

From Agents to APIs

Building Production-Ready AI Systems in Python with Google ADK & FastAPI



Ashmi Banerjee

PhD Candidate @ TU Munich

AI-GDE

@ashmi_banerjee

About Me



Photo from IWD Munich 2023



PhD Candidate at TU Munich (since 2022)

Researching Tourism Recommender Systems
MSc. TU Munich (2019)



Professional Experience

Previously: Data Scientist @ Deutsche Telekom, Intern @ MPI-SWS (2018-19)



AI/ML Google Developer Expert (GDE)

WTM Ambassador



Loves traveling (**45+ countries; 4 continents**) & outdoor activities



Prerequisites



Python 3.10 or later



IDE (E.g., Pycharm, VSCode, Antigravity, etc.)



Install uv (Python package manager in Rust) if you do not have it yet

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

Setup: Environment

1

Clone the Repository

<https://github.com/ashmibanerjee/google-adk-workshop>

```
git clone google-adk-workshop
```

2

Switch Branch

```
git checkout workshop
```

3

Install Dependencies

```
uv sync
```

Creates `.venv` and installs from `pyproject.toml`.

4

Activate Virtual Environment

```
source .venv/bin/activate
```



Scan to clone

Setup the .env file

```
google-adk-workshop/  
├── my_travel_planner/  
│   ├── agent.py      # Root agent definition  
│   ├── .env          # API keys (not committed)  
│   ├── run.py        # Agent runner & session setup  
│   ├── api.py        # FastAPI endpoint  
│   └── pyproject.toml # Project dependencies
```

Note: we do not commit .env files to git, so it will not be there when you clone the repo.

`touch my_travel_planner/.env`

Add your Gemini API key to it:

```
GOOGLE_API_KEY=your_gemini_api_key_here
```

```
GOOGLE_GENAI_USE_VERTEXAI=False
```

Setup: Credentials & Access

Option 1 (Recommended for this workshop)



Set your API keys through Google AI Studio

- Log in to Google AI studio
- Click on “*Get your API key*” (typically on the bottom of the left panel)

If using this API key then in the `.env` file set `GOOGLE_GENAI_USE_VERTEXAI=False` and add `GOOGLE_API_KEY = <your API key>`

Option 2



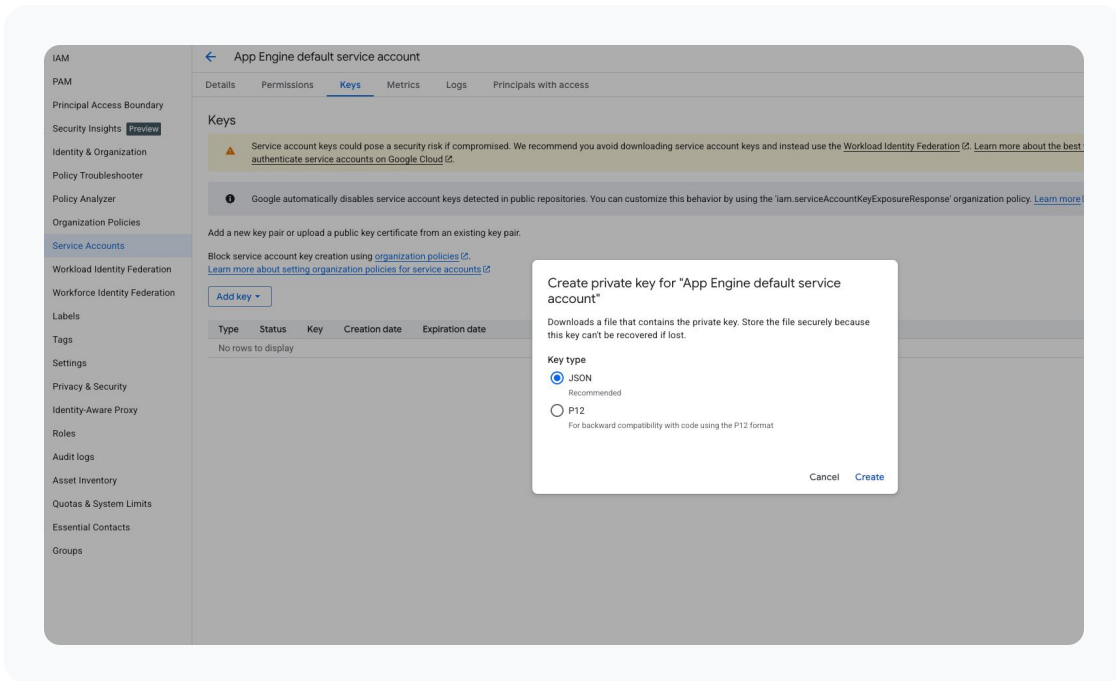
If using VertexAI then download `application_default_credentials.json`

