*",
        "@types/range-parser": "*",
        "@types/send": "*"
      }
    },
    "node_modules/@types/http-errors": {
      "version": "2.0.5",
      "resolved": "https://registry.npmjs.org/@types/http-errors/-/http-errors-2.0.5.tgz",
                                                                                                  "integrity":
"sha512-r8Tayk8HJnX0FztbZN7oVqGccWgw98T/0neJphO91KkmOzug1KkofZURD4UaD5uH8AqcFLfdPErnBod0u71/qg==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/@types/jsonwebtoken": {
      "version": "9.0.10",
      "resolved": "https://registry.npmjs.org/@types/jsonwebtoken/-/jsonwebtoken-9.0.10.tgz",
                                                                                                  "integrity":
"sha512-asx5hIG9Qmf/1oStypjanR7iKTv0gXQ1Ov/jfrX6kS/EO0OFni8orbmGCn0672NHR3kXHwpAwR+B368ZGN/2rA==",
      "license": "MIT",
      "dependencies": {
        "@types/ms": "*",
        "@types/node": "*"
      }
    },
    "node_modules/@types/mime": {
      "version": "1.3.5",
      "resolved": "https://registry.npmjs.org/@types/mime/-/mime-1.3.5.tgz",
                                                                                                  "integrity":
"sha512-/pyBZWSLD2n0dcHE3hq8s8ZvcETHtEuF+3E7XVt0Ig2nvsVQXdghHVcEkIWjy9A0wKfTn97a/PSDYohKIlnP/w==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/@types/ms": {
      "version": "2.1.0",
      "resolved": "https://registry.npmjs.org/@types/ms/-/ms-2.1.0.tgz",
                                                                                                  "integrity":
"sha512-GsCCIZDE/p3i96vtEqx+7dBUGXrc7zeSK3wwPHIaRThS+9OhWIXRqzs4d6k1SVU8g91DrNRWxWUGhp5KXQb2VA==",
      "license": "MIT"
    },
    "node_modules/@types/node": {
      "version": "22.15.33",
      "resolved": "https://registry.npmjs.org/@types/node/-/node-22.15.33.tgz",
                                                                                                  "integrity":
"sha512-wzoocdnnpSxZ+6CjW4ADCK1jVmd1S/J3ArNWfn8FDDQtRm8dkDg7TA+mvek2wNrfCgwuZxqEOiB9B1XCJ6+dbw==",
      "license": "MIT",
      "dependencies": {
        "undici-types": "~6.21.0"
      }
    },
    "node_modules/@types/qs": {
      "version": "6.14.0",
      "resolved": "https://registry.npmjs.org/@types/qs/-/qs-6.14.0.tgz",
                                                                                                    "integrity":
"sha512-eOunJqu0K1923aExK6y8p6fsihYEn/BYuQ4g0CxAAgFc4b/ZLN4CrsRZ55srTdqoiLzU2B2evC+apEIxprEzkQ==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/@types/range-parser": {
      "version": "1.2.7",
      "resolved": "https://registry.npmjs.org/@types/range-parser/-/range-parser-1.2.7.tgz",
                                                                                                    "integrity":
"sha512-hKormJbkJqzQGhziax5PItDUTMAM9uE2XXQmM37dyd4hVM+5aVl7oVxMVUiVQn2oCQFN/LKCZdvSM0pFRqbSmQ==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/@types/send": {
      "version": "0.17.5",
      "resolved": "https://registry.npmjs.org/@types/send/-/send-0.17.5.tgz",
                                                                                                    "integrity":
"sha512-z6F2D3cOStZvuk2SaP6YrwkNO65iTZcwA2ZkSABegdkAh/lf+Aa/YQndZVfmEXT5vgAp6zv06VQ3ejSVjAny4w==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@types/mime": "^1",
        "@types/node": "*"
      }
    },
    "node_modules/@types/serve-static": {
      "version": "1.15.8",
      "resolved": "https://registry.npmjs.org/@types/serve-static/-/serve-static-1.15.8.tgz",
                                                                                                    "integrity":
"sha512-roei0UY3LhpOJvjbIP6ZZFngyLKl5dskOtDhxY5THRSpO+ZI+nzJ+m5yUMzGrp89YRa7lvknKkMYjqQFGwA7Sg==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@types/http-errors": "*",
        "@types/node": "*",
        "@types/send": "*"
      }
    },
    "node_modules/accepts": {
      "version": "1.3.8",
      "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.8.tgz",
                                                                                                    "integrity":
"sha512-PYAthTa2m2VKxuvSD3DPC/Gy+U+sOA1LAuT8mkmRuvw+NACSaeXEQ+NHcVF7rONl6qcaxV3Uuemwawk+7+SJLw==",
      "license": "MIT",
      "dependencies": {
        "mime-types": "~2.1.34",
        "negotiator": "0.6.3"
      },
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/acorn": {
      "version": "8.15.0",
      "resolved": "https://registry.npmjs.org/acorn/-/acorn-8.15.0.tgz",
                                                                                                    "integrity":
"sha512-NZyJarBfL7nWwIq+FDL6Zp/yHEhePMNnnJ0y3qfieCrmNvYct8uvtiV41UvlSe6apAfk0fY1FbWx+NwfmpvtTg==",
      "dev": true,
      "license": "MIT",
      "bin": {
        "acorn": "bin/acorn"
      },
      "engines": {
        "node": ">=0.4.0"
      }
    },
    "node_modules/acorn-walk": {
      "version": "8.3.4",
      "resolved": "https://registry.npmjs.org/acorn-walk/-/acorn-walk-8.3.4.tgz",
                                                                                                   "integrity":
"sha512-ueEepnujpqee2o5aIYnvHU6C0A42MNdsIDeqy5BydrkuC5R1ZuUFnm27EeFJGoEHJQgn3uleRvmTXaJgfXbt4g==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "acorn": "^8.11.0"
      },
      "engines": {
        "node": ">=0.4.0"
      }
    },
    "node_modules/anymatch": {
      "version": "3.1.3",
      "resolved": "https://registry.npmjs.org/anymatch/-/anymatch-3.1.3.tgz",
                                                                                                   "integrity":
"sha512-KMReFUr0B4t+D+OBkjR3KYqvocp2XaSzO55UcB6mgQMd3KbcE+mWTyvVV7D/zsdEbNnV6acZUutkiHQXvTr1Rw==",
      "dev": true,
      "license": "ISC",
      "dependencies": {
        "normalize-path": "^3.0.0",
        "picomatch": "^2.0.4"
      },
      "engines": {
        "node": ">= 8"
      }
    },
    "node_modules/arg": {
      "version": "4.1.3",
      "resolved": "https://registry.npmjs.org/arg/-/arg-4.1.3.tgz",
                                                                                                   "integrity":
"sha512-58S9QDqG0Xx27YwPSt9fJxivjYl432YCwfDMfZ+71RAqUrZef7LrKQZ3LHLOwCS4FLNBplP533Zx895SeOCHvA==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/array-flatten": {
      "version": "1.1.1",
      "resolved": "https://registry.npmjs.org/array-flatten/-/array-flatten-1.1.1.tgz",
                                                                                                   "integrity":
"sha512-PCVAQswWemu6UdxsDFFX/+gVeYqKAod3D3UVm91jHwynguOwAvYPhx8nNlM++NqRcK6CxxpUafjmhIdKiHibqg==",
      "license": "MIT"
    },
    "node_modules/aws-ssl-profiles": {
      "version": "1.1.2",
      "resolved": "https://registry.npmjs.org/aws-ssl-profiles/-/aws-ssl-profiles-1.1.2.tgz",
                                                                                                   "integrity":
"sha512-NZKeq9AfyQvEeNlN0zSYAaWrmBffJh3IELMZfRpJVWgrpEbtEpnjvzqBPf+mxoI287JohRDoa+/nsfqqiZmF6g==",
      "license": "MIT",
      "engines": {
        "node": ">= 6.0.0"
      }
    },
    "node_modules/balanced-match": {
      "version": "1.0.2",
      "resolved": "https://registry.npmjs.org/balanced-match/-/balanced-match-1.0.2.tgz",
                                                                                                   "integrity":
"sha512-3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbxRDZOpH0KW9+CeX4bRAaX0Anxt0tx2MrpRpWwQaPwIlISEJhYU5Pw==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/bcryptjs": {
      "version": "3.0.2",
      "resolved": "https://registry.npmjs.org/bcryptjs/-/bcryptjs-3.0.2.tgz",
                                                                                                    "integrity":
"sha512-k38b3XOZKv60C4E2hVsXTolJWfkGRMbILBIe2IBITXciy5bOsTKot5kDrf3ZfufQtQOUN5mXceUEpU1rTl9Uog==",
      "license": "BSD-3-Clause",
      "bin": {
        "bcrypt": "bin/bcrypt"
      }
    },
    "node_modules/binary-extensions": {
      "version": "2.3.0",
      "resolved": "https://registry.npmjs.org/binary-extensions/-/binary-extensions-2.3.0.tgz",
                                                                                                    "integrity":
"sha512-Ceh+7ox5qe7LJuLHoY0feh3pHuUDHAcRUeyL2VYghZwfpkNIy/+8Ocg0a3UuSoYzavmylwuLWQOf3hl0jjMMIw==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=8"
      },
      "funding": {
        "url": "https://github.com/sponsors/sindresorhus"
      }
    },
    "node_modules/body-parser": {
      "version": "1.20.3",
      "resolved": "https://registry.npmjs.org/body-parser/-/body-parser-1.20.3.tgz",
                                                                                                    "integrity":
"sha512-7rAxByjUMqQ3/bHJy7D6OGXvx/MMc4IqBn/X0fcM1QUcAItpZrBEYhWGem+tzXH90c+G01ypMcYJBO9Y30203g==",
      "license": "MIT",
      "dependencies": {
        "bytes": "3.1.2",
        "content-type": "~1.0.5",
        "debug": "2.6.9",
        "depd": "2.0.0",
        "destroy": "1.2.0",
        "http-errors": "2.0.0",
        "iconv-lite": "0.4.24",
        "on-finished": "2.4.1",
        "qs": "6.13.0",
        "raw-body": "2.5.2",
        "type-is": "~1.6.18",
        "unpipe": "1.0.0"
      },
      "engines": {
        "node": ">= 0.8",
        "npm": "1.2.8000 || >= 1.4.16"
      }
    },
    "node_modules/body-parser/node_modules/debug": {
      "version": "2.6.9",
      "resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",
                                                                                                    "integrity":
"sha512-bC7ElrdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUcNowp6cvspg0jXznRTKDjm/E7AdgFBVeAPVMNcKGsHMA==",
      "license": "MIT",
      "dependencies": {
        "ms": "2.0.0"
      }
    },
    "node_modules/body-parser/node_modules/ms": {
      "version": "2.0.0",
      "resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",
                                                                                                    "integrity":
"sha512-Tpp60P6IUJDTuOq/5Z8cdskzJujfwqfOTkrwIwj7IRISpnkJnT6SyJ4PCPnGMoFjC9ddhal5KVIYtAt97ix05A==",
      "license": "MIT"
    },
    "node_modules/brace-expansion": {
      "version": "1.1.12",
      "resolved": "https://registry.npmjs.org/brace-expansion/-/brace-expansion-1.1.12.tgz",
                                                                                                    "integrity":
"sha512-9T9UjW3r0UW5c1Q7GTwllptXwhvYmEzFhzMfZ9H7FQWt+uZePjZPjBP/W1ZEyZ1twGWom5/56TF4lPcqjnDHcg==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "balanced-match": "^1.0.0",
        "concat-map": "0.0.1"
      }
    },
    "node_modules/braces": {
      "version": "3.0.3",
      "resolved": "https://registry.npmjs.org/braces/-/braces-3.0.3.tgz",
                                                                                                    "integrity":
"sha512-yQbXgO/OSZVD2IsiLlro+7Hf6Q18EJrKSEsdoMzKePKXct3gvD8oLcOQdIzGupr5Fj+EDe8gO/lxc1BzfMpxvA==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "fill-range": "^7.1.1"
      },
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/buffer-equal-constant-time": {
      "version": "1.0.1",
                                                                                                     "resolved":
"https://registry.npmjs.org/buffer-equal-constant-time/-/buffer-equal-constant-time-1.0.1.tgz",
                                                                                                    "integrity":
"sha512-zRpUiDwd/xk6ADqPMATG8vc9VPrkck7T07OIx0gnjmJAnHnTVXNQG3vfvWNuiZIkwu9KrKdA1iJKfsfTVxE6NA==",
      "license": "BSD-3-Clause"
    },
    "node_modules/bytes": {
      "version": "3.1.2",
      "resolved": "https://registry.npmjs.org/bytes/-/bytes-3.1.2.tgz",
                                                                                                    "integrity":
"sha512-/Nf7TyzTx6S3yRJObOAV7956r8cr2+Oj8AC5dt8wSP3BQAoeX58NoHyCU8P8zGkNXStjTSi6fzO6F0pBdcYbEg==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/call-bind-apply-helpers": {
      "version": "1.0.2",
      "resolved": "https://registry.npmjs.org/call-bind-apply-helpers/-/call-bind-apply-helpers-1.0.2.tgz",
                                                                                                    "integrity":
"sha512-Sp1ablJ0ivDkSzjcaJdxEunN5/XvksFJ2sMBFfq6x0ryhQV/2b/KwFe21cMpmHtPOSij8K99/wSfoEuTObmuMQ==",
      "license": "MIT",
      "dependencies": {
        "es-errors": "^1.3.0",
        "function-bind": "^1.1.2"
      },
      "engines": {
        "node": ">= 0.4"
      }
    },
    "node_modules/call-bound": {
      "version": "1.0.4",
      "resolved": "https://registry.npmjs.org/call-bound/-/call-bound-1.0.4.tgz",
                                                                                                    "integrity":
"sha512-+ys997U96po4Kx/ABpBCqhA9EuxJaQWDQg7295H4hBphv3IZg0boBKuwYpt4YXp6MZ5AmZQnU/tyMTlRpaSejg==",
      "license": "MIT",
      "dependencies": {
        "call-bind-apply-helpers": "^1.0.2",
        "get-intrinsic": "^1.3.0"
      },
      "engines": {
        "node": ">= 0.4"
      },
      "funding": {
        "url": "https://github.com/sponsors/ljharb"
      }
    },
    "node_modules/chokidar": {
      "version": "3.6.0",
      "resolved": "https://registry.npmjs.org/chokidar/-/chokidar-3.6.0.tgz",
                                                                                                    "integrity":
"sha512-7VT13fmjotKpGipCW9JEQAusEPE+Ei8nl6/g4FBAmIm0GOOLMua9NDDo/DWp0ZAxCr3cPq5ZpBqmPAQgDda2Pw==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "anymatch": "~3.1.2",
        "braces": "~3.0.2",
        "glob-parent": "~5.1.2",
        "is-binary-path": "~2.1.0",
        "is-glob": "~4.0.1",
        "normalize-path": "~3.0.0",
        "readdirp": "~3.6.0"
      },
      "engines": {
        "node": ">= 8.10.0"
      },
      "funding": {
        "url": "https://paulmillr.com/funding/"
      },
      "optionalDependencies": {
        "fsevents": "~2.3.2"
      }
    },
    "node_modules/concat-map": {
      "version": "0.0.1",
      "resolved": "https://registry.npmjs.org/concat-map/-/concat-map-0.0.1.tgz",
                                                                                                    "integrity":
"sha512-/Srv4dswyQNBfohGpz9o6Yb3Gz3SrUDqBH5rTuhGR7ahtlbYKnVxw2bCFMRljaA7EXHaXZ8wsHdodFvbkhKmqg==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/content-disposition": {
      "version": "0.5.4",
      "resolved": "https://registry.npmjs.org/content-disposition/-/content-disposition-0.5.4.tgz",
                                                                                                    "integrity":
"sha512-FveZTNuGw04cxlAiWbzi6zTAL/lhehaWbTtgluJh4/E95DqMwTmha3KZN1aAWA8cFIhHzMZUvLevkw5Rqk+tSQ==",
      "license": "MIT",
      "dependencies": {
        "safe-buffer": "5.2.1"
      },
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/content-type": {
      "version": "1.0.5",
      "resolved": "https://registry.npmjs.org/content-type/-/content-type-1.0.5.tgz",
                                                                                                    "integrity":
"sha512-nTjqfcBFEipKdXCv4YDQWCfmcLZKm81ldF0pAopTvyrFGVbcR6P/VAAd5G7N+0tTr8QqiU0tFadD6FK4NtJwOA==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/cookie": {
      "version": "0.7.1",
      "resolved": "https://registry.npmjs.org/cookie/-/cookie-0.7.1.tgz",
                                                                                                        "integrity":
"sha512-6DnInpx7SJ2AK3+CTUE/ZM0vWTUboZCegxhC2xiIydHR9jNuTAASBrfEpHhiGOZw/nX51bHt6YQl8jsGo4y/0w==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/cookie-signature": {
      "version": "1.0.6",
      "resolved": "https://registry.npmjs.org/cookie-signature/-/cookie-signature-1.0.6.tgz",
                                                                                                        "integrity":
"sha512-QADzlaHc8icV8I7vbaJXJwod9HWYp8uCqf1xa4OfNu1T7JVxQIrUgOWtHdNDtPiywmFbiS12VjotIXLrKM3orQ==",
      "license": "MIT"
    },
    "node_modules/cors": {
      "version": "2.8.5",
      "resolved": "https://registry.npmjs.org/cors/-/cors-2.8.5.tgz",
                                                                                                        "integrity":
"sha512-KIHbLJqu73RGr/hnbrO9uBeixNGuvSQjul/jdFvS/KFSIH1hWVd1ng7zOHx+YrEfInLG7q4n6GHQ9cDtxv/P6g==",
      "license": "MIT",
      "dependencies": {
        "object-assign": "^4",
        "vary": "^1"
      },
      "engines": {
        "node": ">= 0.10"
      }
    },
    "node_modules/create-require": {
      "version": "1.1.1",
      "resolved": "https://registry.npmjs.org/create-require/-/create-require-1.1.1.tgz",
                                                                                                        "integrity":
"sha512-dcKFX3jn0MpIaXjisoRvexIJVEKzaq7z2rZKxf+MSr9TkdmHmsU4m2lcLojrj/FHl8mk5VxMmYA+ftRkP/3oKQ==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/debug": {
      "version": "4.4.1",
      "resolved": "https://registry.npmjs.org/debug/-/debug-4.4.1.tgz",
                                                                                                        "integrity":
"sha512-KcKCqiftBJcZr++7ykoDIEwSa3XWowTfNPo92BYxjXiyYEVrUQh2aLyhxBCwww+heortUFxEJYcRzosstTEBYQ==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "ms": "^2.1.3"
      },
      "engines": {
        "node": ">=6.0"
      },
      "peerDependenciesMeta": {
        "supports-color": {
          "optional": true
        }
      }
    },
    "node_modules/denque": {
      "version": "2.1.0",
      "resolved": "https://registry.npmjs.org/denque/-/denque-2.1.0.tgz",
                                                                                                        "integrity":
"sha512-HVQE3AAb/pxF8fQAoiqpvg9i3evqug3hoiwakOyZAwJm+6vZehbkYXZ0l4JxS+I3QxM97v5aaRNhj8v5oBhekw==",
      "license": "Apache-2.0",
      "engines": {
        "node": ">=0.10"
      }
    },
    "node_modules/depd": {
      "version": "2.0.0",
      "resolved": "https://registry.npmjs.org/depd/-/depd-2.0.0.tgz",
                                                                                                      "integrity":
"sha512-g7nH6P6dyDioJogAAGprGpCtVImJhpPk/roCzdb3fIh61/s/nPsfR6onyMwkCAR/OlC3yBC0lESvUoQEAssIrw==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/destroy": {
      "version": "1.2.0",
      "resolved": "https://registry.npmjs.org/destroy/-/destroy-1.2.0.tgz",
                                                                                                      "integrity":
"sha512-2sJGJTaXIIaR1w4iJSNoN0hnMY7Gpc/n8D4qSCJw8QqFWXf7cuAgnEHxBpweaVcPevC2l3KpjYCx3NypQQgaJg==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.8",
        "npm": "1.2.8000 || >= 1.4.16"
      }
    },
    "node_modules/diff": {
      "version": "4.0.2",
      "resolved": "https://registry.npmjs.org/diff/-/diff-4.0.2.tgz",
                                                                                                      "integrity":
"sha512-58lmxKSA4BNyLz+HHMUzlOEpg09FV+ev6ZMe3vJihgdxzgcwZ8VoEEPmALCZG9LmqfVoNMMKpttIYTVG6uDY7A==",
      "dev": true,
      "license": "BSD-3-Clause",
      "engines": {
        "node": ">=0.3.1"
      }
    },
    "node_modules/dotenv": {
      "version": "16.5.0",
      "resolved": "https://registry.npmjs.org/dotenv/-/dotenv-16.5.0.tgz",
                                                                                                      "integrity":
"sha512-m/C+AwOAr9/W1UOIZUo232ejMNnJAJtYQjUbHoNTBNTJSvqzzDh7vnrei3o3r3m9blf6ZoDkvcw0VmozNRFJxg==",
      "license": "BSD-2-Clause",
      "engines": {
        "node": ">=12"
      },
      "funding": {
        "url": "https://dotenvx.com"
      }
    },
    "node_modules/dunder-proto": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/dunder-proto/-/dunder-proto-1.0.1.tgz",
                                                                                                      "integrity":
"sha512-KIN/nDJBQRcXw0MLVhZE9iQHmG68qAVIBg9CqmUYjmQIhgij9U5MFvrqkUL5FbtyyzZuOeOt0zdeRe4UY7ct+A==",
      "license": "MIT",
      "dependencies": {
        "call-bind-apply-helpers": "^1.0.1",
        "es-errors": "^1.3.0",
        "gopd": "^1.2.0"
      },
      "engines": {
        "node": ">= 0.4"
      }
    },
    "node_modules/ecdsa-sig-formatter": {
      "version": "1.0.11",
      "resolved": "https://registry.npmjs.org/ecdsa-sig-formatter/-/ecdsa-sig-formatter-1.0.11.tgz",
                                                                                                    "integrity":
"sha512-nagl3RYrbNv6kQkeJIpt6NJZy8twLB/2vtz6yN9Z4vRKHN4/QZJIEbqohALSgwKdnksuY3k5Addp5lg8sVoVcQ==",
      "license": "Apache-2.0",
      "dependencies": {
        "safe-buffer": "^5.0.1"
      }
    },
    "node_modules/ee-first": {
      "version": "1.1.1",
      "resolved": "https://registry.npmjs.org/ee-first/-/ee-first-1.1.1.tgz",
                                                                                                    "integrity":
"sha512-WMwm9LhRUo+WUaRN+vRuETqG89IgZphVSNkdFgeb6sS/E4OrDIN7t48CAewSHXc6C8lefD8KKfr5vY61brQlow==",
      "license": "MIT"
    },
    "node_modules/encodeurl": {
      "version": "2.0.0",
      "resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-2.0.0.tgz",
                                                                                                    "integrity":
"sha512-Q0n9HRi4m6JuGIV1eFlmvJB7ZEVxu93IrMyiMsGC0lrMJMWzRgx6WGquyfQgZVb31vhGgXnfmPNNXmxnOkRBrg==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/es-define-property": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/es-define-property/-/es-define-property-1.0.1.tgz",
                                                                                                    "integrity":
"sha512-e3nRfgfUZ4rNGL232gUgX06QNyyez04KdjFrF+LTRoOXmrOgFKDg4BCdsjW8EnT69eqdYGmRpJwiPVYNrCaW3g==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.4"
      }
    },
    "node_modules/es-errors": {
      "version": "1.3.0",
      "resolved": "https://registry.npmjs.org/es-errors/-/es-errors-1.3.0.tgz",
                                                                                                    "integrity":
"sha512-Zf5H2Kxt2xjTvbJvP2ZWLEICxA6j+hAmMzIlypy4xcBg1vKVnx89Wy0GbS+kf5cwCVFFzdCFh2XSCFNULS6csw==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.4"
      }
    },
    "node_modules/es-object-atoms": {
      "version": "1.1.1",
      "resolved": "https://registry.npmjs.org/es-object-atoms/-/es-object-atoms-1.1.1.tgz",
                                                                                                    "integrity":
"sha512-FGgH2h8zKNim9ljj7dankFPcICIK9Cp5bm+c2gQSYePhpaG5+esrLODihIorn+Pe6FGJzWhXQotPv73jTaldXA==",
      "license": "MIT",
      "dependencies": {
        "es-errors": "^1.3.0"
      },
      "engines": {
        "node": ">= 0.4"
      }
    },
    "node_modules/escape-html": {
      "version": "1.0.3",
      "resolved": "https://registry.npmjs.org/escape-html/-/escape-html-1.0.3.tgz",
                                                                                                    "integrity":
"sha512-NiSupZ4OeuGwr68lGIeym/ksIZMJodUGOSCZ/FSnTxcrekbvqrgdUxlJOMpijaKZVjAJrWrGs/6Jy8OMuyj9ow==",
      "license": "MIT"
    },
    "node_modules/etag": {
      "version": "1.8.1",
      "resolved": "https://registry.npmjs.org/etag/-/etag-1.8.1.tgz",
                                                                                                    "integrity":
"sha512-aIL5Fx7mawVa300al2BnEE4iNvo1qETxLrPI/o05L7z6go7fCw1J6EQmbK4FmJ2AS7kgVF/KEZWufBfdClMcPg==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/express": {
      "version": "4.21.2",
      "resolved": "https://registry.npmjs.org/express/-/express-4.21.2.tgz",
                                                                                                    "integrity":
"sha512-28HqgMZAmih1Czt9ny7qr6ek2qddF4FclbMzwhCREB6OFfH+rXAnuNCwo1/wFvrtbgsQDb4kSbX9de9lFbrXnA==",
      "license": "MIT",
      "dependencies": {
        "accepts": "~1.3.8",
        "array-flatten": "1.1.1",
        "body-parser": "1.20.3",
        "content-disposition": "0.5.4",
        "content-type": "~1.0.4",
        "cookie": "0.7.1",
        "cookie-signature": "1.0.6",
        "debug": "2.6.9",
        "depd": "2.0.0",
        "encodeurl": "~2.0.0",
        "escape-html": "~1.0.3",
        "etag": "~1.8.1",
        "finalhandler": "1.3.1",
        "fresh": "0.5.2",
        "http-errors": "2.0.0",
        "merge-descriptors": "1.0.3",
        "methods": "~1.1.2",
        "on-finished": "2.4.1",
        "parseurl": "~1.3.3",
        "path-to-regexp": "0.1.12",
        "proxy-addr": "~2.0.7",
        "qs": "6.13.0",
        "range-parser": "~1.2.1",
        "safe-buffer": "5.2.1",
        "send": "0.19.0",
        "serve-static": "1.16.2",
        "setprototypeof": "1.2.0",
        "statuses": "2.0.1",
        "type-is": "~1.6.18",
        "utils-merge": "1.0.1",
        "vary": "~1.1.2"
      },
      "engines": {
        "node": ">= 0.10.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/express"
      }
    },
    "node_modules/express-rate-limit": {
      "version": "7.5.1",
      "resolved": "https://registry.npmjs.org/express-rate-limit/-/express-rate-limit-7.5.1.tgz",
                                                                                                    "integrity":
"sha512-7iN8iPMDzOMHPUYllBEsQdWVB6fPDMPqwjBaFrgr4Jgr/+okjvzAy+UHlYYL/Vs0OsOrMkwS6PJDkFlJwoxUnw==",
      "license": "MIT",
      "engines": {
        "node": ">= 16"
      },
      "funding": {
        "url": "https://github.com/sponsors/express-rate-limit"
      },
      "peerDependencies": {
        "express": ">= 4.11"
      }
    },
    "node_modules/express/node_modules/debug": {
      "version": "2.6.9",
      "resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",
                                                                                                    "integrity":
"sha512-bC7ElrdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUcNowp6cvspg0jXznRTKDjm/E7AdgFBVeAPVMNcKGsHMA==",
      "license": "MIT",
      "dependencies": {
        "ms": "2.0.0"
      }
    },
    "node_modules/express/node_modules/ms": {
      "version": "2.0.0",
      "resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",
                                                                                                    "integrity":
"sha512-Tpp60P6IUJDTuOq/5Z8cdskzJujfwqfOTkrwIwj7IRISpnkJnT6SyJ4PCPnGMoFjC9ddhal5KVIYtAt97ix05A==",
      "license": "MIT"
    },
    "node_modules/fill-range": {
      "version": "7.1.1",
      "resolved": "https://registry.npmjs.org/fill-range/-/fill-range-7.1.1.tgz",
                                                                                                    "integrity":
"sha512-YsGpe3WHLK8ZYi4tWDg2Jy3ebRz2rXowDxnld4bkQB00cc/1Zw9AWnC0i9ztDJitivtQvaI9KaLyKrc+hBW0yg==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "to-regex-range": "^5.0.1"
      },
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/finalhandler": {
      "version": "1.3.1",
      "resolved": "https://registry.npmjs.org/finalhandler/-/finalhandler-1.3.1.tgz",
                                                                                                    "integrity":
"sha512-6BN9trH7bp3qvnrRyzsBz+g3lZxTNZTbVO2EV1CS0WIcDbawYVdYvGflME/9QP0h0pYlCDBCTjYa9nZzMDpyxQ==",
      "license": "MIT",
      "dependencies": {
        "debug": "2.6.9",
        "encodeurl": "~2.0.0",
        "escape-html": "~1.0.3",
        "on-finished": "2.4.1",
        "parseurl": "~1.3.3",
        "statuses": "2.0.1",
        "unpipe": "~1.0.0"
      },
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/finalhandler/node_modules/debug": {
      "version": "2.6.9",
      "resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",
                                                                                                    "integrity":
"sha512-bC7ElrdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUcNowp6cvspg0jXznRTKDjm/E7AdgFBVeAPVMNcKGsHMA==",
      "license": "MIT",
      "dependencies": {
        "ms": "2.0.0"
      }
    },
    "node_modules/finalhandler/node_modules/ms": {
      "version": "2.0.0",
      "resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",
                                                                                                    "integrity":
"sha512-Tpp60P6IUJDTuOq/5Z8cdskzJujfwqfOTkrwIwj7IRISpnkJnT6SyJ4PCPnGMoFjC9ddhal5KVIYtAt97ix05A==",
      "license": "MIT"
    },
    "node_modules/forwarded": {
      "version": "0.2.0",
      "resolved": "https://registry.npmjs.org/forwarded/-/forwarded-0.2.0.tgz",
                                                                                                    "integrity":
"sha512-buRG0fpBtRHSTCOASe6hD258tEubFoRLb4ZNA6NxMVHNw2gOcwHo9wyablzMzOA5z9xA9L1KNjk/Nt6MT9aYow==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/fresh": {
      "version": "0.5.2",
      "resolved": "https://registry.npmjs.org/fresh/-/fresh-0.5.2.tgz",
                                                                                                    "integrity":
"sha512-zJ2mQYM18rEFOudeV4GShTGIQ7RbzA7ozbU9I/XBpm7kqgMywgmylMwXHxZJmkVoYkna9d2pVXVXPdYTP9ej8Q==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/fsevents": {
      "version": "2.3.3",
      "resolved": "https://registry.npmjs.org/fsevents/-/fsevents-2.3.3.tgz",
                                                                                                    "integrity":
"sha512-5xoDfX+fL7faATnagmWPpbFtwh/R77WmMMqqHGS65C3vvB0YHrgF+B1YmZ3441tMj5n63k0212XNoJwzlhffQw==",
      "dev": true,
      "hasInstallScript": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "darwin"
      ],
      "engines": {
        "node": "^8.16.0 || ^10.6.0 || >=11.0.0"
      }
    },
    "node_modules/function-bind": {
      "version": "1.1.2",
      "resolved": "https://registry.npmjs.org/function-bind/-/function-bind-1.1.2.tgz",
                                                                                                    "integrity":
"sha512-7XHNxH7qX9xG5mIwxkhumTox/MIRNcOgDrxWsMt2pAr23WHp6MrRlN7FBSFpCpr+oVO0F744iUgR82nJMfG2SA==",
      "license": "MIT",
      "funding": {
        "url": "https://github.com/sponsors/ljharb"
      }
    },
    "node_modules/generate-function": {
      "version": "2.3.1",
      "resolved": "https://registry.npmjs.org/generate-function/-/generate-function-2.3.1.tgz",
                                                                                                    "integrity":
"sha512-eeB5GfMNeevm/GRYq20ShmsaGcmI81kIX2K9XQx5miC8KdHaC6Jm0qQ8ZNeGOi7wYB8OsdxKs+Y2oVuTFuVwKQ==",
      "license": "MIT",
      "dependencies": {
        "is-property": "^1.0.2"
      }
    },
    "node_modules/get-intrinsic": {
      "version": "1.3.0",
      "resolved": "https://registry.npmjs.org/get-intrinsic/-/get-intrinsic-1.3.0.tgz",
                                                                                                    "integrity":
```

