Porta – Chatbot Capabilities & Data Model (Build Spec v0.1)

> Goal: enumerate **everything the chatbot should answer**, plus **exact data we'll store** to ship the MVP quickly. Based on the "Everyday Investor Agent / Porta" concept.

---

1) Chatbot: What it Responds To (Intents + Response Contracts)

Each intent lists: **user examples**, **inputs** (what the backend needs), **tools used**, and **response shape** (to keep UI consistent).

1.1 Portfolio Setup & Management

- **Utterances**: "Upload my watchlist", "Track AAPL, NVDA at 60/40", "Remove TSLA", "Set weight 25% for JPM".
- **Inputs**: CSV/manual tickers, optional weights, cost basis (optional), risk tolerance (optional).
- **Tools**: `get_portfolio()`, ticker resolver.
- **Response**: `{ portfolio_summary, holdings_table, validation_warnings }`.

1.2 Portfolio-Aware News Digest (Ticker-Scoped)

- **Utterances**: "Summarize today's NVDA news", "Any negative headlines for JPM?", "News for my top 3 holdings".
- **Inputs**: tickers, lookback_days, sentiment filter (optional), relevance threshold.
- **Tools**: `get_news(ticker, lookback_days)`.
- **Response**: `{ items:[{headline, source, time, relevance, sentiment, why_it_matters}], sources[] }`.

1.3 SEC Filings: Fetch + Summarize

- **Utterances**: "Give me Apple's latest 10-Q", "Summarize AAPL 8-K from yesterday".
- **Inputs**: ticker, filing_type (10-Q/10-K/8-K/SC 13G, etc.), period.
- **Tools**: `get_filings(ticker, type)` + summarizer.
- **Response**: `{ filing_meta, key_sections[], bullets[], links[] }`.

1.4 Filing Delta (Personalized)

- **Utterances**: "What changed in AAPL's latest 10-Q vs previous?", "Show deltas that matter to my portfolio".
- **Inputs**: ticker, old_filing_id, new_filing_id, user_portfolio_weights.
- **Tools**: `diff_filings(old,new)` + portfolio weight ranking.
- **Response**: `{ deltas:[{section, change_type, excerpt, risk_flag, impact_score}], priority_ranking }`.

1.5 Personalized Alerts (Create/Read/Update/Delete)

- **Utterances**: "Alert me if NVDA drops 5% in a day", "Notify on any 8-K for my holdings", "Mute TSLA alerts for a week".
- **Inputs**: rule_type (price_move/filing/news_keyword), params (threshold, keywords), channels (in-app/email), snooze.
- **Tools**: `set_alert(ticker, rule_type, params)`.
- **Response**: `{ alert_rule, status, next_check_time }`.

1.6 Event Explanations (Why It Matters)

- **Utterances**: "Explain why this DOJ investigation matters for my holdings", "Why does rate hike hurt banks?"
- **Inputs**: detected event (news/filing/market), user portfolio context.
- **Tools**: `explain_event(event, portfolio)`.
- **Response**: `{ explanation, per_ticker_impact:[{ticker, direction, rationale}], severity_badge }`.

1.7 Stress Test Simulator (Unique Angle)

- **Utterances**: "What if oil +20%?", "Fed +100 bps", "Recession shock", "USD strengthens 5%".
- **Inputs**: scenario template + parameters, portfolio holdings.
- **Tools**: `run_stress_test(event, portfolio)`.
- **Response**: `{ scenario, assumptions, per_ticker_effect:[{ticker, \u2191/\u2193/\u2194, score}], heatmap, summary }`.

1.8 Earnings Call Quote Extractor (Stretch)

- **Utterances**: "Key quotes from NVDA earnings", "Mgmt tone for JPM Q2 call".
- **Inputs**: ticker, call date, transcript source.
- **Tools**: transcript fetcher, quote+sentiment extractor.
- **Response**: `{ quotes:[{speaker, quote, topic, sentiment}], takeaways[] }`.

1.9 Thematic Mapper (Stretch)

- **Utterances**: "If US passes new chip export rules, who in my portfolio is impacted?", "AI server spend slowdown impact".
- **Inputs**: theme keywords, mapping ontology, portfolio list.
- **Tools**: theme-to-ticker mapping, rule engine.
- **Response**: `{ theme, impacted_holdings:[{ticker, direction, rationale}], confidence }`.

1.10 Weekly Digest (Stretch)

- **Utterances**: "Give me my weekly portfolio story", "What changed this week?"
- **Inputs**: lookback=7d, portfolio.
- **Tools**: scheduled generator.
- **Response**: `{ highlights, top_risks, top_opportunities, upcoming_events }`.

