

# Qwen Proxy VSCode Extension

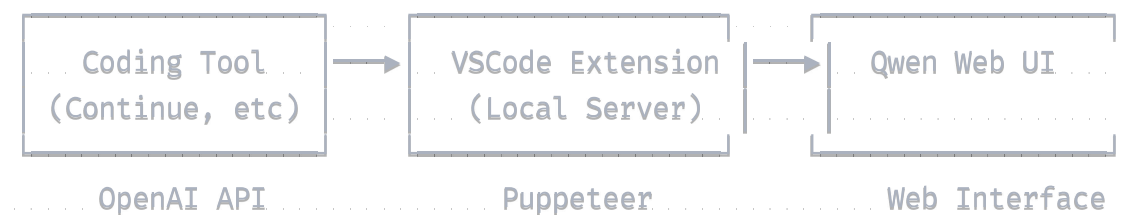
## Project Overview

A VSCode extension that creates an OpenAI-compatible API proxy for routing requests through the Qwen web interface. This allows existing coding assistants to use Qwen models without modification.

## Educational Goals

- Learn VSCode extension development
- Understand web scraping with Puppeteer
- Build API proxy servers
- Handle authentication and session management
- Practice error handling and rate limiting

## Architecture



## Current Status

- ✅ Initial project structure planned
- ⌚ Setting up development environment
- ⌚ Creating VSCode extension boilerplate
- ⌚ Implementing Puppeteer web automation
- ⌚ Building OpenAI-compatible API server
- ⌚ Adding authentication handling
- ⌚ Testing and optimization

## Project Structure

```
qwen-proxy/
├── .env.example
├── .gitignore
├── CHANGELOG.md
├── README.md
├── package.json
├── pyproject.toml
├── requirements.txt
├── src/
│   ├── extension.ts ..... # VSCode extension entry point
│   ├── server.ts ..... # Express server for API
│   ├── qwen-client.ts ..... # Puppeteer automation
│   └── types.ts ..... # TypeScript interfaces
├── test/
│   └── extension.test.ts
├── docs/
│   ├── setup.md
│   └── api.md
```

## Next Steps (Phase 1)

1. Set up development environment with uv
2. Create VSCode extension boilerplate
3. Implement basic Express server
4. Test extension activation and server startup

## Technologies Used

- **TypeScript** - Main development language
- **Puppeteer** - Web automation
- **Express** - API server
- **VSCode Extension API** - Extension framework
- **dotenv** - Environment variable management

## Installation Requirements

- Node.js 18+
- VSCode
- Chrome/Chromium browser