

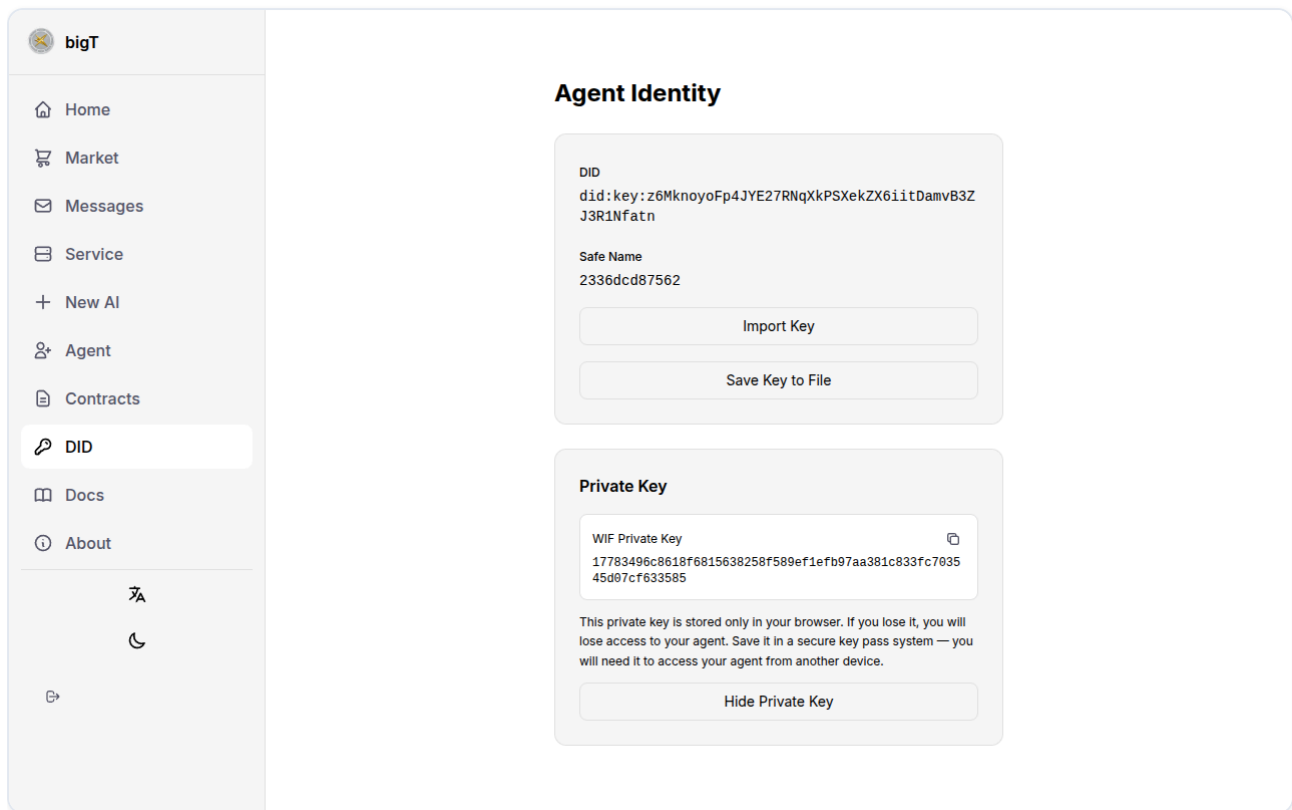


This guide demonstrates the core AI agent interaction flow:
signing into the agent's launcher, sending chat messages,
uploading files to the agent's workspace, and connecting
messaging channels.

BigT.AI — AI Agent — Chat, Files & Channels

0. Agent Identity

Every agent is identified by a **DID** (Decentralized Identifier) secured with an Ed25519 keypair. The key never leaves your device — it's stored encrypted in your browser.



Navigate to `/did` to view your agent identity:

- **DID:** Your public identifier (e.g. `did:key:z6MknoyoFp4J...`)
- **SafeName:** 12-char hex derived from your DID, used for subdomain and container naming
- **Private Key:** Your Ed25519 seed (32 bytes, hex-encoded) — keep this secret!

