

Assignment: Building and Demonstrating Map Servers with OpenAI Agents SDK

Overview

In this assignment, you will explore the Model Context Protocol (MCP) concept, study examples of map servers currently available, and then design and implement 2–3 custom map servers using the OpenAI Agents SDK. Finally, you will record a short screencast demonstrating your agents' capabilities in real time.

Learning Objectives

By completing this assignment, you will be able to:

- Understand the fundamentals of the Model Context Protocol (MCP).
 - Analyze how existing map servers are structured and what features they provide.
 - Implement custom map servers as agent “tools” using the OpenAI Agents SDK.
 - Demonstrate and evaluate agent interactions with map services in real-world scenarios.
-

Prerequisites

- Familiarity with Python and async programming.
 - Installed OpenAI Agents SDK (latest version).
 - Basic knowledge of HTTP/JSON APIs.
 - Access to a Hugging Face account (for reading the article).
-

Part 1: Reading and Exploration

1. Read the MCP blog post on Hugging Face:
<https://huggingface.co/blog/Kseniase/mcp>
2. Explore existing open-source map servers to gain inspiration. You can browse examples from:
 - OpenStreetMap
 - MapLibre

- Leaflet Tile Providers
 - Any other open or public APIs offering mapping, routing, or geocoding services.
3. Write a short summary (300–400 words) highlighting:
 - The key MCP concepts discussed in the Hugging Face article.
 - Core features or design patterns you observed in existing map servers.
-

Part 2: Implementation

1. Select at least two map server ideas, either inspired by existing services or fully original.
 2. **Set up** an OpenAI Agents SDK project in Python.
 3. **Implement** each map server as an agent “tool” following MCP conventions:
 - Define `ServerParams` (e.g., commands, endpoints).
 - Expose at least three distinct operations (e.g., geocoding, routing, point-of-interest search).
 4. **Integrate** each server with an `AssistantAgent` (or similar) so the agent can route user queries to your servers.
 5. **Write unit tests** or simple interactive scripts to verify each server’s functionality.
-

Part 3: Demonstration

- **Record a 5–7 minute screencast** where you:
 1. Briefly introduce your project and the map servers you built.
 2. Show your agent in action—issue queries and demonstrate how it leverages each map server.
 3. Highlight any challenges faced and how you overcame them.
 - **Share your code repository** (e.g., GitHub link) and the video recording in your submission.
-

Deliverables

1. **Written Summary** of the article and existing map servers (PDF or Markdown).
2. **Source Code** for your agent project, including:
 - Python modules implementing each map server.
 - Configuration and setup instructions (README).
 - Tests or example scripts.
3. **Screencast Video** demonstrating your agents (hosted on a platform of your choice, e.g., YouTube unlisted, Zoom recording).
4. **Reflection** (1–2 paragraphs) on lessons learned and potential next steps.