Make your agents AI Engine compatible

When you want to run an agent on your own hardware or infrastructure locally, this is easy to do on any system that support Python > 3.10.

Introduction

This guide demonstrates how to run an agent on your own hardware or infrastructure, making it accessible overAgentverse andDeltaV using theMailroom ↗ . The example uses a locally hosted agent that utilizes LangChain's Wikipedia integration to process requests related to Wikipedia search. To get this agent to beDeltaV ↗ accessible, we need to create a newAgent Function ↗ on the Agentverse for the agent to then allow it to be found in DeltaV.

Prerequisites

- Python 3.10 or newer.
- AnAgentverse ↗(opens in a new tab)
- account to create the agent Functions and Mailroom's mailbox.
- uagents
- andlangchain_community
- libraries installed in your Python environment:* pip install uagents
-
    - ,pip install langchain_community
-
    - .
- pip install wikipedia
- to install Wikipedia.

Agent Configuration

Configure your agent script LangChain's Wikipedia integration for handling Wikipedia search requests.

Agent

from langchain_community . tools import WikipediaQueryRun from langchain_community . utilities import WikipediaAPIWrapper from uagents . setup import fund_agent_if_low from uagents import Agent , Context , Protocol , Model from pydantic import Field from ai_engine import UAgentResponse , UAgentResponseType

Extend your protocol with Wikipedia data fetching

class

WikiReq (Model): search_keyword :

str

=

Field (description = "This describes the keyword you want to search on wiki")

SEED_PHRASE

""

Copy the address shown below

print (f "Your agent's address is: { Agent (seed = SEED_PHRASE).address } ")

AGENT_MAILBOX_KEY

"Your_mailbox_address"

Now your agent is ready to join the agentverse!

WikiAgent

Agent (name = "Wiki Agent" , seed = SEED_PHRASE, mailbox = f " { AGENT_MAILBOX_KEY } @https://agentverse.ai" ,) fund_agent_if_low (WikiAgent.wallet. address ())

funding agent.

wiki_protocol

Protocol ("Wiki Protocol")

@wiki_protocol . on_message (model = WikiReq, replies = {UAgentResponse}) async

def

load_dalle (ctx : Context ,

sender :

str ,

msg : WikiReq): wikipedia =

WikipediaQueryRun (api_wrapper = WikipediaAPIWrapper ()) ctx . logger . info (msg.search_keyword) try : result = wikipedia . run (msg.search_keyword) except

Exception

as e : ctx . logger . info (f "Error generating response: { e } ")

Send an error response back to the user

await ctx . send (sender, UAgentResponse (message = str (result), type = UAgentResponseType.FINAL))

WikiAgent . include (wiki_protocol, publish_manifest = True) WikiAgent . run () Run the script in local machine usingpython agent.py and get the agent's address. Use this agent's address to create amailbox ↗ by heading over to theAgentverse: My Agents ↗ and click onLocal Agents tab. By clicking theConnect Local Agent button and providing the address of the agent we defined above and then the name, you will then be provided with aMailbox API Key to be replaced replace within theYour_Mailbox_key field.

ℹ Remember to replaceYour_Mailbox_key key with Agentverse Mailbox API Key. Usingon_message handler, agent takes the search keyword and returns the information with that keyword to the DeltaV agent for that address.

.run() initialises the agent.

Rerun the scriptpython agent.py , this will initialize the agent so agent can receive messages, and other agents know where to communicate with them.

We define ourprotocol , which is just an string as seen in theWikiRequest object.

Finally, we run our agent as follows:python agent.py

Expected output :

Your agent's address is: agent1qd2gnx72un4yzhxjqlx0z6gr39e8q9v03ehgprrphu9evuhszvrsgmggyjh INFO: [Wiki Agent]: Manifest published successfully: Wiki Protocol INFO: [Wiki Agent]: Almanac registration is up to date! INFO: [Wiki Agent]: Connecting to mailbox server at agentverse.ai INFO: [Wiki Agent]: Mailbox access token acquired

Register your Agent Function on the Agentverse

For this example we set up a really simpleAgent Function . For further information on Agent Functions and registration process, seeAgentverse Functions ↗ andRegister Agent Functions on the Agentverse ↗ resources.

To registerLocal Agents and Functions on the Agentverse, you will first need to log in theAgentverse ↗(opens in a new tab) and head over to theMy Agents tab. Then, click onLocal Agents tab and click one of theConnect Local Agent buttons.

You will need to provide thelocal agent address and make sure it is running on your terminal as only running agents can enroll Agent Functions on the Agentverse!

You can now provide the needed details for your Agent Function in the dedicated fields. Remember to provide detailed descriptions for what yourAgent Function does and theFields for data Models expected.

Interacting on DeltaV

Then we head over toDeltaV ↗(opens in a new tab) and get theAI Engine ↗ to interact with our agent on our behalf.

Follow the above steps so that you can run an agent, create a function for the agent and then have that agent accessible by DeltaV.

Was this page helpful?

Agent Functions Introducing dialogues