If using this then in the `.env` file set

```
GOOGLE_GENAI_USE_VERTEXAI=True
```

```
GOOGLE_APPLICATION_CREDENTIALS=<path/to/your/application_default_credentials.json>
```

Downloading application.json from GCP



Step 1

Open Google Cloud Console

Step 2

Go to **IAM & Admin** > **Service Accounts**

Step 3

Create Service Account & assign roles (e.g., VertexAI)

Step 4

Manage Keys > Add Key > **Create New Key (JSON)**

Downloading application.json from GCP



CRITICAL WARNING

**Do NOT share this file with anyone!
!!!!**

Sharing this can lead to your accounts being compromised and you being charged **\$\$\$\$\$** in hours.

Part 1: Building a basic ADK agent

What is an Agent?

Core Reasoning Logic & Actions

An Agent contains the core reasoning logic of the application. It interprets user requests and decides what actions to take.



LLM
generate responses with an LLM

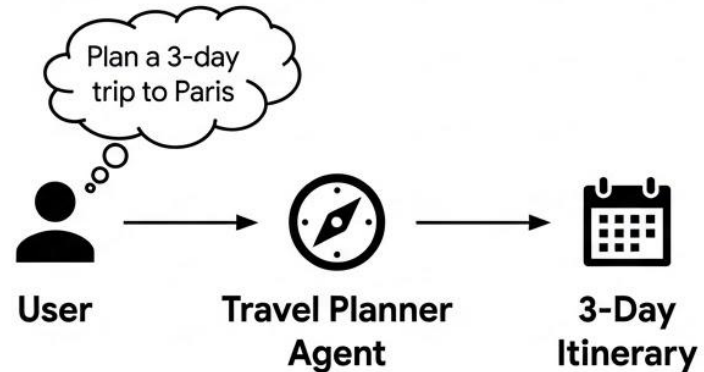


Tools & APIs
call tools or APIs



Collaboration
collaborate with other agents

In our Example: Travel Planner



Step 1: Creating your agent (1)



From the terminal (repo root dir):

1

```
adk create my_travel_planner
```

Note: You can skip this step here, since I already did it for you :)



Edit `.env` (if not done)

3

```
GOOGLE_GENAI_USE_VERTEXAI=True
GOOGLE_API_KEY= add your AI studio API key here
GOOGLE_APPLICATION_CREDENTIALS= <path to your application json file>
```



Edit `my_travel_planner/agent.py`

2

```
from google.adk.agents import Agent

def get_root_agent():
    # The Root Agent (The "Brain")
    travel_planner = Agent(
        name="travel_planner",
        model="gemini-2.5-flash",
        description="A comprehensive travel planning assistant.",
        instruction="""You are a world-class travel planner.
        Recommend places to visit based on the user's query."""
    )
    return travel_planner

root_agent = get_root_agent()
```

Step 1: Creating your agent (2)

