



Crea il tuo primo agente con ADK

Una guida di base per capire cosa stai facendo

Questa guida ti accompagna nella creazione di un agente AI con ADK, spiegando i passaggi essenziali e il loro scopo. Non serve soltanto a copiare dei comandi: serve a capire come si definiscono il ruolo di un agente, le sue regole e il formato delle sue risposte.

Costruiremo Ubaldo, uno chef pragmatico e allergico agli sprechi, che inventa ricette con quello che si trova in frigo.

I comandi del tutorial si eseguono in un terminale Unix. Su Windows useremo WSL con Ubuntu; su Linux puoi usare il terminale della tua distribuzione e su macOS l'app Terminale. Per chi usa Linux o macOS, i passaggi per preparare il terminale e installare i programmi dipendono dal proprio sistema; i comandi per creare e avviare il progetto sono gli stessi.

1. Prepara il terminale

Se usi Windows

Apri PowerShell come amministratore e lancia:

```
wsl --install
```

Riavvia il computer se richiesto. Poi apri il menu Start, cerca Ubuntu e avvialo.

Al primo avvio, Ubuntu ti chiederà di creare un nome utente e una password. La finestra che si apre è il terminale: da lì inserirai i comandi della guida.

Se usi Linux o macOS

Apri il terminale che usi normalmente. Non devi installare WSL né seguire i passaggi Windows di questa sezione.

2. Installa Python e VSCodium

Se usi Windows con WSL e Ubuntu

Nel terminale di Ubuntu, installa Python, gli strumenti per gli ambienti virtuali, pip, Git e quelli necessari per aggiungere il repository di VSCodium:

```
sudo apt update
sudo apt install python3 python3-venv python3-pip git curl gpg -y
```

VSCodium è un editor di codice con un'interfaccia grafica: ti permette di aprire la cartella del progetto e modificare i file senza lavorare soltanto dal terminale. È una distribuzione open source basata su Visual Studio Code.

Potremmo usare editor da terminale come nano, vim o emacs, ma per questa guida scegliamo VSCodium perché è più immediato vedere e modificare i file.

Aggiungi il repository di VSCodium:

```
curl -fsSL https://repo.vscodium.dev/vscodium.gpg \
| gpg --dearmor \
| sudo dd of=/usr/share/keyrings/vscodium.gpg
```

```
sudo curl --output-dir /etc/apt/sources.list.d \
-L0 https://repo.vscodium.dev/vscodium.sources
```

Poi installalo:

```
sudo apt update
sudo apt install codium -y
```

Se usi Linux o macOS

Installa Python 3, Git e VSCodium con il metodo previsto dal tuo sistema. Non devi seguire i comandi d'installazione specifici per Ubuntu qui sopra.



3. Crea la cartella del progetto e l'ambiente Python

Nel terminale, crea la cartella principale della serie ed entra al suo interno:

```
mkdir adk-svuota-frigo
cd adk-svuota-frigo
```

Crea un ambiente virtuale Python:

```
python3 -m venv .venv
```

Attivalo:

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

Se l'attivazione è riuscita, all'inizio della riga del terminale vedrai (.venv). L'ambiente virtuale tiene separati i pacchetti di questo progetto da quelli degli altri programmi Python.

Installa ADK:

```
pip install google-adk
```

4. Crea una chiave per Google AI Studio

Prima di creare l'agente, accedi a [Google AI Studio](#) e crea una chiave API. Il servizio offre un'opzione gratuita: può essere un po' lenta e ha limiti di utilizzo, ma per seguire questa guida è sufficiente.

Con un account gratuito, durante le prove potresti incontrare questi errori:

- 429: hai raggiunto temporaneamente un limite di richieste o di utilizzo.
- 503: il servizio è momentaneamente non disponibile o non riesce a gestire la richiesta.

Possono capitare per i limiti del piano gratuito o per la disponibilità momentanea del servizio. Non significano necessariamente che ci sia un problema nel codice o nell'agente: aspetta qualche secondo e invia di nuovo la richiesta.

Tieni la chiave a portata di mano: la inserirai durante la procedura guidata di ADK. È personale: non inserirla nel codice e non pubblicarla su GitHub.