```
"sha512-9fSjSaos/fRIVIp+xSJlE6lfwhES7LNtKaCBIamHsjr2na1BiABJPo0mOjjz8GJDURarmCPGqaiVg5mfjb98CQ==",
      "license": "MIT",
      "dependencies": {
        "call-bind-apply-helpers": "^1.0.2",
        "es-define-property": "^1.0.1",
        "es-errors": "^1.3.0",
        "es-object-atoms": "^1.1.1",
        "function-bind": "^1.1.2",
        "get-proto": "^1.0.1",
        "gopd": "^1.2.0",
        "has-symbols": "^1.1.0",
        "hasown": "^2.0.2",
        "math-intrinsics": "^1.1.0"
      },
      "engines": {
        "node": ">= 0.4"
      },
      "funding": {
        "url": "https://github.com/sponsors/ljharb"
      }
    },
    "node_modules/get-proto": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/get-proto/-/get-proto-1.0.1.tgz",
                                                                                                         "integrity":
"sha512-sTSfBjoXBp89JvIKIefqw7U2CCebsc74kiY6awiGogKtoSGbgjYE/G/+l9sF3MWFPNc9IcoOC4ODfKHfxFmp0g==",
      "license": "MIT",
      "dependencies": {
        "dunder-proto": "^1.0.1",
        "es-object-atoms": "^1.0.0"
      },
      "engines": {
        "node": ">= 0.4"
      }
    },
    "node_modules/glob-parent": {
      "version": "5.1.2",
      "resolved": "https://registry.npmjs.org/glob-parent/-/glob-parent-5.1.2.tgz",
                                                                                                         "integrity":
"sha512-AOIgSQCepiJYwP3ARnGx+5VnTu2HBYdzbGP45eLw1vr3zB3vZLeyed1sC9hnbcOc9/SrMyM5RPQrkGz4aS9Zow==",
      "dev": true,
      "license": "ISC",
      "dependencies": {
        "is-glob": "^4.0.1"
      },
      "engines": {
        "node": ">= 6"
      }
    },
    "node_modules/gopd": {
      "version": "1.2.0",
      "resolved": "https://registry.npmjs.org/gopd/-/gopd-1.2.0.tgz",
                                                                                                         "integrity":
"sha512-ZUKRh6/kUFoAiTAtTYPZJ3hw9wNxx+BIBOijnlG9PnrJsCcSjs1wyyD6vJpaYtgnzDrKYRSqf3OO6Rfa93xsRg==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.4"
      },
      "funding": {
        "url": "https://github.com/sponsors/ljharb"
      }
    },
    "node_modules/has-flag": {
      "version": "3.0.0",
      "resolved": "https://registry.npmjs.org/has-flag/-/has-flag-3.0.0.tgz",
                                                                                                         "integrity":
```