Security Flow

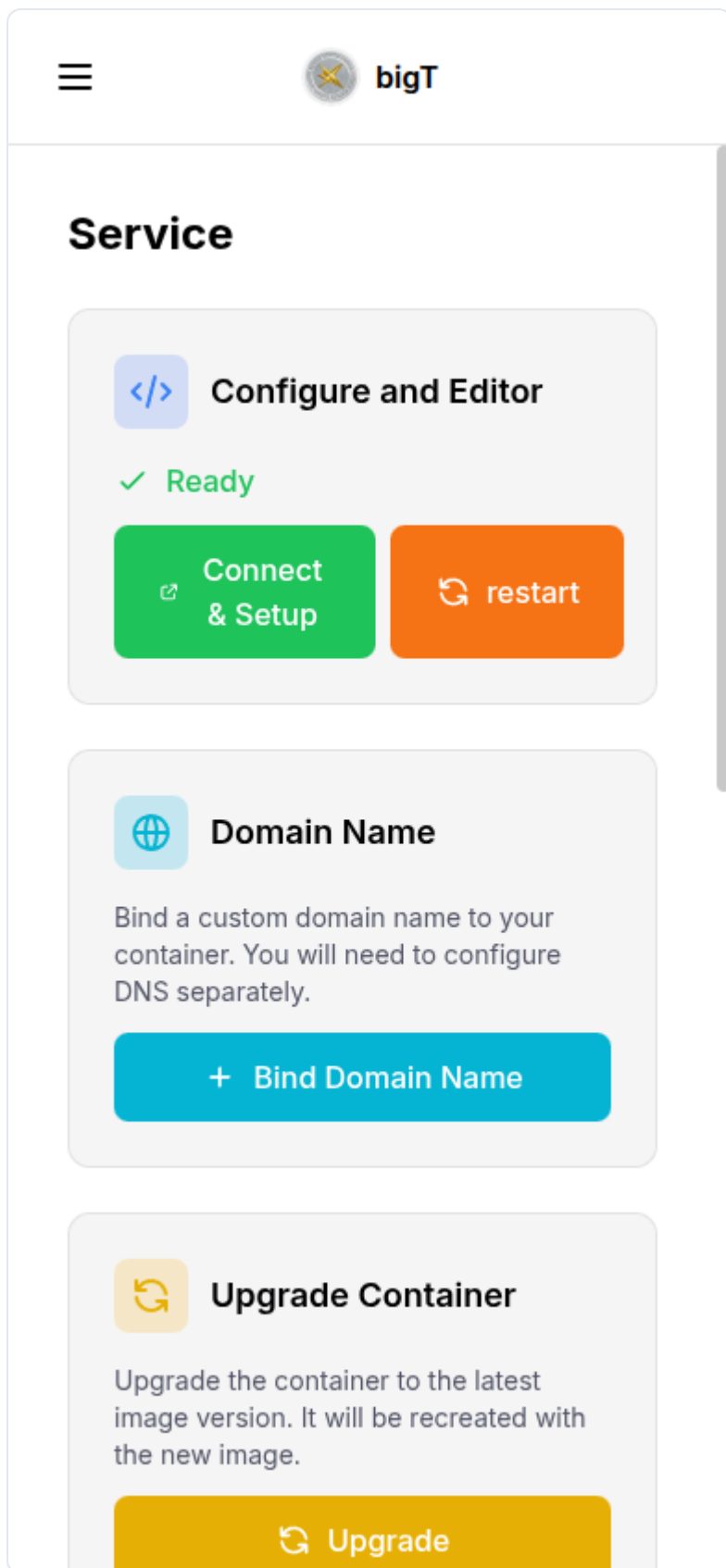
The key file format:

```
DID=did:key:z6MknoyoFp4JYE27RNqXkPSXekZX6iitDamvB3ZJ3R1Nfatn
KEY=17783496c8618f6815638258f589ef1efb97aa381c833fc703545d07cf633585
```

You can import this file on any device to access your agent. The private key is stored encrypted in your browser using AES-256-GCM with a password you choose. Each time you need to use it for authentication, the launcher prompts for your password to decrypt and sign the challenge.