1.11 Definitions / Education (Guardrail-Friendly)

- **Utterances**: "What is an 8-K?", "Explain guidance cut", "What's FCF margin?".
- **Inputs**: term.
- **Tools**: glossary.
- **Response**: `{ definition, example, links }`.

1.12 Onboarding & Help

- **Utterances**: "How do alerts work?", "Show supported file formats for upload".
- **Response**: `{ steps[], example_csv, faq[] }`.

1.13 Scope Boundaries / Refusals

- **Utterances**: "Tell me whether to buy/sell", "Tax advice?".
- **Response**: Decline (educational only) + neutral resources.

---

2) Conversation Patterns (UX Contracts)

- **Answer cards**: headline, timestamp, source chip(s), 'Why it matters' paragraph, badge (🔴/🟡/⚪), actions (\n `Save`, `Create alert`, `Open source`).
- **Follow-ups** suggested automatically: "Compare to last quarter?", "Set alert?", "Stress test this news?".
- **Threading**: Each user prompt creates/continues a `chat_session_id`; messages carry `context_ref` (filing_id, news_id, scenario_id).

---

3) Data We Will Store (Database Model)

Assume Postgres. Types use `snake_case`. Timestamps are UTC with `created_at/updated_at`.

3.1 Core Entities

1) **users** - `id (uuid PK)`, `email (text uniq)`, `display_name (text)`, `auth_provider (text)`, `consent_version (text)`, `time_zone (text)` - Index: `(email)`

2) **portfolios** - `id (uuid PK)`, `user_id (uuid FK users)`, `name (text)`, `base_currency (text default 'USD')` - Unique: `(user_id, name)`

3) **holdings** - `id (uuid PK)`, `portfolio_id (uuid FK portfolios)`, `ticker (text)`, `weight_pct (numeric)`, `cost_basis (numeric null)`, `notes (text)` - Index: `(portfolio_id, ticker)`

4) **instruments** (master ref) - `ticker (text PK)`, `name (text)`, `exchange (text)`, `sector (text)`, `industry (text)`, `country (text)`, `isin (text)`

5) **price_snapshots** - `id (bigserial PK)`, `ticker (text FK instruments)`, `as_of (timestamptz)`, `price (numeric)`, `pct_change_1d (numeric)` - Index: `(ticker, as_of desc)`

6) **news_items** - `id (uuid PK)`, `ticker (text FK instruments)`, `headline (text)`, `source (text)`, `published_at (timestamptz)`, `url (text)`, `summary (text)`, `sentiment (int -1/0/1)`, `relevance (numeric)` - Index: `(ticker, published_at desc)`

7) **filings** - `id (uuid PK)`, `ticker (text FK instruments)`, `filing_type (text)`, `period_end (date)`, `filed_at (timestamptz)`, `accession_no (text uniq)`, `url (text)`, `raw_text (text)` - Index: `(ticker, filed_at desc)`

8) **filing_sections** (optional normalized) - `id (uuid PK)`, `filing_id (uuid FK filings)`, `section (text)`, `content (text)`

9) **filing_deltas** - `id (uuid PK)`, `old_filing_id (uuid FK filings)`, `new_filing_id (uuid FK filings)`, `section (text)`, `change_type (enum add/remove/modify)`, `excerpt_old (text)`, `excerpt_new (text)`, `impact_score (numeric)`, `risk_flag (bool)` - Index: `(new_filing_id, impact_score desc)`

10) **alert_rules** - `id (uuid PK)`, `user_id (uuid FK users)`, `ticker (text null)`, `rule_type (enum price_move/filing/news_keyword)`, `params (jsonb)`, `is_active (bool)`, `channel (enum in_app/email)`, `snooze_until (timestamptz null)` - Index: `(user_id, is_active)`

11) **alert_events** - `id (uuid PK)`, `alert_rule_id (uuid FK alert_rules)`, `triggered_at (timestamptz)`, `payload (jsonb)`, `delivered (bool)` - Index: `(alert_rule_id, triggered_at desc)`

12) **stress_test_scenarios** - `id (uuid PK)`, `user_id (uuid FK users)`, `name (text)`, `params (jsonb)`, `created_from_template (text)`

13) **stress_test_results** - `id (uuid PK)`, `scenario_id (uuid FK stress_test_scenarios)`, `portfolio_id (uuid FK portfolios)`, `run_at (timestamptz)`, `per_ticker (jsonb)`, `summary (text)`