```
"sha512-sKJf1+ceQBr4SMkvQnBDNDtf4TXpVhVGateu0t918bl30FnbE2m4vNLX+VWe/dpjlb+HugGYzW7uQXH98HPEYw==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=4"
      }
    },
    "node_modules/has-symbols": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/has-symbols/-/has-symbols-1.1.0.tgz",
                                                                                                    "integrity":
"sha512-1cDNdwJ2Jaohmb3sg4OmKaMBwuC48sYni5HUw2DvsC8LjGTLK9h+eb1X6RyuOHe4hT0ULCW68iomhjUoKUqlPQ==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.4"
      },
      "funding": {
        "url": "https://github.com/sponsors/ljharb"
      }
    },
    "node_modules/hasown": {
      "version": "2.0.2",
      "resolved": "https://registry.npmjs.org/hasown/-/hasown-2.0.2.tgz",
                                                                                                    "integrity":
"sha512-0hJU9SCPvmMzIBdZFqNPXWa6dqh7WdH0cII9y+CyS8rG3nL48Bclra9HmKhVVUHyPWNH5Y7xDwAB7bfgSjkUMQ==",
      "license": "MIT",
      "dependencies": {
        "function-bind": "^1.1.2"
      },
      "engines": {
        "node": ">= 0.4"
      }
    },
    "node_modules/helmet": {
      "version": "8.1.0",
      "resolved": "https://registry.npmjs.org/helmet/-/helmet-8.1.0.tgz",
                                                                                                    "integrity":
"sha512-jOiHyAZsmnr8LqoPGmCjYAaiuWwjAPLgY8ZX2XrmHawt99/u1y6RgrZMTeoPfpUbV96HOalYgz1qzkRbw54Pmg==",
      "license": "MIT",
      "engines": {
        "node": ">=18.0.0"
      }
    },
    "node_modules/http-errors": {
      "version": "2.0.0",
      "resolved": "https://registry.npmjs.org/http-errors/-/http-errors-2.0.0.tgz",
                                                                                                    "integrity":
"sha512-FtwrG/euBzaEjYeRqOgly7G0qviiXoJWnvEH2Z1plBdXgbyjv34pHTSb9zoeHMyDy33+DWy5Wt9Wo+TURtOYSQ==",
      "license": "MIT",
      "dependencies": {
        "depd": "2.0.0",
        "inherits": "2.0.4",
        "setprototypeof": "1.2.0",
        "statuses": "2.0.1",
        "toidentifier": "1.0.1"
      },
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/iconv-lite": {
      "version": "0.4.24",
      "resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.4.24.tgz",
                                                                                                    "integrity":
"sha512-v3MXnZAcvnywkTUEZomIActle7RXXeedOR31wwl7VlyoXO4Qi9arvSenNQWne1TcRwhCL1HwLI21bEqdpj8/rA==",
      "license": "MIT",
      "dependencies": {
        "safer-buffer": ">= 2.1.2 < 3"
      },
      "engines": {
        "node": ">=0.10.0"
      }
    },
    "node_modules/ignore-by-default": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/ignore-by-default/-/ignore-by-default-1.0.1.tgz",
                                                                                                        "integrity":
"sha512-Ius2VYcGNk7T90CppJqcIkS5ooHUZyIQK+ClZfMfMNFEF9VSE73Fq+906u/CWu92x4gzZMWOwfFYckPObzdEbA==",
      "dev": true,
      "license": "ISC"
    },
    "node_modules/inherits": {
      "version": "2.0.4",
      "resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.4.tgz",
                                                                                                        "integrity":
"sha512-k/vGaX4/Yla3WzyMCvTQOXYeIHvqOKtnqBduzTHpzpQZzAskKMhZ2K+EnBiSM9zGSoIFeMpXKxa4dYeZIQqewQ==",
      "license": "ISC"
    },
    "node_modules/ipaddr.js": {
      "version": "1.9.1",
      "resolved": "https://registry.npmjs.org/ipaddr.js/-/ipaddr.js-1.9.1.tgz",
                                                                                                        "integrity":
"sha512-0KI/607xoxSToH7GjN1FfSbLoU0+btTicjsQSWQlh/hZykN8KpmMf7uYwPW3R+akZ6R/w18ZlXSHBYXiYUPO3g==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.10"
      }
    },
    "node_modules/is-binary-path": {
      "version": "2.1.0",
      "resolved": "https://registry.npmjs.org/is-binary-path/-/is-binary-path-2.1.0.tgz",
                                                                                                        "integrity":
"sha512-ZMERYes6pDydyuGidse7OsHxtbI7WVeUEozgR/g7rd0xUimYNlvZRE/K2MgZTjWy725IfelLeVcEM97mmtRGXw==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "binary-extensions": "^2.0.0"
      },
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/is-extglob": {
      "version": "2.1.1",
      "resolved": "https://registry.npmjs.org/is-extglob/-/is-extglob-2.1.1.tgz",
                                                                                                        "integrity":
"sha512-SbKbANkN603Vi4jEZv49LeVJMn4yGwsbzZworEoyEiutsN3nJYdbO36zfhGJ6QEDpOZIFkDtnq5JRxmvl3jsoQ==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=0.10.0"
      }
    },
    "node_modules/is-glob": {
      "version": "4.0.3",
      "resolved": "https://registry.npmjs.org/is-glob/-/is-glob-4.0.3.tgz",
                                                                                                        "integrity":
"sha512-xelSayHH36ZgE7ZWhli7pW34hNbNl8Ojv5KVmkJD4hBdD3th8Tfk9vYasLM+mXWOZhFkgZfxhLSnrwRr4elSSg==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "is-extglob": "^2.1.1"
      },
      "engines": {
        "node": ">=0.10.0"
      }
    },
    "node_modules/is-number": {
      "version": "7.0.0",
      "resolved": "https://registry.npmjs.org/is-number/-/is-number-7.0.0.tgz",
                                                                                                   "integrity":
"sha512-41Cifkg6e8TylSpdtTpeLVMqvSBEVzTttHvERD741+pnZ8ANv0004MRL43QKPDlK9cGvNp6NZWZUBlbGXYxxng==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=0.12.0"
      }
    },
    "node_modules/is-property": {
      "version": "1.0.2",
      "resolved": "https://registry.npmjs.org/is-property/-/is-property-1.0.2.tgz",
                                                                                                   "integrity":
"sha512-Ks/IoX00TtClbGQr4TWXemAnktAQvYB7HzcCxDGqEZU6oCmb2INHuOoKxbtR+HFkmYWBKv/dOZtGRiAjDhj92g==",
      "license": "MIT"
    },
    "node_modules/jsonwebtoken": {
      "version": "9.0.2",
      "resolved": "https://registry.npmjs.org/jsonwebtoken/-/jsonwebtoken-9.0.2.tgz",
                                                                                                   "integrity":
"sha512-PRp66vJ865SSqOlgqS8hujT5U4AOgMfhrwYIuIhfKaoSCZcirrmASQr8CX7cUg+RMih+hgznrjp99o+W4pJLHQ==",
      "license": "MIT",
      "dependencies": {
        "jws": "^3.2.2",
        "lodash.includes": "^4.3.0",
        "lodash.isboolean": "^3.0.3",
        "lodash.isinteger": "^4.0.4",
        "lodash.isnumber": "^3.0.3",
        "lodash.isplainobject": "^4.0.6",
        "lodash.isstring": "^4.0.1",
        "lodash.once": "^4.0.0",
        "ms": "^2.1.1",
        "semver": "^7.5.4"
      },
      "engines": {
        "node": ">=12",
        "npm": ">=6"
      }
    },
    "node_modules/jwa": {
      "version": "1.4.2",
      "resolved": "https://registry.npmjs.org/jwa/-/jwa-1.4.2.tgz",
                                                                                                   "integrity":
"sha512-eeH5JO+21J78qMvTIDdBXidBd6nG2kZjg5Ohz/1fpa28Z4CcsWUzJ1ZZyFq/3z3N17aZy+ZuBoHljASbL1WfOw==",
      "license": "MIT",
      "dependencies": {
        "buffer-equal-constant-time": "^1.0.1",
        "ecdsa-sig-formatter": "1.0.11",
        "safe-buffer": "^5.0.1"
      }
    },
    "node_modules/jws": {
      "version": "3.2.2",
      "resolved": "https://registry.npmjs.org/jws/-/jws-3.2.2.tgz",
                                                                                                   "integrity":
"sha512-YHlZCB6lMTllWDtSPHz/ZXTsi8S00usEV6v1tjq8tOUZzw7DpSDWVXjXDre6ed1w/pd495ODpHZYSdkRTsa0HA==",
      "license": "MIT",
      "dependencies": {
        "jwa": "^1.4.1",
        "safe-buffer": "^5.0.1"
      }
    },
    "node_modules/lodash.includes": {
      "version": "4.3.0",
      "resolved": "https://registry.npmjs.org/lodash.includes/-/lodash.includes-4.3.0.tgz",
                                                                                                    "integrity":
"sha512-W3Bx6mdkRTGtlJISOvVD/lbqjTlPPUDTMnlXZFnVwi9NKJ6tiAk6LVdlhZMm17VZisqhKcgzpO5Wz91PCt5b0w==",
      "license": "MIT"
    },
    "node_modules/lodash.isboolean": {
      "version": "3.0.3",
      "resolved": "https://registry.npmjs.org/lodash.isboolean/-/lodash.isboolean-3.0.3.tgz",
                                                                                                    "integrity":
"sha512-Bz5mupy2SVbPHURB98VAcw+aHh4vRV5IPNhILUCsOzRmsTmSQ17jIuqopAentWoehktxGd9e/hbIXq980/1QJg==",
      "license": "MIT"
    },
    "node_modules/lodash.isinteger": {
      "version": "4.0.4",
      "resolved": "https://registry.npmjs.org/lodash.isinteger/-/lodash.isinteger-4.0.4.tgz",
                                                                                                    "integrity":
"sha512-DBwtEWN2caHQ9/imiNeEA5ys1JoRtRfY3d7V9wkqtbycnAmTvRRmbHKDV4a0EYc678/dia0jrte4tjYwVBaZUA==",
      "license": "MIT"
    },
    "node_modules/lodash.isnumber": {
      "version": "3.0.3",
      "resolved": "https://registry.npmjs.org/lodash.isnumber/-/lodash.isnumber-3.0.3.tgz",
                                                                                                    "integrity":
"sha512-QYqzpfwO3/CWf3XP+Z+tkQsfaLL/EnUlXWVkIk5FUPc4sBdTehEqZONuyRt2P67PXAk+NXmTBcc97zw9t1FQrw==",
      "license": "MIT"
    },
    "node_modules/lodash.isplainobject": {
      "version": "4.0.6",
      "resolved": "https://registry.npmjs.org/lodash.isplainobject/-/lodash.isplainobject-4.0.6.tgz",
                                                                                                    "integrity":
"sha512-oSXzaWypCMHkPC3NvBEaPHf0KsA5mvPrOPgQWDsbg8n7orZ290M0BmC/jgRZ4vcJ6DTAhjrsSYgdsW/F+MFOBA==",
      "license": "MIT"
    },
    "node_modules/lodash.isstring": {
      "version": "4.0.1",
      "resolved": "https://registry.npmjs.org/lodash.isstring/-/lodash.isstring-4.0.1.tgz",
                                                                                                    "integrity":
"sha512-0wJxfxH1wgO3GrbuP+dTTk7op+6L41QCXbGINEmD+ny/G/eCqGzxyCsh7159S+mgDDcoarnBw6PC1PS5+wUGgw==",
      "license": "MIT"
    },
    "node_modules/lodash.once": {
      "version": "4.1.1",
      "resolved": "https://registry.npmjs.org/lodash.once/-/lodash.once-4.1.1.tgz",
                                                                                                    "integrity":
"sha512-Sb487aTOCr9drQVL8pIxOzVhafOjZN9UU54hiN8PU3uAiSV7lx1yYNpbNmex2PK6dSJoNTSJUUswT651yww3Mg==",
      "license": "MIT"
    },
    "node_modules/long": {
      "version": "5.3.2",
      "resolved": "https://registry.npmjs.org/long/-/long-5.3.2.tgz",
                                                                                                    "integrity":
"sha512-mNAgZ1GmyNhD7AuqnTG3/VQ26o760+ZYBPKjPvugO8+nLbYfX6TVpJPseBvopbdY+qpZ/lKUnmEc1LeZYS3QAA==",
      "license": "Apache-2.0"
    },
    "node_modules/lru-cache": {
      "version": "7.18.3",
      "resolved": "https://registry.npmjs.org/lru-cache/-/lru-cache-7.18.3.tgz",
                                                                                                    "integrity":
"sha512-jumlc0BIUrS3qJGgIkWZsyfAM7NCWiBcCDhnd+3NNM5KbBmLTgHVfWBcg6W+rLUsIpzpERPsvwUP7CckAQSOoA==",
      "license": "ISC",
      "engines": {
        "node": ">=12"
      }
    },
    "node_modules/lru.min": {
      "version": "1.1.2",
      "resolved": "https://registry.npmjs.org/lru.min/-/lru.min-1.1.2.tgz",
                                                                                                    "integrity":
"sha512-Nv9KddBcQSlQopmBHXSsZVY5xsdlZkdH/Iey0BlcBYggMd4two7cZnKOK9vmy3nY0O5RGH99z1PCeTpPqszUYg==",
      "license": "MIT",
      "engines": {
        "bun": ">=1.0.0",
        "deno": ">=1.30.0",
        "node": ">=8.0.0"
      },
      "funding": {
        "type": "github",
        "url": "https://github.com/sponsors/wellwelwel"
      }
    },
    "node_modules/make-error": {
      "version": "1.3.6",
      "resolved": "https://registry.npmjs.org/make-error/-/make-error-1.3.6.tgz",
                                                                                                    "integrity":
"sha512-s8UhlNe7vPKomQhC1qFelMokr/Sc3AgNbso3n74mVPA5LTZwkB9NlXf4XPamLxJE8h0gh73rM94xvwRT2CVInw==",
      "dev": true,
      "license": "ISC"
    },
    "node_modules/math-intrinsics": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/math-intrinsics/-/math-intrinsics-1.1.0.tgz",
                                                                                                    "integrity":
"sha512-/IXtbwEk5HTPyEwyKX6hGkYXxM9nbj64B+ilVJnC/R6B0pH5G4V3b0pVbL7DBj4tkhBAppbQUlf6F6Xl9LHu1g==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.4"
      }
    },
    "node_modules/media-typer": {
      "version": "0.3.0",
      "resolved": "https://registry.npmjs.org/media-typer/-/media-typer-0.3.0.tgz",
                                                                                                    "integrity":
"sha512-dq+qelQ9akHpcOl/gUVRTxVIOkAJ1wR3QAvb4RsVjS8oVoFjDGTc679wJYmUmknUF5HwMLOgb5O+a3KxfWapPQ==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/merge-descriptors": {
      "version": "1.0.3",
      "resolved": "https://registry.npmjs.org/merge-descriptors/-/merge-descriptors-1.0.3.tgz",
                                                                                                    "integrity":
"sha512-gaNvAS7TZ897/rVaZ0nMtAyxNyi/pdbjbAwUpFQpN70GqnVfOiXpeUUMKRBmzXaSQ8DdTX4/0ms62r2K+hE6mQ==",
      "license": "MIT",
      "funding": {
        "url": "https://github.com/sponsors/sindresorhus"
      }
    },
    "node_modules/methods": {
      "version": "1.1.2",
      "resolved": "https://registry.npmjs.org/methods/-/methods-1.1.2.tgz",
                                                                                                    "integrity":
"sha512-iclAHeNqNm68zFtnZ0e+1L2yUIdvzNoauKU4WBA3VvH/vPFieF7qfRlwUZU+DA9P9bPXIS90ulxoUoCH23sV2w==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/mime": {
      "version": "1.6.0",
      "resolved": "https://registry.npmjs.org/mime/-/mime-1.6.0.tgz",
                                                                                                    "integrity":
"sha512-x0Vn8spI+wuJ1O6S7gnbaQg8Pxh4NNHb7KSINmEWKiPE4RKOplvijn+NkmYmmRgP68mc70j2EbeTFRsrswaQeg==",
      "license": "MIT",
      "bin": {
        "mime": "cli.js"
      },
      "engines": {
        "node": ">=4"
      }
    },
    "node_modules/mime-db": {
      "version": "1.52.0",
      "resolved": "https://registry.npmjs.org/mime-db/-/mime-db-1.52.0.tgz",
                                                                                                    "integrity":
"sha512-sPU4uV7dYlvtWJxwwxHD0PuihVNiE7TyAbQ5SWxDCB9mUYvOgroQOwYQQOKPJ8CIbE+1ETVlOoK1UC2nU3gYvg==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/mime-types": {
      "version": "2.1.35",
      "resolved": "https://registry.npmjs.org/mime-types/-/mime-types-2.1.35.tgz",
                                                                                                    "integrity":
"sha512-ZDY+bPm5zTTF+YpCrAU9nK0UgICYPT0QtT1NZWFv4s++TNkcgVaT0g6+4R2uI4MjQjzysHB1zxuWL50hzaeXiw==",
      "license": "MIT",
      "dependencies": {
        "mime-db": "1.52.0"
      },
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/minimatch": {
      "version": "3.1.2",
      "resolved": "https://registry.npmjs.org/minimatch/-/minimatch-3.1.2.tgz",
                                                                                                    "integrity":
"sha512-J7p63hRiAjw1NDEww1W7i37+ByIrOWO5XQQAzZ3VOcL0PNybwpfmV/N05zFAzwQ9USyEcX6t3UO+K5aqBQOIHw==",
      "dev": true,
      "license": "ISC",
      "dependencies": {
        "brace-expansion": "^1.1.7"
      },
      "engines": {
        "node": "*"
      }
    },
    "node_modules/ms": {
      "version": "2.1.3",
      "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
                                                                                                    "integrity":
"sha512-6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTlA==",
      "license": "MIT"
    },
    "node_modules/mysql2": {
      "version": "3.14.1",
      "resolved": "https://registry.npmjs.org/mysql2/-/mysql2-3.14.1.tgz",
                                                                                                    "integrity":
"sha512-7ytuPQJjQB8TNAYX/H2yhL+iQOnIBjAMam361R7UAL0lOVXWjtdrmoL9HYKqKoLp/8UUTRcvo1QPvK9KL7wA8w==",
      "license": "MIT",
      "dependencies": {
        "aws-ssl-profiles": "^1.1.1",
        "denque": "^2.1.0",
        "generate-function": "^2.3.1",
        "iconv-lite": "^0.6.3",
        "long": "^5.2.1",
        "lru.min": "^1.0.0",
        "named-placeholders": "^1.1.3",
        "seq-queue": "^0.0.5",
        "sqlstring": "^2.3.2"
      },
      "engines": {
        "node": ">= 8.0"
      }
    },
    "node_modules/mysql2/node_modules/iconv-lite": {
      "version": "0.6.3",
      "resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.6.3.tgz",
                                                                                                    "integrity":
"sha512-4fCk79wshMdzMp2rH06qWrJE4iolqLhCUH+OiuIgU++RB0+94NlDL81atO7GX55uUKueo0txHNtvEyI6D7WdMw==",
      "license": "MIT",
      "dependencies": {
        "safer-buffer": ">= 2.1.2 < 3.0.0"
      },
      "engines": {
        "node": ">=0.10.0"
      }
    },
    "node_modules/named-placeholders": {
      "version": "1.1.3",
      "resolved": "https://registry.npmjs.org/named-placeholders/-/named-placeholders-1.1.3.tgz",
                                                                                                    "integrity":
"sha512-eLoBxg6wE/rZkJPhU/xRX1WTpkFEwDJEN96oxFrTsqBdbT5ec295Q+CoHrL9IT0DipqKhmGcaZmwOt8OON5x1w==",
      "license": "MIT",
      "dependencies": {
        "lru-cache": "^7.14.1"
      },
      "engines": {
        "node": ">=12.0.0"
      }
    },
    "node_modules/negotiator": {
      "version": "0.6.3",
      "resolved": "https://registry.npmjs.org/negotiator/-/negotiator-0.6.3.tgz",
                                                                                                    "integrity":
"sha512-+EUsqGPLsM+j/zdChZjsnX51g4XrHFOIXwfnCVPGlQk/k5giakcKsuxCObBRu6DSm9opw/O6slWbJdghQM4bBg==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/nodemon": {
      "version": "3.1.10",
      "resolved": "https://registry.npmjs.org/nodemon/-/nodemon-3.1.10.tgz",
                                                                                                    "integrity":
"sha512-WDjw3pJ0/0jMFmyNDp3gvY2YizjLmmOUQo6DEBY+JgdvW/yQ9mEeSw6H5ythl5Ny2ytb7f9C2nIbjSxMNzbJXw==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "chokidar": "^3.5.2",
        "debug": "^4",
        "ignore-by-default": "^1.0.1",
        "minimatch": "^3.1.2",
        "pstree.remy": "^1.1.8",
        "semver": "^7.5.3",
        "simple-update-notifier": "^2.0.0",
        "supports-color": "^5.5.0",
        "touch": "^3.1.0",
        "undefsafe": "^2.0.5"
      },
      "bin": {
        "nodemon": "bin/nodemon.js"
      },
      "engines": {
        "node": ">=10"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/nodemon"
      }
    },
    "node_modules/normalize-path": {
      "version": "3.0.0",
      "resolved": "https://registry.npmjs.org/normalize-path/-/normalize-path-3.0.0.tgz",
                                                                                                    "integrity":
"sha512-6eZs5Ls3WtCisHWp9S2GUy8dqkpGi4BVSz3GaqiE6ezub0512ESztXUwUB6C6IKbQkY2Pnb/mD4WYojCRwcwLA==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=0.10.0"
      }
    },
    "node_modules/object-assign": {
      "version": "4.1.1",
      "resolved": "https://registry.npmjs.org/object-assign/-/object-assign-4.1.1.tgz",
                                                                                                    "integrity":
"sha512-rJgTQnkUnH1sFw8yT6VSU3zD3sWmu6sZhIseY8VX+GRu3P6F7Fu+JNDoXfklElbLJSnc3FUQHVe4cU5hj+BcUg==",
      "license": "MIT",
      "engines": {
        "node": ">=0.10.0"
      }
    },
    "node_modules/object-inspect": {
      "version": "1.13.4",
      "resolved": "https://registry.npmjs.org/object-inspect/-/object-inspect-1.13.4.tgz",
                                                                                                    "integrity":
"sha512-W67iLl4J2EXEGTbfeHCffrjDfitvLANg0UlX3wFUUSTx92KXRFegMHUVgSqE+wvhAbi4WqjGg9czysTV2Epbew==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.4"
      },
      "funding": {
        "url": "https://github.com/sponsors/ljharb"
      }
    },
    "node_modules/on-finished": {
      "version": "2.4.1",
      "resolved": "https://registry.npmjs.org/on-finished/-/on-finished-2.4.1.tgz",
                                                                                                    "integrity":
"sha512-oVlzkg3ENAhCk2zdv7IJwd/QUD4z2RxRwpkcGY8psCVcCYZNq4wYnVWALHM+brtuJjePWiYF/ClmuDr8Ch5+kg==",
      "license": "MIT",
      "dependencies": {
        "ee-first": "1.1.1"
      },
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/parseurl": {
      "version": "1.3.3",
      "resolved": "https://registry.npmjs.org/parseurl/-/parseurl-1.3.3.tgz",
                                                                                                    "integrity":
"sha512-CiyeOxFT/JZyN5m0z9PfXw4SCBJ6Sygz1Dpl0wqjlhDEGGBP1GnsUVEL0p63hoG1fcj3fHynXi9NYO4nWOL+qQ==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/path-to-regexp": {
      "version": "0.1.12",
      "resolved": "https://registry.npmjs.org/path-to-regexp/-/path-to-regexp-0.1.12.tgz",
                                                                                                    "integrity":
"sha512-RA1GjUVMnvYFxuqovrEqZoxxW5NUZqbwKtYz/Tt7nXerk0LbLblQmrsgdeOxV5SFHf0UDggjS/bSeOZwt1pmEQ==",
      "license": "MIT"
    },
    "node_modules/picomatch": {
      "version": "2.3.1",
      "resolved": "https://registry.npmjs.org/picomatch/-/picomatch-2.3.1.tgz",
                                                                                                    "integrity":
"sha512-JU3teHTNjmE2VCGFzuY8EXzCDVwEqB2a8fsIvwaStHhAWJEeVd1o1QD80CU6+ZdEXXSLbSsuLwJjkCBWqRQUVA==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=8.6"
      },
      "funding": {
        "url": "https://github.com/sponsors/jonschlinkert"
      }
    },
    "node_modules/proxy-addr": {
      "version": "2.0.7",
      "resolved": "https://registry.npmjs.org/proxy-addr/-/proxy-addr-2.0.7.tgz",
                                                                                                    "integrity":
"sha512-llQsMLSUDUPT44jdrU/O37qlnifitDP+ZwrmmZcoSKyLKvtZxpyV0n2/bD/N4tBAAZ/gJEdZU7KMraoK1+XYAg==",
      "license": "MIT",
      "dependencies": {
        "forwarded": "0.2.0",
        "ipaddr.js": "1.9.1"
      },
      "engines": {
        "node": ">= 0.10"
      }
    },
    "node_modules/pstree.remy": {
      "version": "1.1.8",
      "resolved": "https://registry.npmjs.org/pstree.remy/-/pstree.remy-1.1.8.tgz",
                                                                                                    "integrity":
"sha512-77DZwxQmxKnu3aR542U+X8FypNzbfJ+C5XQDk3uWjWxn6151aIMGthWYRXTqT1E5oJvg+ljaa2OJi+VfvCOQ8w==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/qs": {
      "version": "6.13.0",
      "resolved": "https://registry.npmjs.org/qs/-/qs-6.13.0.tgz",
                                                                                                    "integrity":
"sha512-+38qI9SOr8tfZ4QmJNplMUxqjbe7LKvvZgWdExBOmd+egZTtjLB67Gu0HRX3u/XOq7UU2Nx6nsjvS16Z9uwfpg==",
      "license": "BSD-3-Clause",
      "dependencies": {
        "side-channel": "^1.0.6"
      },
      "engines": {
        "node": ">=0.6"
      },
      "funding": {
        "url": "https://github.com/sponsors/ljharb"
      }
    },
    "node_modules/range-parser": {
      "version": "1.2.1",
      "resolved": "https://registry.npmjs.org/range-parser/-/range-parser-1.2.1.tgz",
                                                                                                   "integrity":
"sha512-Hrgsx+orqoygnmhFbKaHE6c296J+HTAQXoxEF6gNupROmmGJRoyzfG3ccAveqCBrwr/2yxQ5BVd/GTl5agOwSg==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/raw-body": {
      "version": "2.5.2",
      "resolved": "https://registry.npmjs.org/raw-body/-/raw-body-2.5.2.tgz",
                                                                                                   "integrity":
"sha512-8zGqypfENjCIqGhgXToC8aB2r7YrBX+AQAfIPs/Mlk+BtPTztOvTS01NRW/3Eh60J+a48lt8qsCzirQ6loCVfA==",
      "license": "MIT",
      "dependencies": {
        "bytes": "3.1.2",
        "http-errors": "2.0.0",
        "iconv-lite": "0.4.24",
        "unpipe": "1.0.0"
      },
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/readdirp": {
      "version": "3.6.0",
      "resolved": "https://registry.npmjs.org/readdirp/-/readdirp-3.6.0.tgz",
                                                                                                   "integrity":
"sha512-hOS089on8RduqdbhvQ5Z37A0ESjsqz6qnRcffsMU3495FuTdqSm+7bhJ29JvIOsBDEEnan5DPu9t3To9VRlMzA==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "picomatch": "^2.2.1"
      },
      "engines": {
        "node": ">=8.10.0"
      }
    },
    "node_modules/safe-buffer": {
      "version": "5.2.1",
      "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.2.1.tgz",
                                                                                                   "integrity":
"sha512-rp3So07KcdmmKbGvgaNxQSJr7bGVSVk5S9Eq1F+ppbRo70+YeaDxkw5Dd8NPN+GD6bjnYm2VuPuCXmpuYvmCXQ==",
      "funding": [
        {
          "type": "github",
          "url": "https://github.com/sponsors/feross"
        },
        {
          "type": "patreon",
          "url": "https://www.patreon.com/feross"
        },
        {
          "type": "consulting",
          "url": "https://feross.org/support"
        }
      ],
      "license": "MIT"
    },
    "node_modules/safer-buffer": {
      "version": "2.1.2",
      "resolved": "https://registry.npmjs.org/safer-buffer/-/safer-buffer-2.1.2.tgz",
                                                                                                   "integrity":
"sha512-YZo3K82SD7Riyi0E1EQPojLz7kpepnSQI9IyPbHHg1XXXevb5dJI7tpyN2ADxGcQbHG7vcyRHk0cbwqcQriUtg==",
      "license": "MIT"
    },
    "node_modules/semver": {
      "version": "7.7.2",
      "resolved": "https://registry.npmjs.org/semver/-/semver-7.7.2.tgz",
                                                                                                    "integrity":
"sha512-RF0Fw+rO5AMf9MAyaRXI4AV0Ulj5lMHqVxxdSgiVbixSCXoEmmX/jk0CuJw4+3SqroYO9VoUh+HcuJivvtJemA==",
      "license": "ISC",
      "bin": {
        "semver": "bin/semver.js"
      },
      "engines": {
        "node": ">=10"
      }
    },
    "node_modules/send": {
      "version": "0.19.0",
      "resolved": "https://registry.npmjs.org/send/-/send-0.19.0.tgz",
                                                                                                    "integrity":
"sha512-dW41u5VfLXu8SJh5bwRmyYUbAoSB3c9uQh6L8h/KtsFREPWpbX1lrljJo186Jc4nmci/sGUZ9a0a0J2zgfq2hw==",
      "license": "MIT",
      "dependencies": {
        "debug": "2.6.9",
        "depd": "2.0.0",
        "destroy": "1.2.0",
        "encodeurl": "~1.0.2",
        "escape-html": "~1.0.3",
        "etag": "~1.8.1",
        "fresh": "0.5.2",
        "http-errors": "2.0.0",
        "mime": "1.6.0",
        "ms": "2.1.3",
        "on-finished": "2.4.1",
        "range-parser": "~1.2.1",
        "statuses": "2.0.1"
      },
      "engines": {
        "node": ">= 0.8.0"
      }
    },
    "node_modules/send/node_modules/debug": {
      "version": "2.6.9",
      "resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",
                                                                                                    "integrity":
"sha512-bC7ElrdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUcNowp6cvspg0jXznRTKDjm/E7AdgFBVeAPVMNcKGsHMA==",
      "license": "MIT",
      "dependencies": {
        "ms": "2.0.0"
      }
    },
    "node_modules/send/node_modules/debug/node_modules/ms": {
      "version": "2.0.0",
      "resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",
                                                                                                    "integrity":
"sha512-Tpp60P6IUJDTuOq/5Z8cdskzJujfwqfOTkrwIwj7IRISpnkJnT6SyJ4PCPnGMoFjC9ddhal5KVIYtAt97ix05A==",
      "license": "MIT"
    },
    "node_modules/send/node_modules/encodeurl": {
      "version": "1.0.2",
      "resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-1.0.2.tgz",
                                                                                                    "integrity":
"sha512-TPJXq8JqFaVYm2CWmPvnP2Iyo4ZSM7/QKcSmuMLDObfpH5fi7RUGmd/rTDf+rut/saiDiQEeVTNgAmJEdAOx0w==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/seq-queue": {
      "version": "0.0.5",
      "resolved": "https://registry.npmjs.org/seq-queue/-/seq-queue-0.0.5.tgz",
                                                                                                    "integrity":
"sha512-hr3Wtp/GZIc/6DAGPDcV4/9WoZhjrkXsi5B/07QgX8tsdc6ilr7BFM6PM6rbdAX1kFSDYeZGLipIZZKyQP0O5Q=="
    },
    "node_modules/serve-static": {
      "version": "1.16.2",
      "resolved": "https://registry.npmjs.org/serve-static/-/serve-static-1.16.2.tgz",
                                                                                                    "integrity":
"sha512-VqpjJZKadQB/PEbEwvFdO43Ax5dFBZ2UECszz8bQ7pi7wt//PWe1P6MN7eCnjsatYtBT6EuiClbjSWP2WrIoTw==",
      "license": "MIT",
      "dependencies": {
        "encodeurl": "~2.0.0",
        "escape-html": "~1.0.3",
        "parseurl": "~1.3.3",
        "send": "0.19.0"
      },
      "engines": {
        "node": ">= 0.8.0"
      }
    },
    "node_modules/setprototypeof": {
      "version": "1.2.0",
      "resolved": "https://registry.npmjs.org/setprototypeof/-/setprototypeof-1.2.0.tgz",
                                                                                                    "integrity":
"sha512-E5LDX7Wrp85Kil5bhZv46j8jOeboKq5JMmYM3gVGdGH8xFpPWXUMsNrlODCrkoxMEeNi/XZIwuRvY4XNwYMJpw==",
      "license": "ISC"
    },
    "node_modules/side-channel": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/side-channel/-/side-channel-1.1.0.tgz",
                                                                                                    "integrity":
"sha512-ZX99e6tRweoUXqR+VBrslhda51Nh5MTQwou5tnUDgbtyM0dBgmhEDtWGP/xbKn6hqfPRHujUNwz5fy/wbbhnpw==",
      "license": "MIT",
      "dependencies": {
        "es-errors": "^1.3.0",
        "object-inspect": "^1.13.3",
        "side-channel-list": "^1.0.0",
        "side-channel-map": "^1.0.1",
        "side-channel-weakmap": "^1.0.2"
      },
      "engines": {
        "node": ">= 0.4"
      },
      "funding": {
        "url": "https://github.com/sponsors/ljharb"
      }
    },
    "node_modules/side-channel-list": {
      "version": "1.0.0",
      "resolved": "https://registry.npmjs.org/side-channel-list/-/side-channel-list-1.0.0.tgz",
                                                                                                    "integrity":
"sha512-FCLHtRD/gnpCiCHEiJLOwdmFP+wzCmDEkc9y7NsYxeF4u7Btsn1ZuwgwJGxImImHicJArLP4R0yX4c2KCrMrTA==",
      "license": "MIT",
      "dependencies": {
        "es-errors": "^1.3.0",
        "object-inspect": "^1.13.3"
      },
      "engines": {
        "node": ">= 0.4"
      },
      "funding": {
        "url": "https://github.com/sponsors/ljharb"
      }
    },
    "node_modules/side-channel-map": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/side-channel-map/-/side-channel-map-1.0.1.tgz",
                                                                                                    "integrity":
"sha512-VCjCNfgMsby3tTdo02nbjtM/ewra6jPHmpThenkTYh8pG9ucZ/1P8So4u4FGBek/BjpOVsDCMoLA/iuBKIFXRA==",
      "license": "MIT",
      "dependencies": {
        "call-bound": "^1.0.2",
        "es-errors": "^1.3.0",
        "get-intrinsic": "^1.2.5",
        "object-inspect": "^1.13.3"
      },
      "engines": {
        "node": ">= 0.4"
      },
      "funding": {
        "url": "https://github.com/sponsors/ljharb"
      }
    },
    "node_modules/side-channel-weakmap": {
      "version": "1.0.2",
      "resolved": "https://registry.npmjs.org/side-channel-weakmap/-/side-channel-weakmap-1.0.2.tgz",
                                                                                                    "integrity":
"sha512-WPS/HvHQTYnHisLo9McqBHOJk2FkHO/tlpvldyrnem4aeQp4hai3gythswg6p01oSoTl58rcpiFAjF2br2Ak2A==",
      "license": "MIT",
      "dependencies": {
        "call-bound": "^1.0.2",
        "es-errors": "^1.3.0",
        "get-intrinsic": "^1.2.5",
        "object-inspect": "^1.13.3",
        "side-channel-map": "^1.0.1"
      },
      "engines": {
        "node": ">= 0.4"
      },
      "funding": {
        "url": "https://github.com/sponsors/ljharb"
      }
    },
    "node_modules/simple-update-notifier": {
      "version": "2.0.0",
      "resolved": "https://registry.npmjs.org/simple-update-notifier/-/simple-update-notifier-2.0.0.tgz",
                                                                                                    "integrity":
"sha512-a2B9Y0KlNXl9u/vsW6sTIu9vGEpfKu2wRV6l1H3XEas/0gUIzGzBoP/IouTcUQbm9JWZLH3COxyn03TYlFax6w==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "semver": "^7.5.3"
      },
      "engines": {
        "node": ">=10"
      }
    },
    "node_modules/sqlstring": {
      "version": "2.3.3",
      "resolved": "https://registry.npmjs.org/sqlstring/-/sqlstring-2.3.3.tgz",
                                                                                                    "integrity":
"sha512-qC9iz2FlN7DQl3+wjwn3802RTyjCx7sDvfQEXchwa6CWOx07/WVfh91gBmQ9fahw8snwGEWU3xGzOt4tFyHLxg==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/statuses": {
      "version": "2.0.1",
      "resolved": "https://registry.npmjs.org/statuses/-/statuses-2.0.1.tgz",
                                                                                                    "integrity":
"sha512-RwNA9Z/7PrK06rYLIzFMlaF+l73iwpzsqRIFgbMLbTcLD6cOao82TaWefPXQvB2fOC4AjuYSEndS7N/mTCbkdQ==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/supports-color": {
      "version": "5.5.0",
      "resolved": "https://registry.npmjs.org/supports-color/-/supports-color-5.5.0.tgz",
      "integrity":
"sha512-QjVjwdXIt408MIiAqCX4oUKsgU2EqAGzs2Ppkm4aQYbjm+ZEWEcW4SfFNTr4uMNZma0ey4f5lgLrkB0aX0QMow==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "has-flag": "^3.0.0"
      },
      "engines": {
        "node": ">=4"
      }
    },
    "node_modules/to-regex-range": {
      "version": "5.0.1",
      "resolved": "https://registry.npmjs.org/to-regex-range/-/to-regex-range-5.0.1.tgz",
      "integrity":
"sha512-65P7iz6X5yEr1cwcgvQxbbIw7Uk3gOy5dIdtZ4rDveLqhrdJP+Li/Hx6tyK0NEb+2GCyneCMJiGqrADCSNk8sQ==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "is-number": "^7.0.0"
      },
      "engines": {
        "node": ">=8.0"
      }
    },
    "node_modules/toidentifier": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/toidentifier/-/toidentifier-1.0.1.tgz",
      "integrity":
"sha512-o5sSPKEkg/DIQNmH43V0/uerLrpzVedkUh8tGNvaeXpfpuwjKenlSox/2O/BTlZUtEe+JG7s5YhEz608PlAHRA==",
      "license": "MIT",
      "engines": {
        "node": ">=0.6"
      }
    },
    "node_modules/touch": {
      "version": "3.1.1",
      "resolved": "https://registry.npmjs.org/touch/-/touch-3.1.1.tgz",
      "integrity":
"sha512-r0eojU4bI8MnHr8c5bNo7lJDdI2qXlWWJk6a9EAFG7vbhTjElYhBVS3/miuE0uOuoLdb8Mc/rVfsmm6eo5o9GA==",
      "dev": true,
      "license": "ISC",
      "bin": {
        "nodetouch": "bin/nodetouch.js"
      }
    },
    "node_modules/ts-node": {
      "version": "10.9.2",
      "resolved": "https://registry.npmjs.org/ts-node/-/ts-node-10.9.2.tgz",
      "integrity":
"sha512-f0FFpIdcHgn8zcPSbf1dRevwt047YMnaiJM3u2w2RewrB+fob/zePZcrOyQoLMMO7aBIddLcQIEK5dYjkLnGrQ==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@cspotcode/source-map-support": "^0.8.0",
        "@tsconfig/node10": "^1.0.7",
        "@tsconfig/node12": "^1.0.7",
        "@tsconfig/node14": "^1.0.0",
        "@tsconfig/node16": "^1.0.2",
        "acorn": "^8.4.1",
        "acorn-walk": "^8.1.1",
        "arg": "^4.1.0",
        "create-require": "^1.1.0",
        "diff": "^4.0.1",
        "make-error": "^1.1.1",
        "v8-compile-cache-lib": "^3.0.1",
        "yn": "3.1.1"
      },
      "bin": {
        "ts-node": "dist/bin.js",
        "ts-node-cwd": "dist/bin-cwd.js",
        "ts-node-esm": "dist/bin-esm.js",
        "ts-node-script": "dist/bin-script.js",
        "ts-node-transpile-only": "dist/bin-transpile.js",
        "ts-script": "dist/bin-script-deprecated.js"
      },
      "peerDependencies": {
        "@swc/core": ">=1.2.50",
        "@swc/wasm": ">=1.2.50",
        "@types/node": "*",
        "typescript": ">=2.7"
      },
      "peerDependenciesMeta": {
        "@swc/core": {
          "optional": true
        },
        "@swc/wasm": {
          "optional": true
        }
      }
    },
    "node_modules/type-is": {
      "version": "1.6.18",
      "resolved": "https://registry.npmjs.org/type-is/-/type-is-1.6.18.tgz",
                                                                                                    "integrity":
"sha512-TkRKr9sUTxEH8MdfuCSP7VizJyzRNMjj2J2do2Jr3Kym598JVdEksuzPQCnlFPW4ky9Q+iA+ma9BGm06XQBy8g==",
      "license": "MIT",
      "dependencies": {
        "media-typer": "0.3.0",
        "mime-types": "~2.1.24"
      },
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/typescript": {
      "version": "5.8.3",
      "resolved": "https://registry.npmjs.org/typescript/-/typescript-5.8.3.tgz",
                                                                                                    "integrity":
"sha512-p1diW6TqL9L07nNxvRMM7hMMw4c5XOo/1ibL4aAIGmSAt9slTE1Xgw5KWuof2uTOvCg9BY7ZRi+GaF+7sfgPeQ==",
      "dev": true,
      "license": "Apache-2.0",
      "bin": {
        "tsc": "bin/tsc",
        "tsserver": "bin/tsserver"
      },
      "engines": {
        "node": ">=14.17"
      }
    },
    "node_modules/undefsafe": {
      "version": "2.0.5",
      "resolved": "https://registry.npmjs.org/undefsafe/-/undefsafe-2.0.5.tgz",
                                                                                                    "integrity":
```