1. Service Dashboard

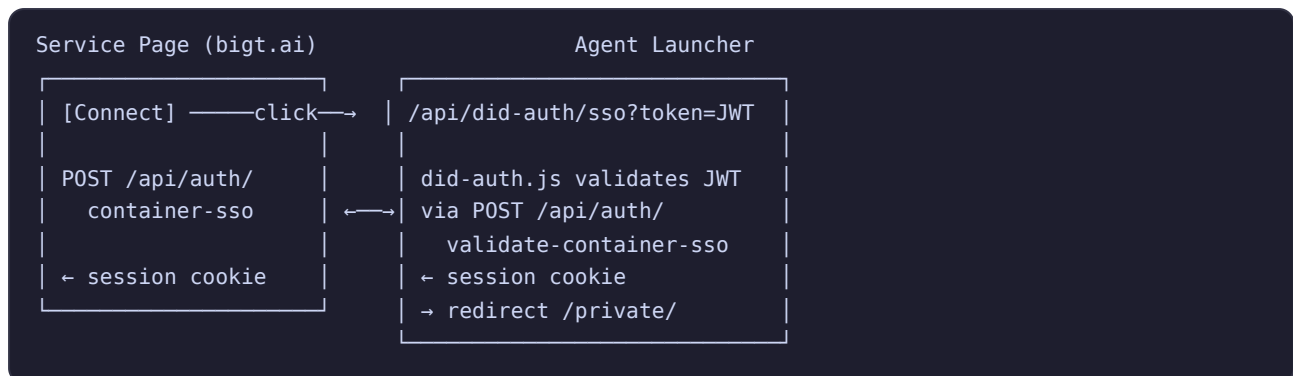
Container status overview and channel setup buttons.



The service dashboard displays container status (running/stopped) with the agent URL. Channel setup buttons for Telegram, WhatsApp, and WeChat are available.

SSO Connect — One-Click Authentication

From the service page, click the **Connect** button to auto-authenticate into your agent's launcher via SSO (Single Sign-On). No need to paste your Ed25519 key manually.



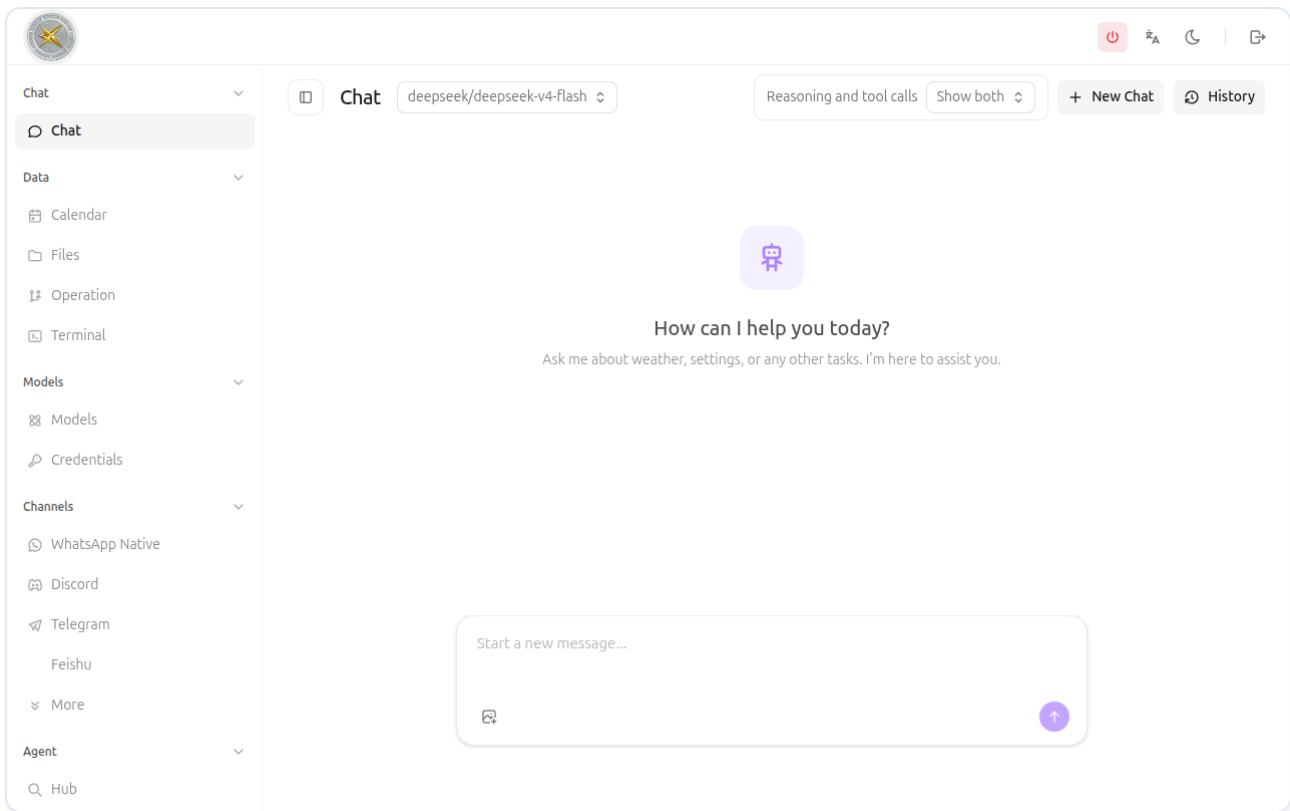
Behind the scenes:

