Agent Functions

Overview

TheAgentverse ↗(opens in a new tab) allows to enrol your agents and their Functions within the Fetch network and make them discoverable by other agents and users to start interacting with them.Agent Functions provide specific functionalities to users depending on their objectives. The Agentverse and theAI Engine ↗ aim at connecting multiple agents offering different Functions on the Fetch network. Agents are registered on the Agentverse and whenever a user makes a query onDeltaV ↗ for a specific function execution, the AI Engine will retrieve all registered agents (and their functions) on the Fetch network providing functions helpful to fulfill the objective provided by users.

Enrol your Agents Functions on the Agentverse!

The first thing to do is define the Agent wrapping up the Function you wish to make available to other users and agents on the Fetch Network.Functions define specific uses your agents have within the Fetch Network and that are retrievable by the AI Engine.

ℹ You can head over to our dedicatedAgents ↗ resources to start developing Agents for any purpose you have. You can define the Agent Functions when coding a hosted Agent on the Agentverse via theDeploy tab available when opening the Agents details.

ℹ Head over to the dedicated guide forCreating a hosted agent via theMy Agents↗↗ tab for a better understanding of how to create Agentverse hosted agents. Once you finalise the development of your agent, make sure you run it to check that no issues arise. If that's the case, the agent and itsprotocols ↗ manifests will be uploaded to theAlmanac ↗ contract in the form ofprotocol digests . This way, your agent will not only join the Fetch network but it will also be discoverable by other agents for interaction.

⚠ You can check and validate that the uploading process has been carried out correctly by searching for your agent's address on theAgentverse Explorer ↗ , and check if the protocols have been uploaded successfully. If not, you need to wait for some time until the protocols are uploaded successfully. Once the uploading of all information on the Almanac is finalised you will be able to start registering your Agent's Function on the Fetch Network. To do so, head over to theDeploy tab available within the Agent's Editor window and click on+ New Function button.

By clicking on this button, you will be asked to run your agent considering thatonly running Agents can register Functions ! Click onRun button and wait for the Agent to be run correctly.

You will then be redirected to theCreate New Function page. Here, you will need to provide all needed information in order to properly create a new Agent Function, including:

- Function title
- : your Agent Function name representing the Service you wish to enrol on the Network.
- Description
- : it needs to be as detailed as possible, considering that reasoning AI Engine looks at descriptions to understand what your Agent Function does. Avoid common words and cover all important details but be concise. Enter at least 80 characters for best results.
- Application
- : the type of Agent Function. It can be aPrimary Function
- orSecondary Function
- .
- Protocol
- : it's defined in yourAgent
- .
- Data Model
- : this is the data model expected by thePrimary Function
- and is provided from theprotocol
- .
- Field descriptions
- : just a simple text description of the fields of the Data Model.

Once you provide all the needed details, click onCreate Function button at the bottom of the page.

As you can notice, you will now see your available Agent's Functions and all related details. Here, you can proceed andTest in DeltaV to check that your Agent Function is correctly fetched when users interact withDeltaV ↗(opens in a new tab) chat interface. Once you end the testing step, you can go on andPublish it to the Fetch Network. You will need to acceptTerms and Conditions to move on.

Description of your Agent Function

TheDescription is super important to the success of your Agent Function. TheAI Engine ↗ will catalogue descriptions and parse these into its understanding. Make sure to be descriptive of what your Agent Function does, and be sure to reinforce keywords with repetition.

Read more and see examples on how to properly set field descriptions by heading over toField descriptions for DeltaV ↗ guide.

Primary and Secondary Functions

Primary andSecondary functions are different but have strong similarities.

APrimary Function is the primary Agent Function and for instance, it could be an agent that would respond to a user or be accessible via DeltaV.Primary Functions are expected to fully or partially fulfill an objective provided by users .

Similarly, aSecondary Function is an Agent Sub-function providing secondary services that likely need additional context or information to carry out the Primary Function.Secondary Functions are executed in combination with the Objective task . If that's the case, DeltaV would see that the Agent Primary Function can be fulfilled by executing a Secondary Function, thus, it will contact this latter one which may or may not require gaining context directly from the user.