4

From the terminal (repo root dir):

```
adk run my_travel_planner
```

 **Debugging Advice: Make sure you are in the virtualenv .venv in the terminal!**

```
(x.google-adk) ashmi@MacBook-Pro:~/code/adk$ cd main && adk run my_travel_planner
log setup complete: /var/folders/gk/1_hpk0b0v7p9ks2r_splw4000gn/1/agents_log/agent_20260906_151155.log
To access latest log: tail -f /var/folders/gk/1_hpk0b0v7p9ks2r_splw4000gn/1/agents_log/agent_latest.log
Running agent travel_planner, type exit to exit.
[user]: Where to go in March for a weekend? I like art & museums and am based in London.
[travel_planner]: As a Londoner with a love for art and museums, you're perfectly positioned to enjoy a fantastic weekend getaway in March! March weather can still be a bit chilly, making indoor activities like museum visits an ideal choice.

Here are a few recommendations, balancing ease of travel with world-class cultural offerings:

## **Top Picks for an Art-Filled Weekend in March:**

1. Paris, France
   * Why it's great: An undisputed capital of art and culture, Paris is incredibly easy to reach from London by Eurostar, making it perfect for a weekend. You can be enjoying a croissant by the Seine in just over two hours.
   * Art & Museums: The sheer number of iconic institutions is staggering. You could spend days in the Louvre (Mona Lisa, Winged Victory of Samothrace), the Musee d'Orsay (Impressionist masterpieces in a stunning former railway station), or the Centre Pompidou (modern and contemporary art). Don't forget the Musee de L'Orangerie for Monet's Water Lilies, or the charming Musee Rodin.
   * Travel: Eurostar from St Pancras International (Approx. 2h 20m).

2. Amsterdam, Netherlands
   * Why it's great: Compact, walkable, and boasting a fantastic "Museumplein" (Museum Square), Amsterdam is another excellent choice easily accessible by short flight or train.
   * Art & Museums: The Rijksmuseum houses Dutch Masters like Rembrandt and Vermeer, while the Van Gogh Museum is dedicated to the works of its namesake. For modern and contemporary art, the Stedelijk Museum is a must-visit. You can also explore numerous smaller galleries.
   * Travel: Short flights from London airports (Approx. 1h 15m) or Eurostar from St Pancras International (Approx. 3h 40m).

3. Berlin, Germany
   * Why it's great: Berlin offers a unique blend of historical significance and a vibrant contemporary art scene. It's a city that continuously reinvents itself, perfect for those who appreciate both classical and cutting-edge culture.
   * Art & Museums: Head to Museum Island, a UNESCO World Heritage site featuring five world-class museums, including the Pergamon Museum (currently undergoing renovations but sections are open, check their website) and the Altes Museum. For modern art, explore the Hamburger Bahnhof - National Gallery of Contemporary Art or the famous East Side Gallery, an open-air gallery on a section of the Berlin Wall.
   * Travel: Flights from London airports (Approx. 1h 45m).

## Alternatively: A London Staycation

If you prefer to maximise your time without any travel, London itself offers an unparalleled array of museums and galleries. You could dedicate a whole weekend to an area you haven't fully explored, or revisit old favourites with a fresh perspective.

* Ideas: Explore the British Museum in Bloomsbury, wander through the numerous galleries in Kensington (V&A, Science Museum, Natural History Museum), or immerse yourself in contemporary art at the Tate Modern on the South Bank.

No matter which you choose, March is a great time to dive into indoor cultural experiences. Enjoy planning your art-filled weekend!
[user]:
```

Step 1: Creating your agent (2)