```
"sha512-WxONCrssBM8TSPRqN5EmsjVrsv4A8X12J4ArBiiayv3DyyG3ZlIg6yysuuSYdZsVz3TKcTg2fd//Ujd4CHV1iA==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/undici-types": {
      "version": "6.21.0",
      "resolved": "https://registry.npmjs.org/undici-types/-/undici-types-6.21.0.tgz",
                                                                                                    "integrity":
"sha512-iwDZqg0QAGrg9Rav5H4n0M64c3mkR59cJ6wQp+7C4nI0gsmExaedaYLNO44eT4AtBBwjbTiGPMlt2Md0T9H9JQ==",
      "license": "MIT"
    },
    "node_modules/unpipe": {
      "version": "1.0.0",
      "resolved": "https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tgz",
                                                                                                    "integrity":
"sha512-pjy2bYhSsufwWlKwPc+l3cN7+wuJlK6uz0YdJEOlQDbl6jo/YlPi4mb8agUkVC8BF7V8NuzeyPNqRksA3hztKQ==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/utils-merge": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/utils-merge/-/utils-merge-1.0.1.tgz",
                                                                                                    "integrity":
"sha512-pMZTvIkT1d+TFGvDOqodOclx0QWkkgi6Tdoa8gC8ffGAAqz9pzPTZWAybbsHHoED/ztMtkv/VoYTYyShUn81hA==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.4.0"
      }
    },
    "node_modules/v8-compile-cache-lib": {
      "version": "3.0.1",
      "resolved": "https://registry.npmjs.org/v8-compile-cache-lib/-/v8-compile-cache-lib-3.0.1.tgz",
                                                                                                    "integrity":
"sha512-wa7YjyUGfNZngI/vtK0UHAN+lgDCxBPCylVXGp0zu59Fz5aiGtNXaq3DhIov063MorB+VfufLh3JlF2KdTK3xg==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/vary": {
      "version": "1.1.2",
      "resolved": "https://registry.npmjs.org/vary/-/vary-1.1.2.tgz",
                                                                                                    "integrity":
"sha512-BNGbWLfd0eUPabhkXUVm0j8uuvREyTh5ovRa/dyow/BqAbZJyC+5fU+IzQOzmAKzYqYRAISoRhdQr3eIZ/PXqg==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.8"
      }
    },
    "node_modules/yn": {
      "version": "3.1.1",
      "resolved": "https://registry.npmjs.org/yn/-/yn-3.1.1.tgz",
                                                                                                    "integrity":
"sha512-Ux4ygGWsu2c7isFWe8Yu1YluJmqVhxqK2cLXNQA5AcC3QfbGNpM7fu0Y8b/z16pXLnFxZYvWhd3fhBY9DLmC6Q==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=6"
      }
    }
  }
}
```

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\package.json