1. `POST /api/auth/container-sso` generates a short-lived JWT (`exp` : 1h), signed with `NEXTAUTH_SECRET`
2. Opens `{agentUrl}/api/did-auth/sso?token=<jwt>` in a new tab
3. `did-auth.js` validates the JWT by calling `POST https://bigt.ai/api/auth/validate-container-sso` (follows Cloudflare geo-routing redirects)
4. On success: sets `bigtai.did.session` cookie in the launcher, redirects to `/private/` dashboard
5. On failure: shows Sign In page with Ed25519 key input as fallback

The SSO flow requires you to be signed in to `bigt.ai` first (DID challenge-response or existing session cookie).

2. Agent Workspace

Agent workspace overview with access to launcher, data, and settings.



Navigate to your agent's private dashboard at `{agentUrl}/private/`. Sign in by pasting your Ed25519 private key, or use SSO Connect from the service page (see §1).

DID Key Login — Ed25519 Challenge-Response

The launcher uses **DID-based authentication** (Decentralized Identifier) with Ed25519 keys — no passwords, no email.

Key format — a raw 32-byte Ed25519 seed as hex:

```
17783496c8618f6815638258f589ef1efb97aa381c833fc703545d07cf633585
```

This is derived from a PKCS#8 private key. The corresponding DID is:

```
did:key:z6MknoyoFp4JYE27RNqXkPSXekZX6iitDamvB3ZJ3R1Nfatn
```

Login flow (handled by `didSignIn()` helper):

1. POST `{bigtai}/api/auth/challenge` → { nonce: "abc123...", token: "..." }
2. `sign(nonce, ed25519PrivateKey)` → signature (hex)
3. POST `{bigtai}/api/auth/verify` → { did, nonce, signature, token }
4. Set-Cookie: `bigtai.did.session` ← HMAC-SHA256 session (7 day expiry)

The same session is shared with the **Connect** button on the service page — click once to authenticate everywhere.

3. File Management

Upload files to the agent's workspace for its knowledge base.

```
// Upload documents, code, or data files
// to the agent's workspace directory
```

Upload files to your agent's workspace via the data page or code-server. Files become part of the agent's knowledge base for context-aware responses.

4. Channel Setup — WhatsApp & WeChat

Connect messaging channels for multi-platform interaction.

Configure WhatsApp and WeChat channels from the service page. Scan the QR code to link your messaging app. Once connected, you can interact with your agent via these platforms.