4

From the terminal (repo root dir):

```
adk run my_travel_planner
```

```

(google-adk) ashmi@Mac: ~ % cd /Users/ashmi/Projects/adk ; adk run my_travel_planner
log setup complete: /var/folders/gk/1_hpk0b0v7p9ks2r_splw40000gn/1/agents_log/agent_20260306_151155.log
To access latest log: tail -f /var/folders/gk/1_hpk0b0v7p9ks2r_splw40000gn/1/agents_log/agent_latest.log
Running agent travel_planner, type exit to exit.
[user]: Where to go in March for a weekend? I like art & museums and am based in London.
[travel_planner]: As a Londoner with a love for art and museums, you're perfectly positioned to enjoy a fantastic weekend getaway in March! March weather can still be a bit chilly, making indoor activities like museum visits an ideal choice.

Here are a few recommendations, balancing ease of travel with world-class cultural offerings:

## **Top Picks for an Art-filled Weekend in March:**

1. **Paris, France**
   - **Why it's great:** An undisputed capital of art and culture, Paris is incredibly easy to reach from London by Eurostar, making it perfect for a weekend. You can enjoy a croissant by the Seine in just over two hours.
   - **Art & Museums:** The sheer number of iconic institutions is staggering. You could spend days in the **Louvre** (Mona Lisa, Winged Victory of Samothrace), the **Musée d'Orsay** (Impressionist masterpieces in a stunning former railway station), or the **Centre Pompidou** (modern and contemporary art). Don't forget the **Musée de l'Orangerie** for Monet's Water Lilies, or the charming **Musée Rodin**.
   - **Travel:** Eurostar from St Pancras International (Approx. 2h 20m).

2. **Amsterdam, Netherlands**
   - **Why it's great:** Compact, walkable, and boasting a fantastic "Museumplein" (Museum Square), Amsterdam is another excellent choice easily accessible by short flight or train.
   - **Art & Museums:** The **Rijksmuseum** houses Dutch Masters like Rembrandt and Vermeer, while the **Van Gogh Museum** is dedicated to the works of its namesake. For modern and contemporary art, the **Stedelijk Museum** is a must-visit. You can also explore numerous smaller galleries.
   - **Travel:** Short flights from London airports (Approx. 1h 15m) or Eurostar from St Pancras International (Approx. 3h 40m).

3. **Berlin, Germany**
   - **Why it's great:** Berlin offers a unique blend of historical significance and a vibrant contemporary art scene. It's a city that continuously reinvents itself, perfect for those who appreciate both classical and cutting-edge culture.
   - **Art & Museums:** Head to **Museum Island**, a UNESCO World Heritage site featuring five world-class museums, including the **Pergamon Museum** (currently undergoing renovations but sections are open, check their website) and the **Altes Museum**. For modern art, explore the **Hamburger Bahnhof - National Gallery of Contemporary Art** or the famous **East Side Gallery**, an open-air gallery on a section of the Berlin Wall.
   - **Travel:** Flights from London airports (Approx. 1h 45m).

## **Alternatively: A London Staycation**

If you prefer to maximise your time without any travel, London itself offers an unparalleled array of museums and galleries. You could dedicate a whole weekend to an area you haven't fully explored, or revisit old favourites with a fresh perspective.

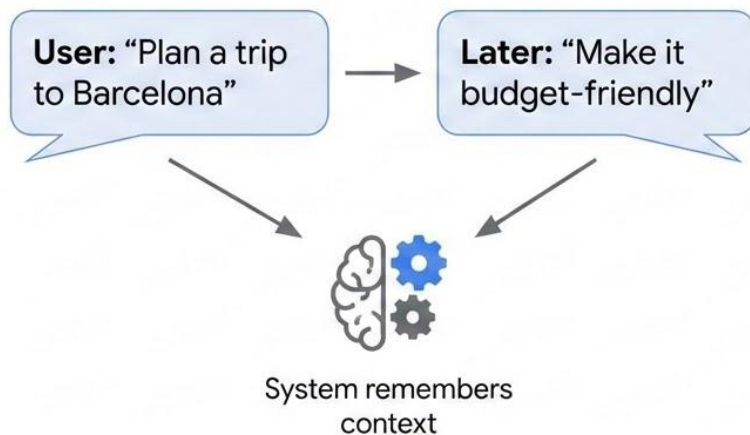
* **Ideas:** Explore the **British Museum** in Bloomsbury, wander through the numerous galleries in Kensington (V&A, Science Museum, Natural History Museum), or immerse yourself in contemporary art at the **Tate Modern** on the South Bank.

No matter which you choose, March is a great time to dive into indoor cultural experiences. Enjoy planning your art-filled weekend!
[user]:
  
```



Managing Conversation State

Why sessions matter: Agents need memory across messages.



ADK sessions store:



Conversation history



User information



Intermediate agent state

Executing the Agent: The Runner



The **Runner** orchestrates the full interaction.