```
{
  "name": "be_wismon_keuangan",
  "version": "1.0.0",
  "description": "Backend API for Wismon Keuangan - Financial Management System",
  "main": "dist/index.js",
  "scripts": {
    "build": "tsc",
    "start": "node dist/index.js",
    "dev": "nodemon src/index.ts",
    "dev:watch": "nodemon --exec ts-node src/index.ts",
    "clean": "rimraf dist",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "nodejs",
    "typescript",
    "express",
    "api",
    "financial",
    "management"
  ],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "@types/bcryptjs": "^2.4.6",
    "@types/cors": "^2.8.19",
    "@types/jsonwebtoken": "^9.0.10",
    "bcryptjs": "^3.0.2",
    "cors": "^2.8.5",
    "dotenv": "^16.4.7",
    "express": "^4.21.2",
    "express-rate-limit": "^7.5.1",
    "helmet": "^8.1.0",
    "jsonwebtoken": "^9.0.2",
    "mysql2": "^3.11.5"
  },
  "devDependencies": {
    "@types/express": "^5.0.0",
    "@types/node": "^22.10.2",
    "nodemon": "^3.1.9",
    "ts-node": "^10.9.2",
    "typescript": "^5.7.2"
  }
}
```

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\README.md

```
# Wismon Keuangan Backend API

Backend API for Wismon Keuangan - Financial Management System built with Node.js, TypeScript, and Express.js.

