

langchain-ai / agent-auth-payments

🔍

📁

<> Code

🕒 Issues

🔗 Pull requests

🎬 Actions

📅 Projects

🛡 Security

📊 Insights

👁

🔗

☆

Use this template ▾

an open source fullstack chat agent with authentication, request credits and payments built in

📄 MIT license

☆ 89 stars

🔗 24 forks

👁 1 watching

🔗 Branches

📈 Activity

📋 Custom properties

🏷 Tags

📄 Public template repository

🔗

🔗 1 Branch

🏷 0 Tags

🔗

🏷

🔍 Go to file

t

Go to file

+

Add file ▾

<> Code ▾

⋮

	starmorph rename agentic-saas-template to agent-auth-payments ✓	1e13834 · 2 weeks ago 🕒
📁 .github/workflows	update CI dev server check	2 months ago
📁 .vscode	init fullstack chat monorepo	2 months ago
📁 apps	rename agentic-saas-template to ag...	2 weeks ago
📁 docs	rename agentic-saas-template to ag...	2 weeks ago
📄 .codespellignore	add back codespellignore for CI	2 months ago
📄 .gitignore	drop turbo, --> pnpm	2 months ago
📄 .npmrc	init monorepo fullstack chat app	2 months ago
📄 LICENSE	init fullstack chat monorepo	2 months ago
📄 README.md	rename agentic-saas-template to ag...	2 weeks ago
📄 package.json	rename agentic-saas-template to ag...	2 weeks ago
📄 pnpm-lock.yaml	drop turbo, --> pnpm	2 months ago
📄 pnpm-workspace.yaml	init monorepo fullstack chat app	2 months ago
📄 supabase-schema.sql	docs improvements + supabase SQL...	2 months ago
📄 tsconfig.json	init fullstack chat monorepo	2 months ago

Agent with Auth and Payments Repo

A monorepo containing a Agent with Auth and Payments application with LangGraph agents and Next.js UI.

Architecture

This monorepo contains two main applications:

- `apps/web` - Next.js chat UI application with LangGraph integration
- `apps/agents` - LangGraph.js ReAct agents backend

Quick Start

Terminal Tab 1:

```
# Clone the repo
git clone https://github.com/langchain-ai/agent-auth-payments.git

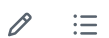
# **Environment Files**: Copy the `.env.example` files to `.env` and fill in credentials
cp apps/web/.env.example apps/web/.env
cp apps/agents/.env.example apps/agents/.env

# Install dependencies for all apps
pnpm install

# Start development servers for both apps
pnpm dev
```

Database Setup

 [README](#)  [MIT license](#)



What gets set up:

- ✓ Users table with Stripe integration
- ✓ Row Level Security (RLS) policies
- ✓ Automatic user profile creation
- ✓ Performance indexes and triggers

Terminal Tab 2: Stripe Webhook (for purchases + credits)

```
stripe listen --events
customer.subscription.created,customer.subscription.updated,customer.subscription.deleted --
forward-to localhost:3000/api/webhooks/stripe

## add stripe webhook key to apps/web/.env
STRIPE_WEBHOOK_SECRET=""
```

You're ready to use the app!

Use the App

1. Open localhost:3000
2. Sign up -> confirm email
3. login
4. pricing page --> purchase credits
 - a. should see stripe events in Terminal Tab 3
5. should see success page, new credits added
6. back to home, chat with app, credits get deducted



Package Management

This monorepo uses **pnpm workspaces** for efficient dependency management and task orchestration.

Available Scripts

Root Level Commands

```
# Development
pnpm dev          # Start all apps in development mode (parallel)
pnpm build        # Build all apps
pnpm lint         # Lint all apps
pnpm lint:fix     # Fix linting issues in all apps
pnpm format       # Format code in all apps
pnpm format:check # Check code formatting in all apps
pnpm test         # Run tests in all apps
pnpm test:int     # Run integration tests in all apps
pnpm clean        # Clean all build artifacts and node_modules

# Individual App Commands
pnpm web:dev      # Start only the web app
pnpm web:build    # Build only the web app
pnpm agents:dev   # Start only the agents app
pnpm agents:build # Build only the agents app
pnpm agents:test  # Test only the agents app
pnpm agents:test:int # Integration tests for agents app
```



Project Structure

```
├─ apps/
│  ├─ web/                # Next.js chat UI
│  │  ├─ src/
│  │  ├─ package.json
│  │  └─ ...
│  └─ agents/             # LangGraph agents
│     ├─ src/
│     ├─ package.json
│     └─ ...
└─ package.json           # Root package.json with workspaces
```



```
├─ pnpm-workspace.yaml    # pnpm workspace configuration
├─ .npmrc                 # pnpm configuration
└─ README.md
```

Technology Stack

Web App (apps/web)

- **Framework:** Next.js 15
- **UI:** Radix UI + Tailwind CSS + shadcn/ui
- **Auth:** Supabase,
- **Payments:** Stripe SDK
- **State:** Nuqs, Zustand
- **Package Manager:** pnpm

Agents App (apps/agents)

- **Runtime:** Node.js + TypeScript
- **Framework:** LangGraph.js
- **AI:** LangChain + Anthropic
- **Auth:** Langgraph Middleware
- **Testing:** Jest
- **Package Manager:** pnpm

Development Workflow

Adding Dependencies

```
# Add to specific app
pnpm --filter web add <package>
pnpm --filter agents add <package>

# Add dev dependency to specific app
pnpm --filter web add -D <package>

# Add to root (for tooling)
pnpm add -D <package> -w
```



Running Tests

```
# All tests
pnpm test

# Only agents tests
pnpm agents:test
```



```
# Integration tests
pnpm test:int
```

Building for Production

```
# Build all apps
pnpm build

# Build specific app
pnpm web:build
pnpm agents:build
```



Deployment

Each app can be deployed independently:

- **Web App:** Deploy to Vercel, Netlify, or any Node.js hosting
- **Agents:** Deploy to any Node.js hosting or containerize with Docker

Contributing

1. Install dependencies: `pnpm install`
2. Start development: `pnpm dev`
3. Make your changes
4. Run tests: `pnpm test`
5. Format code: `pnpm format`
6. Submit a pull request

Releases

No releases published

Packages

No packages published

Languages