14) **chat_sessions** - `id (uuid PK)`, `user_id (uuid FK users)`, `title (text)`, `started_at (timestamptz)`

15) **chat_messages** - `id (uuid PK)`, `chat_session_id (uuid FK chat_sessions)`, `sender (enum user/assistant/system)`, `text (text)`, `context_ref (jsonb null)`, `created_at (timestamptz)`

16) **user_preferences** - `user_id (uuid PK)`, `news_keywords_blocklist (text[])`, `min_relevance (numeric)`, `default_lookback_days (int)`, `education_mode (bool)`

17) **digests** (stretch) - `id (uuid PK)`, `user_id (uuid FK users)`, `period (enum weekly/monthly)`, `generated_at (timestamptz)`, `content (jsonb)`

18) **transcript_snippets** (stretch) - `id (uuid PK)`, `ticker (text FK instruments)`, `call_time (timestamptz)`, `speaker (text)`, `quote (text)`, `topic (text)`, `sentiment (int)`

19) **themes** (stretch) - `id (uuid PK)`, `name (text uniq)`, `keywords (text[])`, `rules (jsonb)`

20) **theme_impacts** (stretch) - `id (uuid PK)`, `theme_id (uuid FK themes)`, `ticker (text FK instruments)`, `direction (int -1/0/1)`, `rationale (text)`, `confidence (numeric)`

21) **source_links** (for strict citations) - `id (uuid PK)`, `entity_type (text)`, `entity_id (uuid)`, `label (text)`, `url (text)`

22) **audit_logs** - `id (bigserial PK)`, `user_id (uuid)`, `action (text)`, `meta (jsonb)`, `created_at (timestamptz)`

23) **rate_limits** - `user_id (uuid PK)`, `window_start (timestamptz)`, `tokens_used (int)`

24) **vendor_integrations** - `id (uuid PK)`, `name (text)`, `auth (jsonb)`, `scopes (text[])`, `status (text)`

3.2 Indexing & Search

- **Full-text** on `news_items.headline`, `filings.raw_text`, `transcript_snippets.quote`.
- **GIN** index for `jsonb` fields used in filters (`alert_rules.params`, `stress_test_results.per_ticker`).
- **Materialized views**: `mv_portfolio_top_holdings`, `mv_recent_alerts`, `mv_latest_filings_by_ticker`.

3.3 Data Retention & Privacy

- Store **only**: portfolio tickers/weights, alert prefs, chat logs needed for product quality.
- Do **not** store: bank creds, SSN, addresses, or brokerage API tokens (unless explicit feature added later).
- **Retention**: chats 90 days (configurable), logs 180 days, sources permanent (public data).
- **User controls**: export/delete portfolio & chats; per-alert mute/snooze; opt-out of sentiment analysis.

---

4) Event Triggers & Background Jobs

- **Price watcher** (1–5 min): compute 1D % moves; fire `price_move` alerts.
- **Filing watcher** (5–10 min): poll SEC; insert `filings`; run `filing_deltas` vs previous; notify.
- **News fetcher** (5–10 min): ingest news; rank by portfolio weight * relevance.
- **Digest builder** (weekly): compile `digests`.
- **Theme refresher** (daily): update theme → ticker impacts.

- **Cleanup** (daily): retention pruning, archive old snapshots.

---

5) Stress Test Method (MVP Heuristics)

- **Scenarios**: `oil +20%`, `rates +100 bps`, `USD +5%`, `market -10%`, `AI capex slowdown`.
- **Mapping**: sector/industry rules (e.g., airlines \u2193 on oil ↑; energy producers \u2191).
- **Computation**: per holding `impact_score = base_sensitivity * user_weight * scenario_multiplier`.
- **Output**: badge (🔴negative / 🟡neutral / 🟢positive), heatmap list.

---

6) API Surface (Internal)

- `POST /portfolio` (create/update from CSV)
- `GET /portfolio/:id`
- `POST /alerts` (create rule)
- `PATCH /alerts/:id` (snooze/mute)
- `GET /news?ticker=...&lookback=...`
- `GET /filings?ticker=...&type=...`
- `GET /filings/:id/delta?compare_to=prev`
- `POST /stress-tests` (create+run) → returns result id
- `GET /stress-tests/:id`
- `POST /chat` (session_id, text) → structured response contract

---

7) UI Payload Contracts (for Frontend Tiles)

- **News Card**: `{ticker, headline, source, published_at, sentiment, relevance, why_it_matters, link}`
- **Filing Delta Card**: `{ticker, filing_type, filed_at, top_deltas[], severity}`
- **Alert Card**: `{priority, reason, timestamp, actions[]}`
- **Stress Test Card**: `{scenario, per_ticker_effect[], summary}`
- **Portfolio Tile**: `{weights[], top_holdings[], risk_notes[]}`

---

8) Guardrails & Compliance

- **Educational only** banner on every analytical response.
- **Citations required** for news/filings; store in `source_links` and attach to responses.
- **Scope filter**: refuse non-finance personal chat; provide help/definitions instead.