5. Crea il progetto ADK

Assicurati di trovarti nella cartella principale adk-svuota-frigo, quella che contiene l'ambiente .venv. Poi lancia:

```
adk create step1_agente_base
```

adk create avvia la creazione guidata di un nuovo progetto ADK. step1_agente_base è il nome della cartella che conterrà i file iniziali dell'agente.

La procedura ti chiederà prima quale modello usare. Per questa guida, scegli gemini-3.8-flash:

```
Choose a model for the root agent:
1. gemini-3.5-flash
2. gemini-3.8-flash
3. Other models (fill later)
Choose model (1, 2, 3): 2
```

Poi ti chiederà quale backend usare:

```
1. Google AI
2. Vertex AI
3. Login with Google
Choose a backend (1, 2, 3):
```

Scegli Google AI, l'opzione che useremo con la chiave creata in [Google AI Studio](#). Quando richiesto, inserisci la chiave API.

Le altre opzioni permettono di collegarsi tramite Vertex AI oppure di autenticarsi con il proprio account Google.



6. Apri il progetto e inserisci il codice di Ubaldo

Entra nella cartella creata da ADK:

```
cd step1_agente_base
```

Aprila con VSCodium:

```
codium .
```

Nel progetto, apri agent.py e sostituisci il contenuto con il codice di Ubaldo:

```
from google.adk.agents import Agent
from google.genai.types import GenerateContentConfig, ThinkingConfig
root_agent = Agent(
    name="ubaldo_chef", model="gemini-3.8-flash",
    generate_content_config=GenerateContentConfig(thinking_config=ThinkingConfig(thinking_level="low")),
    description=(
        "Ubaldo, uno chef burbero ma dal cuore d'oro, pragmatico e allergico agli sprechi, "
        "che inventa ricette lampo con quello che si trova in frigo."
    ),
    instruction="""
Ti chiami Ubaldo. Sei un cuoco di vecchia scuola, pragmatico, schietto e intollerante agli sprechi.
Non fai complimenti inutili, parli in modo diretto e un po' ruvido, ma alla fine aiuti sempre chi è in
difficoltà davanti al frigo vuoto.

REGOLE FERREE (NON VIOLARE MAI):
1. Usa SOLO gli ingredienti che l'utente ti elenca. Gli unici extra concessi sono:
acqua, sale, pepe e olio extravergine d'oliva.
2. Tempo massimo di preparazione + cottura: 15 minuti.
3. Pentolame massimo consentito: 1 sola padella (o pentola) + 1 tagliere.
4. Se l'utente ti elenca cose non commestibili o assurde, riprendilo aspramente.

FORMATO DI RISPOSTA OBBLIGATORIO:
Rispondi sempre e solo usando questa scheda:
- 🍴 NOME DEL PIATTO: (Un nome onesto ma rustico)
- ⌚ TEMPO E DIFFICOLTÀ: (Stima reale in minuti)
- 📋 PIANO D'AZIONE CRONOLOGICO: (Passaggi numerati secchi e dritti al punto)
- 💡 IL CONSIGLIO DI UBALDO: (Una dritta pratica per non rovinare tutto)
"""
)
```

Il ruolo, le regole e il formato della risposta sono definiti nelle istruzioni passate all'agente. In questa prima versione Ubaldo non ha ancora strumenti per leggere la dispensa o aggiornare la lista della spesa: quelli arriveranno nelle puntate successive.

7. Avvia e prova l'agente

Dal terminale, assicurati di essere dentro step1_agente_base e che (.venv) sia attivo. Avvia l'interfaccia di sviluppo di ADK:

```
adk web
```

Apri nel browser l'indirizzo indicato dal terminale, di solito:

```
http://localhost:8000
```

Seleziona l'agente e prova, per esempio:

Ho due uova, mezzo zucchino, un barattolo di ceci e del rosmarino. Ho 15 minuti.

Controlla che Ubaldo proponga una ricetta rispettando i suoi limiti e usando la scheda prevista. Se compare un errore 429 o 503, aspetta qualche secondo e invia di nuovo la richiesta. Puoi anche elencare un ingrediente assurdo o non commestibile e vedere come reagisce.

In questa prima puntata abbiamo creato un agente con un ruolo preciso, regole esplicite e un formato di risposta definito. Nelle prossime gli daremo strumenti per leggere la dispensa, aggiornare la lista della spesa e collaborare con altri agenti.