Step 2.1: Setup Session & Runner

Edit run.py

```
APP_NAME = "travel-planner_app"
USER_ID = "user_1"
SESSION_ID = "session_001"

async def setup_session_and_runner(root_agent: Agent = None, session_id: str = SESSION_ID):
    # Setup Runner for execution
    session_service = InMemorySessionService()
    session = await session_service.create_session(app_name=APP_NAME, user_id=USER_ID, session_id=session_id)
    runner = Runner(agent=root_agent, app_name=APP_NAME, session_service=session_service)
    return session, runner
```

 InMemorySessionService keeps everything in RAM

 The **Runner** orchestrates the full interaction.

Step 2.2: Call Async Agent()

 Edit run.py

```
from google.genai import types

async def call_agent_async(query: str, root_agent: Agent = None, session_id: str = SESSION_ID) → str:
    content = types.Content(role='user', parts=[types.Part(text=query)])
    session, runner = await setup_session_and_runner(root_agent=root_agent, session_id=session_id)
    events = runner.run_async(user_id=USER_ID, session_id=session_id, new_message=content)
    final_response_text = "No response received."
    async for event in events:
        # Key Concept: is_final_response() marks the concluding message for the turn.
        if event.is_final_response():
            if event.content and event.content.parts:
                final_response_text = event.content.parts[0].text
            elif event.actions and event.actions.escalate:
                final_response_text = f"Agent escalated: {event.error_message or 'No specific message.'}"
            break

    print(f"<<< Agent Response: {final_response_text}")
    return final_response_text
```

Step 2.3: Run Agent Pipeline

● ● ● Edit run.py

```
async def run_agent_pipeline(query: str) → str:
    root_agent = get_root_agent()
    return await call_agent_async(query=query, root_agent=root_agent, session_id=SESSION_ID)

if __name__ == "__main__":
    user_query = ("I'm planning a trip to Paris in the spring. What are some must-see attractions and local events "
                 "during that time?")
    asyncio.run(run_agent_pipeline(query=user_query))
```

Step 2.3: Run Agent Pipeline

Edit `run.py`

```
async def run_agent_pipeline(query: str) → str:
    root_agent = get_root_agent()
    return await call_agent_async(query=query, root_agent=root_agent, session_id=SESSION_ID)

if __name__ == "__main__":
    user_query = ("I'm planning a trip to Paris in the spring. What are some must-s
                  "during that time?")
    asyncio.run(run_agent_pipeline(query=user_query))
```



Part 2: Connecting agent with FastAPI

What is FastAPI?

FastAPI is a modern Python framework for building **APIs quickly and efficiently**.

Key Features

- ⚡ **High performance** (Starlette & Uvicorn)
- 🐍 Designed for **Python 3.7+**
- 🧠 Uses **type hints** for validation

Advanced Capabilities

- 📄 **Auto API docs** (Swagger/OpenAPI)
- 🔌 Easy integration with **ML & LLMs**

Why use it in AI apps?

- **Quickly expose** models or agents as robust APIs
- **Seamless connection** to frontends, tools, and orchestration services

What does FastAPI enable?

1. Reusable Services

- Expose agents via APIs (`/chat`, `/recommend`)
- Accessible from frontend, CLI, or services

2. Decoupling

- Frontend doesn't need agent logic
- Communicates via API for faster iteration

3. Orchestration

- Deploy agents as separate microservices
- Enables pipelines & distributed systems

4. Production Ready

- Async support & Data validation
- Auto API docs (Swagger)
- Easy deployment (Docker, K8s)

Step 0: Initialize FastAPI

 Edit `api.py`

```
from fastapi import FastAPI
app = FastAPI(title="Travel Planner AI", description="Powered by Google ADK + Gemini")
```

Step 1: Define Schemas (optional)

 Edit `api.py`

```
from pydantic import BaseModel
class QueryRequest(BaseModel):
    query: str

class QueryResponse(BaseModel):
    query: str
    response: str
```

Step 2: Create the endpoint

● ● ● Edit `api.py`

```
from run import run_agent_pipeline

@app.post("/ask", response_model=QueryResponse)
async def ask_agent(request: QueryRequest):
    response = await run_agent_pipeline(query=request.query)
    return QueryResponse(query=request.query, response=response)
```