---

9) Edge Cases & Rules

- Unknown ticker → resolver + clarifying prompt.
- Duplicate uploads → merge strategy by `ticker`; last write wins for weights.
- Conflicting alerts → dedupe window (e.g., 30 min) per ticker+rule.
- Empty news day → fallback to weekly context or earnings calendar.
- Massive filings (10-K) → sectioned summaries + lazy load.

---

10) Minimal Seed List (to build fast)

- **Sectors sensitivities** JSON for stress tests.
- **Glossary** for 30 finance terms.
- **Five scenario templates** (rates, oil, USD, recession, AI-capex).
- **Top 2 news sources** + SEC EDGAR JSON feeds.

Ready-to-Build Checklist

- [ ] Tables & indexes created
- [ ] Price/news/filing ingestors running
- [ ] Alerts engine (rules → triggers → deliveries)
- [ ] Delta diff pipeline
- [ ] Chat orchestration (intent router → tool calls → response cards)
- [ ] UI tiles wired to payload contracts
- [ ] Guardrails & citations

1A) Chatbot-First Build Spec (v1.0) — Version 1

This section isolates **only the chatbot**: intents → slots → tool calls → response JSON, plus what we persist for the bot to work reliably.

A. Intent Taxonomy (exhaustive for MVP)

1. `portfolio.create_or_replace`
2. `portfolio.view`
3. `portfolio.edit.add_tickers`
4. `portfolio.edit.remove_tickers`
5. `portfolio.edit.update_weights`
6. `news.digest.by_ticker`
7. `news.digest.for_portfolio`
8. `news.search.keyword`
9. `filing.fetch`
10. `filing.summarize`

11. `filing.delta`
12. `alert.create`
13. `alert.list`
14. `alert.update` (mute/snooze/activate/deactivate)
15. `alert.delete`
16. `stress_test.run`
17. `education.define_term`
18. `help.onboarding`
19. `safety.refusal` (buy/sell, tax/estate advice, etc.)

B. Entity/Slot Schema (NLU)

- `ticker` (AAPL) [multi]
- `company_name` (Apple Inc.)
- `weight_pct` (numeric 0–100)
- `lookback_days` (int, default from prefs)
- `date_range` (ISO start/end)
- `filing_type` (10-K, 10-Q, 8-K, 13F, etc.)
- `filing_period` (e.g., FY2024, Q2 2025)
- `keywords` (array of strings)
- `threshold_pct` (e.g., 5)
- `rule_type` (price_move | filing | news_keyword)
- `channel` (in_app | email)
- `scenario_type` (rates_up, oil_spike, usd_up, recession, ai_capex_slow)
- `scenario_params` (json)
- `sentiment_filter` (-1 | 0 | 1 | any)
- `relevance_min` (0–1)

Parser order of operations: **Ticker resolver → Filing type normalizer → Date/percentage extractor → Fallback keyword capture**.

C. Routing Table (Intent → Tool Calls)

- `portfolio.*` → `get_portfolio`, holdings writer
- `news.*` → `get_news`
- `filing.*` → `get_filings`, `diff_filings`
- `alert.*` → `set_alert`, alert store
- `stress_test.run` → `run_stress_test`
- `education.*` → glossary store
- `help.*` → static help content

D. Response JSON Contracts (authoritative)

D1. `portfolio.create_or_replace`

Request slots: `tickers[]`, `weights[]?`, `base_currency?`, `cost_basis?` Response:

```
{
  "type": "portfolio.summary",
  "portfolio_id": "uuid",
  "holdings": [{"ticker":"AAPL","weight_pct":40}],
  "validation_warnings": ["Weights sum to 98%; normalized to 100%"],
  "next_actions": ["news.digest.for_portfolio", "alert.create"]
}
```