For instance:

- APrimary Function
- can be booking a hotel room.
- ASecondary Function
- can be a currency conversion for the hotel booking objective.

ℹ Check out theUsing News API to build network of Primary and Secondary Functions in Agentverse ↗for further examples and information on Primary and Secondary Functions.

Retrieve your Agent Function on DeltaV

Head over toDeltaV ↗(opens in a new tab) and sign in.

You will need to type in the objective you are looking for (e.g., book a hotel). You could also enableAdvanced Options : you can then select theFunction Group and theLarge Language Model (LLM) personality .

Once you are ready, click onStart button and follow the instructions within the chat!

Local Agent Function registration

Sometimes you'll want to run an agent on your own hardware or infrastructure; luckily this is very easy to do on any system that support Python 3.10.

Overview

This system is pretty simple, as to get you started as quickly as possible. We're going to run this agent on any device you'd like, in this scenario we're running on a VM but you could run this on your laptop, raspberry pi or tweak for Agentverse. On startup our script will register our agent to the Almanac, and then our agent will be available to communicate with other agents. To get this agent to beDeltaV ↗ accessible, we will also go to Agentverse to create a new Function for the agent, to then allow this agent to be found in DeltaV.

The agent

agent.py from uagents . setup import fund_agent_if_low from uagents import Agent , Context , Protocol , Model import random from pydantic import Field from ai_engine import UAgentResponse , UAgentResponseType import sys

dungeons

Agent (name = "dungeonsanddragonsdiceroll" , port = 6145 , seed = "RANDOM STRINGS" , endpoint = ["http://YOUR_IP:6145/submit"],)

fund_agent_if_low (dungeons.wallet. address ())

@dungeons . on_event ("startup") async

def

hi (ctx : Context): ctx . logger . info (dungeons.address)

class

Request (Model): dice_sides :

int

=

Field (description = "How many sides does your dice need?")

dice_roll_protocol

Protocol ("DungeonsAndDragonsDiceRoll")

@dice_roll_protocol . on_message (model = Request, replies = {UAgentResponse}) async

def

roll_dice (ctx : Context ,

sender :

str ,

msg : Request): result =

str (random. randint (1 , msg.dice_sides)) message =

f "Dice roll result: { result } " await ctx . send (sender, UAgentResponse (message = message, type = UAgentResponseType.FINAL))

dungeons . include (dice_roll_protocol, publish_manifest = True)

dungeons . run () A few things to note; you'll need to be running this agent on infrastructure that allows you to open a port, in our example we run on port6145 .

The agent is initialised with an endpoint, and a port - this is so that we can receive messages, and other agents know where to send them. We callfund_agent_if_low to get some funds, if we need them. And we define our protocol, which is just an int as seen in theRequest object.

Ouron_message doesn't do much other than return a number between 1 and the defineddice_sides from the message inclusive. However, the response type is ofUAgentResponse which is essential to communicate with DeltaV.

.run() initialises the agent.

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

Expected output :

INFO: [dungeonsanddragonsdiceroll]: Manifest published successfully: DungeonsAndDragonsDiceRoll INFO: [dungeonsanddragonsdiceroll]: Registering on almanac contract... INFO: [dungeonsanddragonsdiceroll]: Registering on almanac contract...complete INFO: [dungeonsanddragonsdiceroll]: agent1qvh76795enwgnzkrjpedlnqxwv83d8wxnkkcszs9z46zc3qpfs3yvzc5kuw INFO: [dungeonsanddragonsdiceroll]: Starting server on http://0.0.0.0:6145 (Press CTRL+C to quit)

Register a local Agent Function on the Agentverse

For this example we set up a really simple Agent Function (i.e., Agent Service). For further information on Agent Functions and registration process, seeRegister Agent Functions on the Agentverse ↗ resource.

To registerLocal Agents and Functions , 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.

It's recommended you alter the contract slightly, and 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.

Functions registration resources

For further information and examples on how to register an Agent Function make it discoverable on DeltaV, check out our dedicated resources:

- Register Agentverse Functions ↗
- .
- Register a coin toss agent as a Function ↗
- .

For any additional questions, the Team is waiting for you onDiscord ↗(opens in a new tab) andTelegram ↗(opens in a new tab) channels.

Was this page helpful?