## ¿ Getting Started

### Prerequisites

- Node.js (v16 or higher)
- npm or yarn package manager

### Installation

1. Navigate to the backend directory:
```

```bash
cd be_wismon_keuangan
```

2. Install dependencies:

```bash
npm install
```

### Development

Start the development server with hot reload:

```bash
npm run dev
```

The server will start on `http://localhost:3000`

### Available Scripts

- `npm run dev` - Start development server with nodemon
- `npm run dev:watch` - Start development server with ts-node
- `npm run build` - Build the project for production
- `npm run start` - Start the production server
- `npm run clean` - Remove build artifacts

### API Endpoints

- `GET /` - Welcome message and API status
- `GET /health` - Health check endpoint

## ¿¿ Tech Stack

- **Runtime**: Node.js
- **Language**: TypeScript
- **Framework**: Express.js
- **Development**: nodemon, ts-node

## ¿ Project Structure

```
be_wismon_keuangan/
¿¿¿ src/
¿   ¿¿¿ index.ts          # Application entry point
¿¿¿ dist/                 # Compiled JavaScript (after build)
¿¿¿ package.json
¿¿¿ tsconfig.json
¿¿¿ README.md
```

## ¿ Configuration

The application uses the following configuration:

- Port: 3000 (default) or `PORT` environment variable
- TypeScript target: ES2022
- Module system: CommonJS

## ¿ Development Notes

- Source files are located in the `src/` directory
- Compiled files are output to the `dist/` directory
- The development server automatically restarts when files change
- Strict TypeScript configuration is enabled for better code quality

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\tsconfig.json

```
{
  "compilerOptions": {
    /* Visit https://aka.ms/tsconfig to read more about this file */

    /* Projects */
    // "incremental": true,                              /* Save .tsbuildinfo files to allow for incremental
compilation of projects. */
    // "composite": true,                                /* Enable constraints that allow a TypeScript project
to be used with project references. */
    // "tsBuildInfoFile": "./.tsbuildinfo",              /* Specify the path to .tsbuildinfo incremental
compilation file. */
    // "disableSourceOfProjectReferenceRedirect": true,  /* Disable preferring source files instead of
declaration files when referencing composite projects. */
    // "disableSolutionSearching": true,                 /* Opt a project out of multi-project reference
checking when editing. */
    // "disableReferencedProjectLoad": true,             /* Reduce the number of projects loaded automatically
by TypeScript. */

    /* Language and Environment */
    "target": "ES2022" /* Set the JavaScript language version for emitted JavaScript and include compatible
library declarations. */,
    "lib": [
      "ES2022"
    ] /* Specify a set of bundled library declaration files that describe the target runtime environment. */,
    // "jsx": "preserve",                                /* Specify what JSX code is generated. */
    // "libReplacement": true,                           /* Enable lib replacement. */
    // "experimentalDecorators": true,                   /* Enable experimental support for legacy experimental
decorators. */
    // "emitDecoratorMetadata": true,                    /* Emit design-type metadata for decorated
declarations in source files. */
    // "jsxFactory": "",                                 /* Specify the JSX factory function used when
targeting React JSX emit, e.g. 'React.createElement' or 'h'. */
    // "jsxFragmentFactory": "",                         /* Specify the JSX Fragment reference used for
fragments when targeting React JSX emit e.g. 'React.Fragment' or 'Fragment'. */
    // "jsxImportSource": "",                            /* Specify module specifier used to import the JSX
factory functions when using 'jsx: react-jsx*'. */
    // "reactNamespace": "",                             /* Specify the object invoked for 'createElement'.
This only applies when targeting 'react' JSX emit. */
    // "noLib": true,                                    /* Disable including any library files, including the
default lib.d.ts. */
    // "useDefineForClassFields": true,                  /* Emit ECMAScript-standard-compliant class fields. */
    // "moduleDetection": "auto",                        /* Control what method is used to detect module-format
JS files. */

    /* Modules */
    "module": "commonjs" /* Specify what module code is generated. */,
    "rootDir": "./src" /* Specify the root folder within your source files. */,
    "moduleResolution": "node10" /* Specify how TypeScript looks up a file from a given module specifier. */,
    "baseUrl": "./" /* Specify the base directory to resolve non-relative module names. */,
    "paths": {
      "@/*": ["src/*"]
    } /* Specify a set of entries that re-map imports to additional lookup locations. */,
    // "rootDirs": [],                                   /* Allow multiple folders to be treated as one when
resolving modules. */
    // "typeRoots": [],                                  /* Specify multiple folders that act like
'./node_modules/@types'. */
    // "types": [],                                      /* Specify type package names to be included without
being referenced in a source file. */
    // "allowUmdGlobalAccess": true,                     /* Allow accessing UMD globals from modules. */
    // "moduleSuffixes": [],                             /* List of file name suffixes to search when resolving
a module. */
```

```
    // "allowImportingTsExtensions": true,                  /* Allow imports to include TypeScript file
extensions. Requires '--moduleResolution bundler' and either '--noEmit' or '--emitDeclarationOnly' to be set.
*/
    // "rewriteRelativeImportExtensions": true,            /* Rewrite '.ts', '.tsx', '.mts', and '.cts' file
extensions in relative import paths to their JavaScript equivalent in output files. */
    // "resolvePackageJsonExports": true,                /* Use the package.json 'exports' field when resolving
package imports. */
    // "resolvePackageJsonImports": true,                /* Use the package.json 'imports' field when resolving
imports. */
    // "customConditions": [],                                /* Conditions to set in addition to the
resolver-specific defaults when resolving imports. */
    // "noUncheckedSideEffectImports": true,             /* Check side effect imports. */
    "resolveJsonModule": true /* Enable importing .json files. */,
    // "allowArbitraryExtensions": true,                 /* Enable importing files with any extension, provided
a declaration file is present. */
    // "noResolve": true,                                  /* Disallow 'import's, 'require's or '<reference>'s
from expanding the number of files TypeScript should add to a project. */

    /* JavaScript Support */
    // "allowJs": true,                                        /* Allow JavaScript files to be a part of your
program. Use the 'checkJS' option to get errors from these files. */
    // "checkJs": true,                                  /* Enable error reporting in type-checked JavaScript
files. */
    // "maxNodeModuleJsDepth": 1,                        /* Specify the maximum folder depth used for checking
JavaScript files from 'node_modules'. Only applicable with 'allowJs'. */

    /* Emit */
    // "declaration": true,                              /* Generate .d.ts files from TypeScript and JavaScript
files in your project. */
    // "declarationMap": true,                           /* Create sourcemaps for d.ts files. */
    // "emitDeclarationOnly": true,                      /* Only output d.ts files and not JavaScript files. */
    // "sourceMap": true,                                    /* Create source map files for emitted JavaScript
files. */
    // "inlineSourceMap": true,                                 /* Include sourcemap files inside the emitted
JavaScript. */
    // "noEmit": true,                                   /* Disable emitting files from a compilation. */
    // "outFile": "./",                                  /* Specify a file that bundles all outputs into one
JavaScript file. If 'declaration' is true, also designates a file that bundles all .d.ts output. */
    "outDir": "./dist" /* Specify an output folder for all emitted files. */,
    "removeComments": true /* Disable emitting comments. */,
    // "importHelpers": true,                            /* Allow importing helper functions from tslib once
per project, instead of including them per-file. */
    // "downlevelIteration": true,                              /* Emit more compliant, but verbose and less
performant JavaScript for iteration. */
    // "sourceRoot": "",                                  /* Specify the root path for debuggers to find the
reference source code. */
    // "mapRoot": "",                                    /* Specify the location where debugger should locate
map files instead of generated locations. */
    // "inlineSources": true,                            /* Include source code in the sourcemaps inside the
emitted JavaScript. */
    // "emitBOM": true,                                  /* Emit a UTF-8 Byte Order Mark (BOM) in the beginning
of output files. */
    // "newLine": "crlf",                                /* Set the newline character for emitting files. */
    // "stripInternal": true,                            /* Disable emitting declarations that have '@internal'
in their JSDoc comments. */
    // "noEmitHelpers": true,                            /* Disable generating custom helper functions like
'__extends' in compiled output. */
    // "noEmitOnError": true,                            /* Disable emitting files if any type checking errors
are reported. */
    // "preserveConstEnums": true,                            /* Disable erasing 'const enum' declarations in
generated code. */
    // "declarationDir": "./",                                  /* Specify the output directory for generated
declaration files. */

    /* Interop Constraints */
    // "isolatedModules": true,                              /* Ensure that each file can be safely transpiled
```

```
without relying on other imports. */
    // "verbatimModuleSyntax": true,                     /* Do not transform or elide any imports or exports
not marked as type-only, ensuring they are written in the output file's format based on the 'module' setting.
*/
    // "isolatedDeclarations": true,                     /* Require sufficient annotation on exports so other
tools can trivially generate declaration files. */
    // "erasableSyntaxOnly": true,                       /* Do not allow runtime constructs that are not part
of ECMAScript. */
    // "allowSyntheticDefaultImports": true,             /* Allow 'import x from y' when a module doesn't have
a default export. */
    "esModuleInterop": true /* Emit additional JavaScript to ease support for importing CommonJS modules. This
enables 'allowSyntheticDefaultImports' for type compatibility. */,
    // "preserveSymlinks": true,                         /* Disable resolving symlinks to their realpath. This
correlates to the same flag in node. */
    "forceConsistentCasingInFileNames": true /* Ensure that casing is correct in imports. */,

    /* Type Checking */
    "strict": true /* Enable all strict type-checking options. */,
    "noImplicitAny": true /* Enable error reporting for expressions and declarations with an implied 'any'
type. */,
    "strictNullChecks": true /* When type checking, take into account 'null' and 'undefined'. */,
    "strictFunctionTypes": true /* When assigning functions, check to ensure parameters and the return values
are subtype-compatible. */,
    // "strictBindCallApply": true,                      /* Check that the arguments for 'bind', 'call', and
'apply' methods match the original function. */
    // "strictPropertyInitialization": true,             /* Check for class properties that are declared but
not set in the constructor. */
    // "strictBuiltinIteratorReturn": true,              /* Built-in iterators are instantiated with a
'TReturn' type of 'undefined' instead of 'any'. */
    // "noImplicitThis": true,                           /* Enable error reporting when 'this' is given the
type 'any'. */
    // "useUnknownInCatchVariables": true,               /* Default catch clause variables as 'unknown' instead
of 'any'. */
    // "alwaysStrict": true,                             /* Ensure 'use strict' is always emitted. */
    // "noUnusedLocals": true,                           /* Enable error reporting when local variables aren't
read. */
    // "noUnusedParameters": true,                       /* Raise an error when a function parameter isn't
read. */
    // "exactOptionalPropertyTypes": true,               /* Interpret optional property types as written,
rather than adding 'undefined'. */
    "noImplicitReturns": true /* Enable error reporting for codepaths that do not explicitly return in a
function. */,
    "noFallthroughCasesInSwitch": true /* Enable error reporting for fallthrough cases in switch statements.
*/,
    "noUncheckedIndexedAccess": true /* Add 'undefined' to a type when accessed using an index. */,
    // "noImplicitOverride": true,                       /* Ensure overriding members in derived classes are
marked with an override modifier. */
    // "noPropertyAccessFromIndexSignature": true,       /* Enforces using indexed accessors for keys declared
using an indexed type. */
    // "allowUnusedLabels": true,                        /* Disable error reporting for unused labels. */
    // "allowUnreachableCode": true,                     /* Disable error reporting for unreachable code. */

    /* Completeness */
    // "skipDefaultLibCheck": true,                      /* Skip type checking .d.ts files that are included
with TypeScript. */
    "skipLibCheck": true /* Skip type checking all .d.ts files. */
  },
  "include": ["src/**/*"],
  "exclude": ["node_modules", "dist", "**/*.test.ts", "**/*.spec.ts"]
}
```

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\src\index.ts

```
import express from "express";
import cors from "cors";
import dotenv from "dotenv";
import { testConnections } from "./config/database";
import authRoutes from "./routes/authRoutes";
import paymentRoutes from "./routes/paymentRoutes";

// Load environment variables
dotenv.config();

const app = express();
const PORT = process.env.PORT || 3000;
const CORS_ORIGIN = process.env.CORS_ORIGIN || "http://localhost:3000";

// Middleware - Allow multiple origins for development
const allowedOrigins = [
  "http://localhost:3000",
  "http://localhost:8080",
  "http://localhost:54128",
  /^http:\/\/localhost:\d+$/, // Allow any localhost port
];

app.use(
  cors({
    origin: (origin, callback) => {
      // Allow requests with no origin (like mobile apps or curl requests)
      if (!origin) return callback(null, true);

      // Check if origin is in allowed list or matches localhost pattern
      const isAllowed = allowedOrigins.some((allowedOrigin) => {
        if (typeof allowedOrigin === "string") {
          return allowedOrigin === origin;
        } else {
          return allowedOrigin.test(origin);
        }
      });

      if (isAllowed) {
        callback(null, true);
      } else {
        callback(new Error("Not allowed by CORS"));
      }
    },
    credentials: true,
    methods: ["GET", "POST", "PUT", "DELETE", "OPTIONS"],
    allowedHeaders: ["Content-Type", "Authorization"],
  })
);
app.use(express.json({ limit: "10mb" }));
app.use(express.urlencoded({ extended: true, limit: "10mb" }));

// Request logging middleware
app.use((req, res, next) => {
  console.log(`${new Date().toISOString()} - ${req.method} ${req.path}`);
  next();
});

// Routes
app.use("/api/auth", authRoutes);
app.use("/api/payments", paymentRoutes);

// Health check endpoint
app.get("/health", (req, res) => {
  res.json({
    status: "healthy",
    uptime: process.uptime(),
    timestamp: new Date().toISOString(),
    environment: process.env.NODE_ENV || "development",
  });
});

// Basic route
app.get("/", (req, res) => {
  res.json({
    message: "Wismon Keuangan Backend API",
    status: "running",
    timestamp: new Date().toISOString(),
    version: "1.0.0",
    endpoints: {
      auth: {
        login: "POST /api/auth/login",
        profile: "GET /api/auth/profile",
        verify: "POST /api/auth/verify",
      },
      payments: {
        history: "GET /api/payments/history",
        summary: "GET /api/payments/summary",
        detail: "GET /api/payments/detail/:id",
        refresh: "POST /api/payments/refresh",
        types: "GET /api/payments/types",
      },
    },
  });
});

// Error handling middleware
app.use(
  (
    error: any,
    req: express.Request,
    res: express.Response,
    next: express.NextFunction
  ) => {
    console.error("Unhandled error:", error);
    res.status(500).json({
      success: false,
      message: "Internal server error",
      errors: [
        process.env.NODE_ENV === "development"
          ¿ error.message
          : "Something went wrong",
      ],
    });
  }
);

// 404 handler
app.use("*", (req, res) => {
  res.status(404).json({
    success: false,
    message: "Endpoint not found",
    errors: [`${req.method} ${req.originalUrl} not found`],
  });
});

// Start server function
async function startServer() {
  try {
    console.log("¿ Starting Wismon Keuangan Backend Server...");
    console.log(`¿ Environment: ${process.env.NODE_ENV || "development"}`);

    // Test database connections
    console.log("¿ Testing database connections...");
    const dbResults = await testConnections();

    const successCount = Object.values(dbResults).filter(Boolean).length;
    const totalCount = Object.keys(dbResults).length;

    if (successCount === totalCount) {
      console.log("¿ All database connections successful!");
    } else {
      console.log(
        `¿¿  ${successCount}/${totalCount} database connections successful`
      );
      console.log("Database connection results:", dbResults);
    }

    // Start the server
    app.listen(PORT, () => {
      console.log("¿ Server started successfully!");
      console.log(`¿ API endpoint: http://localhost:${PORT}`);
      console.log(`¿ API documentation: http://localhost:${PORT}`);
      console.log(`¿ Health check: http://localhost:${PORT}/health`);
      console.log("");
      console.log("Available Endpoints:");
      console.log("  Authentication:");
      console.log("    POST /api/auth/login      - Student login");
      console.log("    GET  /api/auth/profile    - Get student profile");
      console.log("    POST /api/auth/verify     - Verify JWT token");
      console.log("  Payments:");
      console.log(
        "    GET  /api/payments/history     - Payment history with filters"
      );
      console.log("    GET  /api/payments/summary     - Payment summary/recap");
      console.log("    GET  /api/payments/detail/:id  - Transaction details");
      console.log("    POST /api/payments/refresh     - Refresh payment data");
      console.log(
        "    GET  /api/payments/types       - Available payment types"
      );
      console.log("");
      console.log("¿ Ready to accept requests!");
    });
  } catch (error) {
    console.error("¿ Failed to start server:", error);
    process.exit(1);
  }
}

// Handle graceful shutdown
process.on("SIGTERM", () => {
  console.log("¿ SIGTERM received. Shutting down gracefully...");
  process.exit(0);
});

process.on("SIGINT", () => {
  console.log("¿ SIGINT received. Shutting down gracefully...");
  process.exit(0);
});

// Handle unhandled promise rejections
process.on("unhandledRejection", (reason, promise) => {
  console.error("¿ Unhandled Rejection at:", promise, "reason:", reason);
  process.exit(1);
});

// Start the server
startServer();
```

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\src\config\database.ts

```
import mysql from "mysql2/promise";
import dotenv from "dotenv";

dotenv.config();

export interface DatabaseConfig {
  host: string;
  port: number;
  user: string;
  password: string;
  database: string;
  connectionLimit: number;
  acquireTimeout: number;
  timeout: number;
}

// SSO Database Configuration
const ssoDbConfig: DatabaseConfig = {
  host: process.env.DB_SSO_HOST || "127.0.0.1",
  port: parseInt(process.env.DB_SSO_PORT || "3306"),
  user: process.env.DB_SSO_USER || "root",
  password: process.env.DB_SSO_PASSWORD || "",
  database: process.env.DB_SSO_NAME || "wirahusada_sso",
  connectionLimit: 10,
  acquireTimeout: 60000,
  timeout: 60000,
};