D2. `news.digest.by_ticker`

Slots: `ticker`, `lookback_days?`, `sentiment_filter?`, `relevance_min?`


```
{
  "type": "news.digest",
  "ticker": "NVDA",
  "items": [
    {"headline":"…","source":"…","published_at":"…","relevance":
0.82,"sentiment":1,
     "why_it_matters":"…","url":"…"}
  ],
  "sources": ["Brave", "Reuters"],
  "generated_at": "2025-08-22T13:00:00Z"
}
```


D3. `filing.fetch` **/** `filing.summarize`

Slots: `ticker`, `filing_type`, `filing_period?`


```
{
  "type": "filing.summary",
  "ticker": "AAPL",
  "filing": {"type":"10-Q","filed_at":"…","url":"…"},
  "key_sections": [{"section":"MD&A","takeaways":["…","…"]}],
  "bullets": ["Revenue +8% y/y","Gross margin +120 bps"],
  "citations": ["…"]
}
```


D4. `filing.delta`

Slots: `ticker`, `old_filing_id?`, `new_filing_id?` (auto-resolve to latest two)

```
{
  "type": "filing.delta",
  "ticker":"AAPL",
  "top_deltas":[
    {"section":"Risk Factors","change_type":"modify",
     "excerpt_old":"…","excerpt_new":"…",
     "impact_score":0.76,"risk_flag":true}
  ],
  "priority_ranking": [{"section":"Liquidity","score":0.64}],
  "citations": ["…"]
}
```

D5. `alert.create`

Slots: `ticker?`, `rule_type`, `threshold_pct?`, `keywords?`, `channel?`

```
{
  "type":"alert.created",
  "alert_rule": {"id":"uuid","rule_type":"price_move","params":{"threshold_pct":
5,"direction":"down"}},
  "status":"active",
  "next_check_time":"2025-08-22T13:05:00Z"
}
```

D6. `stress_test.run`

Slots: `scenario_type`, `scenario_params?`, `portfolio_id`

```
{
  "type":"stress_test.result",
  "scenario":"rates_up_100bps",
  "assumptions":["Banks ↑ NIM short-term","Long-duration tech ↓"],
  "per_ticker_effect":[{"ticker":"JPM","direction":"up","score":0.4}],
  "summary":"Net mildly positive given 25% banks weight"
}
```

D7. `education.define_term`

```
{"type":"education.definition","term":"8-K","definition":"…","example":"…","links":
["…"]}
```

E. Conversation State & Memory

- **Per session**: `chat_session_id`, `last_intent`, `unfilled_slots`, `context_ref` (filing_id/news_id/scenario_id), `suggested_next_actions[]`.
- **Cross-session (user memory)**: portfolio ids, alert prefs, glossary progress (optional), default lookback, news keyword blocklist.
- **Thread safety**: deterministic tool ordering; each tool call logged with `tool_invocation_id`.

F. Persistence (Chatbot-specific tables/additions)

New/expanded tables (on top of §3): - **intent_logs**: `id, chat_message_id, intent, confidence, slots_json, routed_tools[], duration_ms` - **tool_invocations**: `id, intent_log_id, tool_name, request_json, response_meta, success, latency_ms` - **citations**: `id, message_id, source_label, url, hash` - **saved_insights**: `id, user_id, type(news|filing|stress), payload_json, created_at` - **feedback**: `id, message_id, user_id, rating(1–5), comment`

G. Guardrails & Refusals (Bot level)

- **Refusal templates**: neutral tone, educational redirect, show glossary/help links.
- **No recommendations**: never output buy/sell/hold; transform to risk/considerations.
- **Citations mandatory** for news/filings outputs.

H. Latency Budget (MVP)

- Intent detection + slot fill: $\leq$ 200 ms
- Single tool call (news/filing): P50 $\leq$ 1.5 s; P95 $\leq$ 4 s
- Multi-tool (delta diff): P50 $\leq$ 3 s; P95 $\leq$ 7 s

I. Error Handling & Fallbacks

- Resolver fails → ask for ticker confirmation; continue once resolved.
- Empty news → fallback to 7-day digest or earnings calendar mention.
- Filing missing → return latest available + note the gap.
- Tool timeout → degrade gracefully with cached summary if present.

J. Test Plan (NLU & E2E)

- 10 utterances per intent; success if $\geq$ 90% correct routing.
- Golden outputs with snapshot tests for response JSON.
- Red-team prompts for safety/refusal.

K. Defaults (Config)

- `default_lookback_days=3`
- `relevance_min=0.6`
- `sentiment_filter=any`
- `max_items_per_digest=6`

(End chatbot-first spec.)

Version History

- **v1.0** (current): Chatbot-First Build Spec promoted from v0.2 to Version 1 as requested.