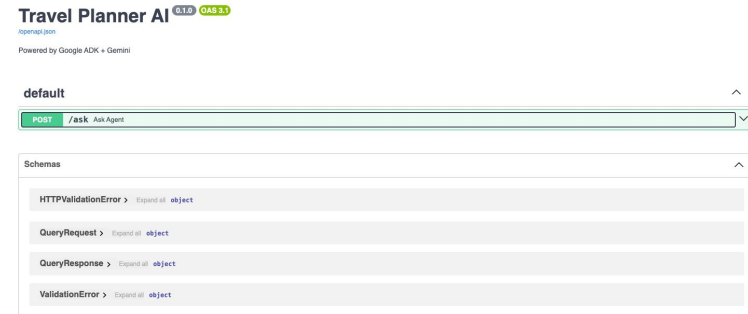
Step 3: Run the FastAPI Server

Execution Command

```
uv run uvicorn api:app --reload
```

Local Endpoints

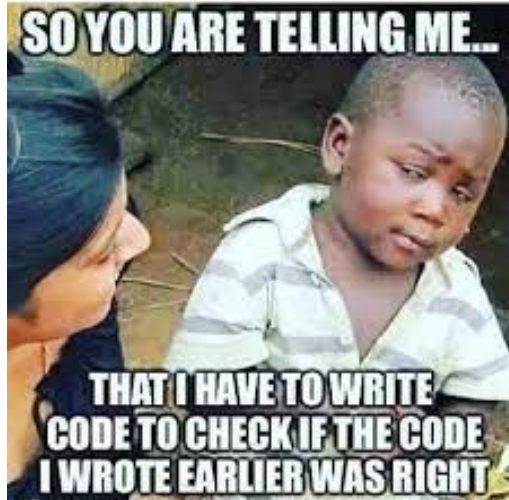
- Server URL: <http://localhost:8000>
- Swagger UI: <http://localhost:8000/docs>





Next Steps

Test



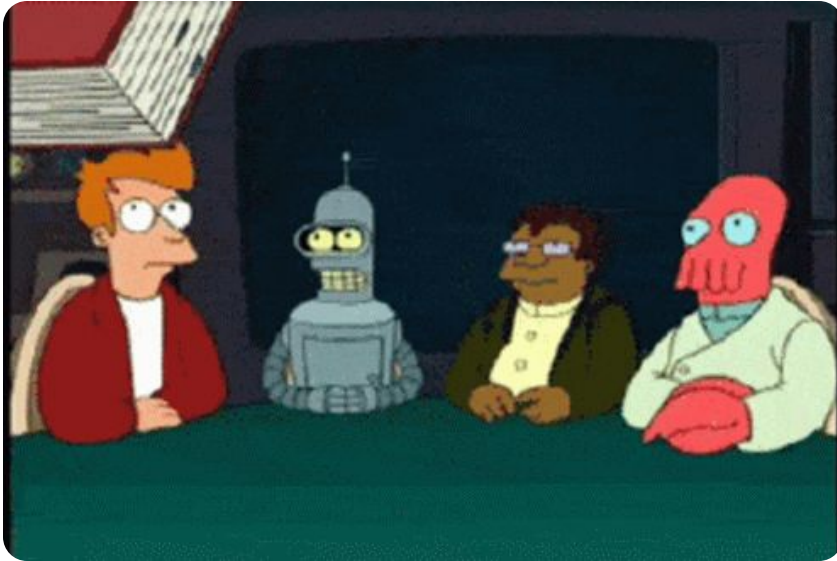
Containerize



Deploy



Recap



01. Agent Building

Building LLM agents using Google ADK

02. API Integration

Connecting the ADK agent with a FastAPI endpoint

03. Deployment

Potential ways to deploy the application

Thank You! Time for Q&A!



Talk Feedback

<https://bit.ly/ashmib-feedback>



Ashmi Banerjee

PhD Candidate @ TU Munich,
ML GDE, WTM Ambassador
@ashmi_banerjee

Scan me for feedback!