// WIS Database Configuration
const wisDbConfig: DatabaseConfig = {
  host: process.env.DB_WIS_HOST || "127.0.0.1",
  port: parseInt(process.env.DB_WIS_PORT || "3306"),
  user: process.env.DB_WIS_USER || "root",
  password: process.env.DB_WIS_PASSWORD || "",
  database: process.env.DB_WIS_NAME || "wirahusada_wis",
  connectionLimit: 10,
  acquireTimeout: 60000,
  timeout: 60000,
};

// WISMON Database Configuration
const wismonDbConfig: DatabaseConfig = {
  host: process.env.DB_WISMON_HOST || "127.0.0.1",
  port: parseInt(process.env.DB_WISMON_PORT || "3306"),
  user: process.env.DB_WISMON_USER || "root",
  password: process.env.DB_WISMON_PASSWORD || "",
  database: process.env.DB_WISMON_NAME || "wirahusada_wismon",
  connectionLimit: 10,
  acquireTimeout: 60000,
  timeout: 60000,
};

// Create connection pools
export const ssoPool = mysql.createPool(ssoDbConfig);
export const wisPool = mysql.createPool(wisDbConfig);
export const wismonPool = mysql.createPool(wismonDbConfig);

// Test database connections
export const testConnections = async () => {
  const results = {
    sso: false,
    wis: false,
    wismon: false,
  };
  try {
    const ssoConnection = await ssoPool.getConnection();
    console.log("¿ SSO Database connected successfully");
    console.log(
      `¿ SSO Connected to: ${ssoDbConfig.host}:${ssoDbConfig.port}/${ssoDbConfig.database}`
    );
    ssoConnection.release();
    results.sso = true;
  } catch (error) {
    console.error("¿ SSO Database connection failed:", error);
  }

  try {
    const wisConnection = await wisPool.getConnection();
    console.log("¿ WIS Database connected successfully");
    console.log(
      `¿ WIS Connected to: ${wisDbConfig.host}:${wisDbConfig.port}/${wisDbConfig.database}`
    );
    wisConnection.release();
    results.wis = true;
  } catch (error) {
    console.error("¿ WIS Database connection failed:", error);
  }

  try {
    const wismonConnection = await wismonPool.getConnection();
    console.log("¿ WISMON Database connected successfully");
    console.log(
      `¿ WISMON Connected to: ${wismonDbConfig.host}:${wismonDbConfig.port}/${wismonDbConfig.database}`
    );
    wismonConnection.release();
    results.wismon = true;
  } catch (error) {
    console.error("¿ WISMON Database connection failed:", error);
  }

  return results;
};

// Execute query helper functions
export const executeSsoQuery = async (query: string, params¿: any[]) => {
  try {
    const [results] = await ssoPool.execute(query, params);
    return results;
  } catch (error) {
    console.error("SSO Database query error:", error);
    throw error;
  }
};

export const executeWisQuery = async (query: string, params¿: any[]) => {
  try {
    const [results] = await wisPool.execute(query, params);
    return results;
  } catch (error) {
    console.error("WIS Database query error:", error);
    throw error;
  }
};

export const executeWismonQuery = async (query: string, params¿: any[]) => {
  console.log("¿¿ WISMON DB - Executing query:", {
    query: query.replace(/\s+/g, " ").trim(),
    params,
    paramsTypes: params¿.map((p) => ({ value: p, type: typeof p })),
    timestamp: new Date().toISOString(),
  });

  try {
    const [results] = await wismonPool.execute(query, params);

    console.log("¿¿ WISMON DB - Query successful:", {
      resultCount: Array.isArray(results) ¿ results.length : "Not array",
      resultType: typeof results,
      firstResult: Array.isArray(results) ¿ results[0] : results,
    });

    return results;
  } catch (error) {
    console.error("¿¿ WISMON DB - Query error:", {
      error,
      query: query.replace(/\s+/g, " ").trim(),
      params,
      errorMessage: error instanceof Error ¿ error.message : "Unknown error",
    });
    throw error;
  }
};

// Export pools for direct access if needed
export { ssoDbConfig, wisDbConfig, wismonDbConfig };

// Keep backward compatibility
export const pool = ssoPool; // Default to SSO for auth
export const executeQuery = executeSsoQuery;
```

File:

D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\src\controllers\authController.ts

```
import { Request, Response } from "express";
import { AuthService } from "../services/authService";
import { ApiResponse, LoginRequest } from "../types";

const authService = new AuthService();

export class AuthController {
  /**
   * POST /auth/login
   * Login endpoint for students
   */
  async login(req: Request, res: Response): Promise<void> {
    console.log("¿ AUTH LOGIN - Request received:", {
      body: req.body,
      headers: req.headers,
      timestamp: new Date().toISOString(),
    });

    try {
      const { namam_nim, nrm }: LoginRequest = req.body;

      console.log("¿ AUTH LOGIN - Extracted credentials:", {
        namam_nim,
        nrm,
        namam_nim_type: typeof namam_nim,
        nrm_type: typeof nrm,
      });

      if (!namam_nim || !nrm) {
```

```
      console.log("¿ AUTH LOGIN - Missing credentials");
      res.status(400).json({
        success: false,
        message: "Student name/NIM and NRM are required",
        errors: ["namam_nim and nrm fields are required"],
      } as ApiResponse);
      return;
    }

    console.log("¿ AUTH LOGIN - Attempting authentication...");
    const loginResult = await authService.login({ namam_nim, nrm });

    console.log("¿ AUTH LOGIN - Authentication successful:", {
      hasToken: !!loginResult.token,
      userNrm: loginResult.user¿.nrm,
      userNim: loginResult.user¿.nim,
    });

    res.status(200).json({
      success: true,
      message: "Login successful",
      data: loginResult,
    } as ApiResponse);
  } catch (error) {
    console.error("¿ AUTH LOGIN - Error:", error);

    const errorMessage =
      error instanceof Error ¿ error.message : "Login failed";
    const statusCode =
      errorMessage.includes("not found") || errorMessage.includes("invalid")
        ¿ 401
        : 500;

    res.status(statusCode).json({
      success: false,
      message: errorMessage,
      errors: [errorMessage],
    } as ApiResponse);
  }
}

/**
 * GET /auth/profile
 * Get current user profile (requires authentication)
 */
async getProfile(req: Request, res: Response): Promise<void> {
  try {
    const user = (req as any).user; // Set by authenticateToken middleware

    if (!user || !user.nrm) {
      res.status(401).json({
        success: false,
        message: "Authentication required",
        errors: ["User not found in request"],
      } as ApiResponse);
      return;
    }

    const student = await authService.getStudentProfile(user.nrm);

    if (!student) {
      res.status(404).json({
        success: false,
        message: "Student profile not found",
        errors: ["Student data not found"],
      } as ApiResponse);
```

```
        return;
      }

      res.status(200).json({
        success: true,
        message: "Profile retrieved successfully",
        data: {
          nrm: student.nrm,
          nim: student.nim,
          namam: student.namam,
          tgdaftar: student.tgdaftar,
          tplahir: student.tplahir,
        },
      } as ApiResponse);
    } catch (error) {
      console.error("Get profile error:", error);

      res.status(500).json({
        success: false,
        message: "Failed to retrieve profile",
        errors: [error instanceof Error ¿ error.message : "Unknown error"],
      } as ApiResponse);
    }
  }

  /**
   * POST /auth/verify
   * Verify JWT token
   */
  async verifyToken(req: Request, res: Response): Promise<void> {
    try {
      const user = (req as any).user; // Set by authenticateToken middleware

      res.status(200).json({
        success: true,
        message: "Token is valid",
        data: {
          nrm: user.nrm,
          nim: user.nim,
          namam: user.namam,
        },
      } as ApiResponse);
    } catch (error) {
      res.status(401).json({
        success: false,
        message: "Invalid token",
        errors: ["Token verification failed"],
      } as ApiResponse);
    }
  }
}
```

File:

D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\src\controllers\paymentController.ts

```
import { Request, Response } from "express";
import { PaymentService } from "../services/paymentService";
import { ApiResponse, PaymentHistoryQuery } from "../types";

const paymentService = new PaymentService();

export class PaymentController {
  /**
```

```
 * GET /payments/history
 * Get payment history with filtering and pagination
 */
async getPaymentHistory(req: Request, res: Response): Promise<void> {
  console.log("¿ PAYMENT HISTORY - Request received:", {
    query: req.query,
    timestamp: new Date().toISOString(),
  });

  try {
    const user = (req as any).user; // Set by authenticateToken middleware

    console.log("¿ PAYMENT HISTORY - User from token:", {
      nrm: user¿.nrm,
      nrm_type: typeof user¿.nrm,
      hasUser: !!user,
    });

    if (!user || !user.nrm) {
      console.log("¿ PAYMENT HISTORY - Authentication failed");
      res.status(401).json({
        success: false,
        message: "Authentication required",
        errors: ["User not found in request"],
      } as ApiResponse);
      return;
    }

    // Parse query parameters
    const filters: PaymentHistoryQuery = {
      page: parseInt(req.query.page as string) || 1,
      limit: parseInt(req.query.limit as string) || 20,
      startDate: req.query.startDate as string,
      endDate: req.query.endDate as string,
      type: req.query.type as string,
      sortBy:
        (req.query.sortBy as "tanggal" | "jumlah" | "type") || "tanggal",
      sortOrder: (req.query.sortOrder as "asc" | "desc") || "desc",
    };

    console.log("¿ PAYMENT HISTORY - Parsed filters:", {
      filters,
      userNrm: user.nrm,
      userNrmType: typeof user.nrm,
    });

    console.log("¿ PAYMENT HISTORY - Calling service...");
    const result = await paymentService.getPaymentHistory(user.nrm, filters);

    console.log("¿ PAYMENT HISTORY - Service result:", {
      dataCount: result¿.data¿.length || 0,
      total: result¿.total,
      page: result¿.page,
      limit: result¿.limit,
      firstItem: result¿.data¿.[0],
    });

    res.status(200).json({
      success: true,
      message: "Payment history retrieved successfully",
      data: result,
    } as ApiResponse);
  } catch (error) {
    console.error("¿ PAYMENT HISTORY - Error:", error);

    res.status(500).json({
      success: false,
      message: "Failed to retrieve payment history",
      errors: [error instanceof Error ¿ error.message : "Unknown error"],
    } as ApiResponse);
  }
}

/**
 * GET /payments/summary
 * Get payment summary/recapitulation
 */
async getPaymentSummary(req: Request, res: Response): Promise<void> {
  try {
    const user = (req as any).user; // Set by authenticateToken middleware

    if (!user || !user.nrm) {
      res.status(401).json({
        success: false,
        message: "Authentication required",
        errors: ["User not found in request"],
      } as ApiResponse);
      return;
    }

    const summary = await paymentService.getPaymentSummary(user.nrm);

    res.status(200).json({
      success: true,
      message: "Payment summary retrieved successfully",
      data: summary,
    } as ApiResponse);
  } catch (error) {
    console.error("Get payment summary error:", error);

    res.status(500).json({
      success: false,
      message: "Failed to retrieve payment summary",
      errors: [error instanceof Error ¿ error.message : "Unknown error"],
    } as ApiResponse);
  }
}

/**
 * GET /payments/detail/:id
 * Get detailed transaction information
 */
async getTransactionDetail(req: Request, res: Response): Promise<void> {
  try {
    const user = (req as any).user; // Set by authenticateToken middleware
    const transactionId = req.params.id;

    if (!user || !user.nrm) {
      res.status(401).json({
        success: false,
        message: "Authentication required",
        errors: ["User not found in request"],
      } as ApiResponse);
      return;
    }

    if (!transactionId) {
      res.status(400).json({
        success: false,
        message: "Transaction ID is required",
        errors: ["Transaction ID parameter is missing"],
      } as ApiResponse);
      return;
    }

    const detail = await paymentService.getTransactionDetail(
      transactionId,
      user.nrm
    );

    if (!detail) {
      res.status(404).json({
        success: false,
        message: "Transaction not found",
        errors: ["Transaction not found or does not belong to this user"],
      } as ApiResponse);
      return;
    }

    res.status(200).json({
      success: true,
      message: "Transaction detail retrieved successfully",
      data: detail,
    } as ApiResponse);
  } catch (error) {
    console.error("Get transaction detail error:", error);

    res.status(500).json({
      success: false,
      message: "Failed to retrieve transaction detail",
      errors: [error instanceof Error ¿ error.message : "Unknown error"],
    } as ApiResponse);
  }
}

/**
 * POST /payments/refresh
 * Refresh payment data
 */
async refreshPaymentData(req: Request, res: Response): Promise<void> {
  try {
    const user = (req as any).user; // Set by authenticateToken middleware

    if (!user || !user.nrm) {
      res.status(401).json({
        success: false,
        message: "Authentication required",
        errors: ["User not found in request"],
      } as ApiResponse);
      return;
    }

    const refreshed = await paymentService.refreshPaymentData(user.nrm);

    if (refreshed) {
      res.status(200).json({
        success: true,
        message: "Payment data refreshed successfully",
        data: { refreshed: true },
      } as ApiResponse);
    } else {
      res.status(400).json({
        success: false,
        message: "Failed to refresh payment data",
        errors: ["No payment data found for this user"],
      } as ApiResponse);
    }
  } catch (error) {
      console.error("Refresh payment data error:", error);

      res.status(500).json({
        success: false,
        message: "Failed to refresh payment data",
        errors: [error instanceof Error ¿ error.message : "Unknown error"],
      } as ApiResponse);
    }
  }

  /**
   * GET /payments/types
   * Get available payment types for filtering
   */
  async getPaymentTypes(req: Request, res: Response): Promise<void> {
    console.log("¿ PAYMENT TYPES - Request received:", {
      timestamp: new Date().toISOString(),
    });

    try {
      console.log("¿ PAYMENT TYPES - Calling service...");
      const paymentTypes = await paymentService.getPaymentTypes();

      console.log("¿ PAYMENT TYPES - Service result:", {
        count: paymentTypes¿.length || 0,
        types: paymentTypes,
      });

      res.status(200).json({
        success: true,
        message: "Payment types retrieved successfully",
        data: paymentTypes,
      } as ApiResponse);
    } catch (error) {
      console.error("¿ PAYMENT TYPES - Error:", error);

      res.status(500).json({
        success: false,
        message: "Failed to retrieve payment types",
        errors: [error instanceof Error ¿ error.message : "Unknown error"],
      } as ApiResponse);
    }
  }
}
```

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\src\routes\authRoutes.ts

```
import { Router } from "express";
import { AuthController } from "../controllers/authController";
import { authenticateToken } from "../utils/auth";

const router = Router();
const authController = new AuthController();

// Public routes
router.post("/login", authController.login.bind(authController));

// Protected routes (require authentication)
router.get(
  "/profile",
  authenticateToken,
  authController.getProfile.bind(authController)
);
router.post(
  "/verify",
  authenticateToken,
  authController.verifyToken.bind(authController)
);

export default router;
```

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\src\routes\paymentRoutes.ts

```
import { Router } from "express";
import { PaymentController } from "../controllers/paymentController";
import { authenticateToken } from "../utils/auth";

const router = Router();
const paymentController = new PaymentController();

// All payment routes require authentication
router.use(authenticateToken);

// Payment history with filtering and pagination
router.get(
  "/history",
  paymentController.getPaymentHistory.bind(paymentController)
);

// Payment summary/recapitulation
router.get(
  "/summary",
  paymentController.getPaymentSummary.bind(paymentController)
);

// Transaction detail
router.get(
  "/detail/:id",
  paymentController.getTransactionDetail.bind(paymentController)
);

// Refresh payment data
router.post(
  "/refresh",
  paymentController.refreshPaymentData.bind(paymentController)
);

// Get payment types for filtering
router.get("/types", paymentController.getPaymentTypes.bind(paymentController));

export default router;
```

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\src\services\authService.ts