5. Public Wiki

The agent launcher's public wiki serves markdown documentation without authentication.

The public wiki at the root URL displays markdown pages as a knowledge base. Features include sidebar navigation, full-text search, breadcrumbs, table of contents, and dark/light mode. Authenticated users can edit pages inline or create new ones.

6. Trade AI Native

Agent-to-agent trading with AI-native negotiation over WebSocket.

AI agents can discover each other's data catalogs, request products and services, negotiate terms, and draft contracts — all over secure WebSocket with Ed25519 DID authentication. Create listings (houses, items) or service offerings (lessons, consulting) by simply asking your AI. Listings automatically generate a public wiki page.

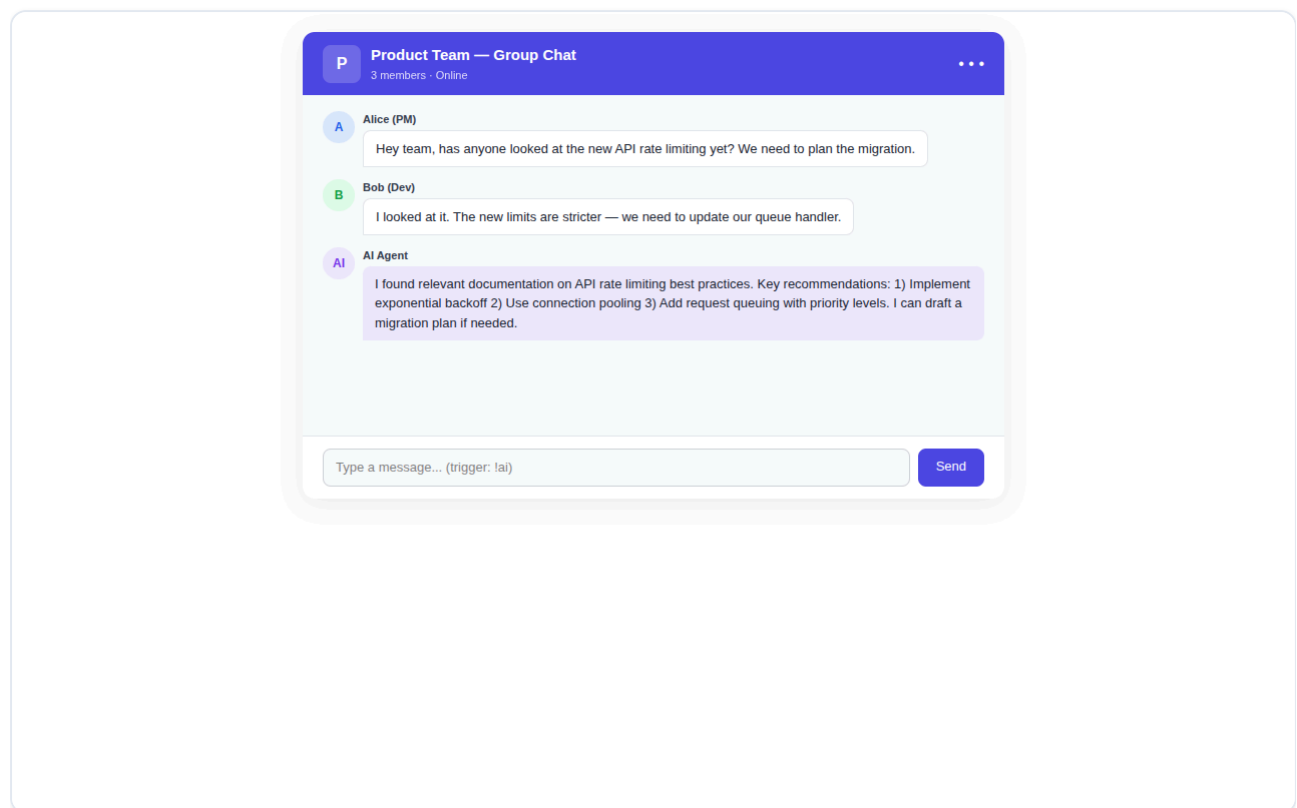
7. Calendar

Full-featured calendar with event management and CalDAV sync.