```
import { executeSsoQuery, executeWisQuery } from "../config/database";
import { User, Student, LoginRequest, LoginResponse } from "../types";
import { generateToken } from "../utils/auth";

export class AuthService {
  /**
   * Find student by name or NIM and NRM
   */
  async findStudent(namam_nim: string, nrm: string): Promise<Student | null> {
    try {
      // Try to find student by NIM first
      let query = `
        SELECT nrm, nim, namam, tgdaftar, tplahir, kdagama
        FROM mahasiswa
        WHERE nim = ¿ AND nrm = ¿
      `;
      let results = (await executeWisQuery(query, [namam_nim, nrm])) as any[];

      if (results.length > 0) {
        return results[0] as Student;
      }

      // If not found by NIM, try by student name (exact match, case-insensitive)
      query = `
        SELECT nrm, nim, namam, tgdaftar, tplahir, kdagama
        FROM mahasiswa
        WHERE LOWER(namam) = LOWER(¿) AND nrm = ¿
      `;
      results = (await executeWisQuery(query, [namam_nim, nrm])) as any[];

      if (results.length > 0) {
        return results[0] as Student;
      }

      // Try with spaces removed (for cases like "HanikZaimatusSholichah" vs "Hanik Zaimatus Sholichah")
      // But still exact match after removing spaces
      const namam_nim_no_spaces = namam_nim.replace(/\s+/g, "");
      query = `
        SELECT nrm, nim, namam, tgdaftar, tplahir, kdagama
        FROM mahasiswa
        WHERE LOWER(REPLACE(namam, ' ', '')) = LOWER(¿) AND nrm = ¿
      `;
      results = (await executeWisQuery(query, [
        namam_nim_no_spaces,
        nrm,
      ])) as any[];

      if (results.length > 0) {
        return results[0] as Student;
      }

      return null;
    } catch (error) {
      console.error("Error finding student:", error);
      throw new Error("Database error while finding student");
    }
  }

  /**
   * Check if user exists in SSO system (optional - for future authentication)
   */
  async findSsoUser(username: string): Promise<User | null> {
    try {
      const query = `
        SELECT id, username, email, created_at
        FROM user
        WHERE username = ¿
      `;
      const results = (await executeSsoQuery(query, [username])) as any[];

      if (results.length > 0) {
        return results[0] as User;
      }

      return null;
    } catch (error) {
      console.error("Error finding SSO user:", error);
      return null; // Don't throw error - SSO is optional for now
    }
  }

  /**
   * Login student - validates student data and generates JWT token
   */
  async login(loginData: LoginRequest): Promise<LoginResponse> {
    const { namam_nim, nrm } = loginData;

    // Validate input
    if (!namam_nim || !nrm) {
      throw new Error("Student name/NIM and NRM are required");
    }

    // Find student in WIS database
    const student = await this.findStudent(namam_nim, nrm);

    if (!student) {
      throw new Error("Student not found or invalid credentials");
    }

    // Generate JWT token
    const token = generateToken({
      nrm: student.nrm,
      nim: student.nim,
      namam: student.namam,
    });

    return {
      token,
      user: {
        nrm: student.nrm,
        nim: student.nim,
        namam: student.namam,
      },
    };
  }

  /**
   * Get student profile by NRM (for authenticated requests)
   */
  async getStudentProfile(nrm: string): Promise<Student | null> {
    try {
      const query = `
        SELECT nrm, nim, namam, tgdaftar, tplahir, kdagama
        FROM mahasiswa
        WHERE nrm = ¿
      `;
      const results = (await executeWisQuery(query, [nrm])) as any[];

      if (results.length > 0) {
        return results[0] as Student;
      }

      return null;
    } catch (error) {
      console.error("Error getting student profile:", error);
      throw new Error("Database error while getting student profile");
    }
  }
}
```

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\src\services\paymentService.ts

```
import { executeWismonQuery, executeWisQuery } from "../config/database";
import {
  PaymentHistoryItem,
  PaymentSummary,
  TransactionDetail,
  PaymentHistoryQuery,
  Student,
} from "../types";

export class PaymentService {
  /**
   * Get payment history for a student with filtering and pagination
   */
  async getPaymentHistory(
    nrm: string,
    filters: PaymentHistoryQuery = {}
  ): Promise<{
    data: PaymentHistoryItem[];
    total: number;
    page: number;
    limit: number;
  }> {
    console.log("¿¿ PAYMENT SERVICE - getPaymentHistory called:", {
      nrm,
      nrmType: typeof nrm,
      filters,
      timestamp: new Date().toISOString(),
    });

    try {
      const {
        page = 1,
        limit = 20,
        startDate,
        endDate,
        type,
        sortBy = "tanggal",
        sortOrder = "desc",
      } = filters;

      const offset = (page - 1) * limit;

      console.log("¿¿ PAYMENT SERVICE - Processed parameters:", {
        page,
        limit,
        offset,
        startDate,
        endDate,
        type,
        sortBy,
        sortOrder,
      });

      // Build WHERE conditions
      const conditions = ["tpm.nrm = ¿"];
      const params = [nrm];

      if (startDate) {
        conditions.push("t.tanggal >= ¿");
        params.push(startDate);
      }

      if (endDate) {
        conditions.push("t.tanggal <= ¿");
        params.push(endDate);
      }
      if (type) {
        conditions.push("a_credit.nama LIKE ¿");
        params.push(`%${type}%`);
      }

      // Build ORDER BY clause
      let orderByClause = "";
      switch (sortBy) {
        case "tanggal":
          orderByClause = `ORDER BY t.tanggal ${sortOrder.toUpperCase()}, t.jam ${sortOrder.toUpperCase()}`;
          break;
        case "jumlah":
          orderByClause = `ORDER BY t.nominal ${sortOrder.toUpperCase()}`;
          break;
        case "type":
          orderByClause = `ORDER BY a_credit.nama ${sortOrder.toUpperCase()}`;
          break;
        default:
          orderByClause = `ORDER BY t.tanggal ${sortOrder.toUpperCase()}, t.jam ${sortOrder.toUpperCase()}`;
      }

      // Main query for payment history
      const query = `
        SELECT
          t.no as id,
          t.no as tx_id,
          DATE_FORMAT(t.tanggal, '%d %M %Y') as tanggal,
               DATE_FORMAT(CONCAT(t.tanggal, ' ', IFNULL(t.jam, '00:00:00')), '%d %M %Y, %H:%i:%s WIB') as
tanggal_full,
          a_credit.nama as type,
          CONCAT('Rp ', FORMAT(t.nominal, 0)) as jumlah,
          'LUNAS' as status,
          CONCAT(a_debit.kode, ' - ', a_debit.nama) as method,
          a_debit.kode as method_code
        FROM transaksi t
        INNER JOIN t_pembayaranmahasiswa tpm ON t.no = tpm.no
        LEFT JOIN akun a_debit ON t.kodeakundebit = a_debit.kode
        LEFT JOIN akun a_credit ON t.kodeakunkredit = a_credit.kode
        WHERE ${conditions.join(" AND ")}
        ${orderByClause}
        LIMIT ¿ OFFSET ¿
      `;

      params.push(limit.toString(), offset.toString());

      console.log("¿¿ PAYMENT SERVICE - Final query params:", {
        conditions: conditions.join(" AND "),
        params,
        paramsTypes: params.map((p) => typeof p),
      });

      // Count query for total records
      const countQuery = `
        SELECT COUNT(*) as total
        FROM transaksi t
        INNER JOIN t_pembayaranmahasiswa tpm ON t.no = tpm.no
        LEFT JOIN akun a_debit ON t.kodeakundebit = a_debit.kode
        LEFT JOIN akun a_credit ON t.kodeakunkredit = a_credit.kode
        WHERE ${conditions.join(" AND ")}
      `;

      const countParams = params.slice(0, -2); // Remove limit and offset

      console.log("¿¿ PAYMENT SERVICE - Executing queries:", {
        mainQuery: query.replace(/\s+/g, " ").trim(),
        countQuery: countQuery.replace(/\s+/g, " ").trim(),
        mainParams: params,
        countParams,
      });

      // Execute queries
      const [results, countResults] = await Promise.all([
        executeWismonQuery(query, params) as Promise<any[]>,
        executeWismonQuery(countQuery, countParams) as Promise<any[]>,
      ]);

      console.log("¿¿ PAYMENT SERVICE - Raw query results:", {
        resultsCount: results¿.length || 0,
        countResult: countResults¿.[0],
        firstResult: results¿.[0],
        allResults: results,
      });

      const total = (countResults as any[])[0]¿.total || 0;

      console.log("¿¿ PAYMENT SERVICE - Returning data:", {
        dataCount: results¿.length || 0,
        total,
        page,
        limit,
      });

      return {
        data: results as PaymentHistoryItem[],
        total,
        page,
        limit,
      };
    } catch (error) {
      console.error("Error getting payment history:", error);
      throw new Error("Database error while getting payment history");
    }
  }

  /**
   * Get payment summary/recapitulation for a student
   */
  async getPaymentSummary(nrm: string): Promise<PaymentSummary> {
    try {
      const query = `
        SELECT
          a_credit.nama as payment_type,
          SUM(t.nominal) as total_amount
        FROM transaksi t
        INNER JOIN t_pembayaranmahasiswa tpm ON t.no = tpm.no
        LEFT JOIN akun a_credit ON t.kodeakunkredit = a_credit.kode
        WHERE tpm.nrm = ¿
        GROUP BY a_credit.nama, a_credit.kode
        ORDER BY total_amount DESC
      `;

      const results = (await executeWismonQuery(query, [nrm])) as any[];

      const breakdown: { [key: string]: string } = {};
      let totalPayment = 0;

      results.forEach((row: any) => {
        const amount = parseFloat(row.total_amount) || 0;
        breakdown[row.payment_type] = `Rp ${amount.toLocaleString("id-ID")}`;
        totalPayment += amount;
      });
```

```
      return {
        total_pembayaran: `Rp ${totalPayment.toLocaleString("id-ID")}`,
        breakdown,
      };
    } catch (error) {
      console.error("Error getting payment summary:", error);
      throw new Error("Database error while getting payment summary");
    }
  }

  /**
   * Get detailed transaction information
   */
  async getTransactionDetail(
    transactionId: string,
    nrm: string
  ): Promise<TransactionDetail | null> {
    try {
      // Get transaction details
      const query = `
        SELECT
          t.no as id,
          t.no as tx_id,
          DATE_FORMAT(t.tanggal, '%d %M %Y') as tanggal,
          DATE_FORMAT(CONCAT(t.tanggal, ' ', IFNULL(t.jam, '00:00:00')), '%d %M %Y, %H:%i:%s WIB') as
tanggal_full,
          a_credit.nama as type,
          CONCAT('Rp ', FORMAT(t.nominal, 0)) as jumlah,
          'LUNAS' as status,
          CONCAT(a_debit.kode, ' - ', a_debit.nama) as method,
          a_debit.kode as method_code,
          tpm.prodi
        FROM transaksi t
        INNER JOIN t_pembayaranmahasiswa tpm ON t.no = tpm.no
        LEFT JOIN akun a_debit ON t.kodeakundebit = a_debit.kode
        LEFT JOIN akun a_credit ON t.kodeakunkredit = a_credit.kode
        WHERE t.no = ¿ AND tpm.nrm = ¿
      `;

      const results = (await executeWismonQuery(query, [
        transactionId,
        nrm,
      ])) as any[];

      if (results.length === 0) {
        return null;
      }

      const transaction = results[0];

      // Get student information
      const studentQuery = `
        SELECT nrm, nim, namam
        FROM mahasiswa
        WHERE nrm = ¿
      `;

      const studentResults = (await executeWisQuery(studentQuery, [
        nrm,
      ])) as any[];
      const student = studentResults[0] || {};

      // Build transaction detail
      const transactionDetail: TransactionDetail = {
        id: transaction.id,
        tanggal: transaction.tanggal,
        tanggal_full: transaction.tanggal_full,
        type: transaction.type,
        jumlah: transaction.jumlah,
        status: transaction.status,
        tx_id: transaction.tx_id,
        method: transaction.method,
        method_code: transaction.method_code,
        student_name: student.namam || "Unknown",
        student_nim: student.nim || "Unknown",
        student_prodi: transaction.prodi || "Unknown",
        payment_breakdown: {
          [transaction.type]: transaction.jumlah,
        },
      };

      return transactionDetail;
    } catch (error) {
      console.error("Error getting transaction detail:", error);
      throw new Error("Database error while getting transaction detail");
    }
  }

  /**
   * Refresh payment data (mainly for triggering cache refresh if needed)
   */
  async refreshPaymentData(nrm: string): Promise<boolean> {
    try {
      // This could include cache clearing logic in the future
      // For now, just verify the student exists
      const query = `
        SELECT COUNT(*) as count
        FROM t_pembayaranmahasiswa
        WHERE nrm = ¿
      `;

      const results = (await executeWismonQuery(query, [nrm])) as any[];
      return results.length > 0;
    } catch (error) {
      console.error("Error refreshing payment data:", error);
      return false;
    }
  }

  /**
   * Get available payment types from the database
   */
  async getPaymentTypes(): Promise<{ kode: string; nama: string }[]> {
    try {
      const query = `
        SELECT DISTINCT a.kode, a.nama
        FROM akun a
        INNER JOIN transaksi t ON t.kodeakunkredit = a.kode
        INNER JOIN t_pembayaranmahasiswa tpm ON t.no = tpm.no
        WHERE a.kode >= 400 AND a.kode < 500
        ORDER BY a.kode
      `;

      console.log("¿¿ PAYMENT SERVICE - Getting payment types from akun table");

      const results = (await executeWismonQuery(query, [])) as any[];

      console.log("¿¿ PAYMENT SERVICE - Payment types retrieved:", {
        count: results¿.length || 0,
        types: results,
      });
```

```
      return results as { kode: string; nama: string }[];
    } catch (error) {
      console.error("Error getting payment types:", error);
      throw new Error("Database error while getting payment types");
    }
  }
}
```

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\src\types\index.ts

```
// User and Authentication Types
export interface User {
  id: string;
  username: string;
  email¿: string;
  created_at¿: Date;
}

export interface UserRole {
  username: string;
  app: string;
  rid: string;
}

export interface Student {
  nrm: string;
  nim: string;
  namam: string;
  tgdaftar¿: Date;
  tplahir¿: Date;
  kdagama¿: string;
  email¿: string;
}

export interface Employee {
  nip: string;
  nama: string;
  kdagama¿: string;
}

// Payment Types
export interface PaymentTransaction {
  id: string;
  kode_transaksi: string;
  tanggal: Date;
  total: number;
  status: string;
  jenistransaksi_id¿: string;
  akun_debit¿: string;
  akun_kredit¿: string;
}

export interface PaymentType {
  id: string;
  kode: string;
  nama_jenis: string;
}

export interface Account {
  id: string;
  kode: string;
  nama_akun: string;
}
```

```
export interface StudentPayment {
  no: string;
  nrm: string;
  prodi¿: string;
  semester¿: string;
  tahun_akademik¿: string;
  jumlah: number;
  tanggal_bayar: Date;
}

// API Response Types
export interface ApiResponse<T = any> {
  success: boolean;
  message: string;
  data¿: T;
  errors¿: string[];
}

export interface LoginRequest {
  namam_nim: string; // Can be student name or NIM
  nrm: string;
}

export interface LoginResponse {
  token: string;
  user: {
    nrm: string;
    nim: string;
    namam: string;
  };
}

export interface PaymentHistoryQuery {
  page¿: number;
  limit¿: number;
  startDate¿: string;
  endDate¿: string;
  type¿: string;
  sortBy¿: "tanggal" | "jumlah" | "type";
  sortOrder¿: "asc" | "desc";
}

export interface PaymentHistoryItem {
  id: string;
  tanggal: string;
  tanggal_full: string;
  type: string;
  jumlah: string;
  status: string;
  tx_id: string;
  method: string;
  method_code: string;
}

export interface PaymentSummary {
  total_pembayaran: string;
  breakdown: {
    [key: string]: string;
  };
}

export interface TransactionDetail extends PaymentHistoryItem {
  student_name: string;
  student_nim: string;
  student_prodi: string;
  payment_breakdown: {
    [key: string]: string;
  };
}
```

File: D:\Android\sso_wis_wismon_nodejs\be_wismon_keuangan\src\utils\auth.ts

```
import jwt from "jsonwebtoken";
import { Request, Response, NextFunction } from "express";

const JWT_SECRET = process.env.JWT_SECRET || "your-super-secret-jwt-key";

export interface JWTPayload {
  nrm: string;
  nim: string;
  namam: string;
  iat¿: number;
  exp¿: number;
}

export const generateToken = (
  payload: Omit<JWTPayload, "iat" | "exp">
): string => {
  return jwt.sign(payload, JWT_SECRET, { expiresIn: "24h" });
};

export const verifyToken = (token: string): JWTPayload => {
  return jwt.verify(token, JWT_SECRET) as JWTPayload;
};

export const authenticateToken = (
  req: Request,
  res: Response,
  next: NextFunction
): void => {
  const authHeader = req.headers["authorization"];
  const token = authHeader && authHeader.split(" ")[1]; // Bearer TOKEN

  if (!token) {
    res.status(401).json({
      success: false,
      message: "Access token required",
    });
    return;
  }

  try {
    const decoded = verifyToken(token);
    (req as any).user = decoded;
    next();
  } catch (error) {
    res.status(403).json({
      success: false,
      message: "Invalid or expired token",
    });
  }
};

export const extractTokenFromHeader = (authHeader¿: string): string | null => {
  if (!authHeader) return null;
  const parts = authHeader.split(" ");
  if (parts.length !== 2 || parts[0] !== "Bearer") return null;
  return parts[1] || null;
};
```