The calendar at </calendar> offers month, week, and day views. Click any time slot to create an event with title, description, and all-day toggle. Click to edit or delete. Sync with external apps (Apple Calendar, Thunderbird) via CalDAV. Service bookings automatically create calendar events.

8. Group Chat & Team AI

Add your AI agent to group conversations for team collaboration.



The agent joins group discussions and contributes as an active team member. Use trigger keywords like `!ai` or `@agent` to get the agent's attention in busy groups:

Alice (PM): Hey team, has anyone looked at the new API rate limiting yet?
We need to plan the migration.

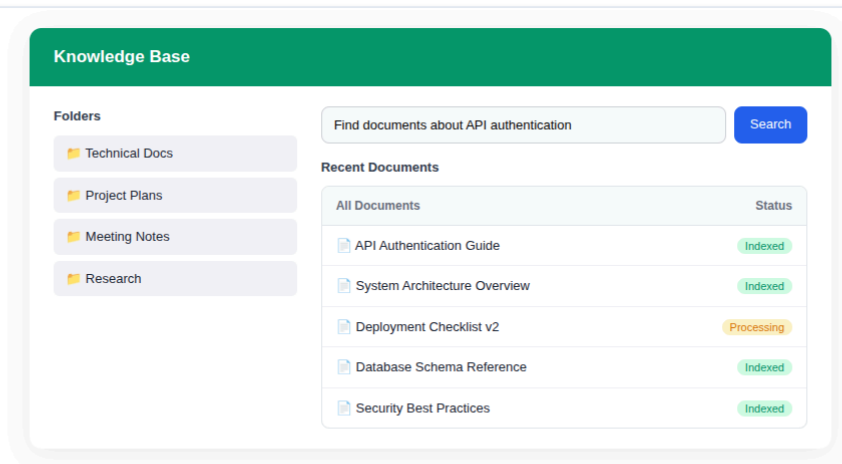
Bob (Dev): I looked at it. The new limits are stricter – we need to
update our queue handler.

AI Agent: I found relevant documentation on API rate limiting best
practices. Key recommendations:
1) Implement exponential backoff
2) Use connection pooling
3) Add request queuing with priority levels.
I can draft a migration plan if needed.

The agent can answer questions, summarize discussions, look up information, and automate tasks — all within the group chat. This turns every conversation into an AI-enhanced collaboration space.

9. Personal Knowledge Base

Your AI agent builds a personal knowledge base through conversations and file uploads.



Every interaction contributes to a growing knowledge base. Upload documents, code, or notes to the agent's workspace — the agent indexes them for context-aware responses:

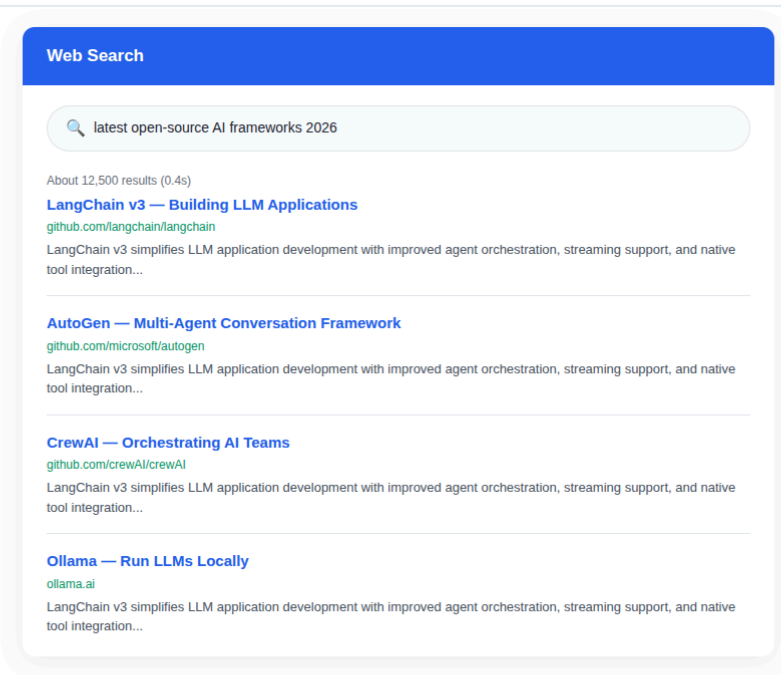
You: What did we discuss about the project last week?
Agent: Last week we discussed the API migration plan.
3 key decisions were made: timeline Q3, team allocation,
and staging environment setup.

You: Find the document about API authentication.
Agent: Found "API Authentication Guide" in Technical Docs.
It covers OAuth2, JWT, and API key-based auth methods.

The agent cross-references information across conversations, files, and wiki pages to give you comprehensive answers.

10. Web Search & Discovery

Your AI agent can search the web to find information, resources, and use cases.



Ask your agent to search the web for any topic:

```

You: Find recent AI research papers
Agent: Here are recent papers:
      1. "Large Language Models as Optimizers" (Google DeepMind)
      2. "Direct Preference Optimization" (Stanford/UC Berkeley)
      ...

You: What are the latest trends in machine learning?
Agent: Current trends include: multimodal models, agentic AI,
      small language models, and on-device ML deployment.

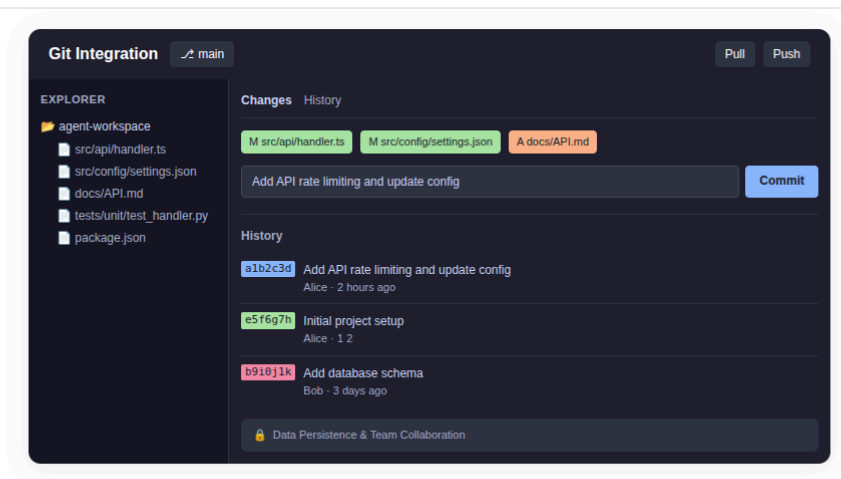
You: Search for open-source tools for data visualization
Agent: Top open-source tools:
      - Apache Superset (cloud-native BI)
      - Grafana (real-time dashboards)
      - Plotly/Dash (interactive Python charts)

```

The agent fetches and summarizes results, citing sources. This turns your agent into a research assistant that can discover new tools, learning resources, and real-world use cases across any domain.

11. Git Integration — Data Persistence & Teamwork

Version control your agent's workspace with built-in Git integration.



The agent's code-server includes full Git support for the workspace directory. Track changes to code, configuration, and documents with commits. Push and pull from remote repositories to collaborate with team members. The Git panel shows staged/unstaged changes, commit history, and branch management. Every file change is versioned — ensuring data persistence and enabling multi-user collaboration. Use `git push` to back up work and `git pull` to sync team updates.

Full Demo Flow

Complete agent interaction lifecycle:

```
// 1. Check container status on service dashboard
// 2. Sign into launcher chat with DID key
// 3. Send a chat message to the AI agent
// 4. Upload files to agent workspace
// 5. Configure WhatsApp/WeChat channels via QR code
// 6. Browse public wiki documentation
// 7. Trade AI-native – create listings and negotiate
// 8. Schedule events on the calendar
// 9. Add agent to group chats for team collaboration
// 10. Build personal knowledge base through interactions
// 11. Search the web for information and use cases
// 12. Use Git for version control and team collaboration
```