Creating an Agentverse hosted agent

Introduction

On theAgentverse ↗(opens in a new tab) you can create and host any type of agent you want to create using theMy Agents ↗ tab. Managed agents are currently a beta feature and therefore do not fully support the entireAgents ↗ toolset for development. Improvements and upgrades are planned for the future!

ℹ Check out theAgentverse: allowed imports ↗ guide for a better understanding of the packages available for development of Agents on the Agentverse. Let's start by heading over to theAgentverse: My Agents tab, click on the+ New Agent . Here, there are 2 different ways you can start creating your agent. You can either choose to create an agent:

1. From aBlank Agent
2. .
3. From aSkeleton Agent
4. .

Creating an agent from a blank script

You can create your AI Agent from a blank script by navigating towards theAgentverse ↗(opens in a new tab) , signing in and choosing theMy Agents tab. Here, click on the+ Agents button.

Here, click on the+ New Agent button and chooseBlank Agent .

You will need to provide aname for the agent you wish to create.

Once you select it, your agent will be ready to be coded and a box will appear in theMy Agents page with different data being displayed about the agent you have just created, including thename and theaddress of the agent.

If you click on your agent box, the code editor will be displayed. This is theAgent Editor ; it allows you to create, edit and refine the code for the agent you have in mind all in one place. The agent editor provides various basic information about your agent, including theagent address andwallet address . For additional information on Agent addresses, have a look at our dedicated documentation:Agents address ↗ .

The Agent Editor has an integratedagent.py and.env files that allow you to start programming immediately! The Agent Editor offers you the option of structuring your project into several file windows, which are located on the left-hand side of the editor window. All you have to do is enter a name for the new file to be added. An additional feature of the Agent Editor is theAgent Logs ; a built-in terminal that displays the output of your script after it has been executed. The Agent Logs provides multiple filters for the different log levels.

The Agent Logs feature is available by clicking thethird bottom icon on the left side of the Agent Editor. The first and second buttons enable you to exploreAgents Secrets andAgent Storage features.

ℹ This way, we aim at making agents' development as quick and efficient as possible. This is a great tool to determine if your code runs smoothly and to check if any error arises and solve it immediately!

Create your first hosted agent on the Agentverse!

You are now ready to start using the Agentverse to create and edit your agents in few minutes! Let's begin with creating a simple agent printing"Hello world!" in the Agent Logs on an interval of 3 seconds by using the.on_interval() decorator.

Here's an example:

To run the agent, simply click on theRun button. You can stop the script by clicking on theStop button.

The output will be printed in the Agent Logs window:

ℹ Within the Agent Editor, you can import and utilize a predefined set of modules to develop your ideas. These pre-approved modules offer a diverse range of functionalities, allowing you to build complex agents. For further information on modules being available within the Agentverse Editor, have a look at the following resource:Agentverse: allowed imports ↗ .

Create an agent based on a template

You can also create an agent by choosingSkeleton Agent or one of the template options available within theGetting Started ,DeltaV Integration ,Smart Services or theAI/ML tabs. A description of each template agent is provided when hovering above the option you want to choose.

There are different categories you can choose from:

1. Getting Started
2. : this category helps you in creating basic agents using the Agents technology.
3. You can:
4.
    - Create your first agent.
5.
    - Send messages between two agents.
6.
    - Storing data.
7.
    - Organize your agents with protocols.
8.
    - Create an agent requesting data from a website.
9.
    - Create an agent sending tokens to another agent.
10.
    - Create an agent interacting with an oracle smart contract.
11. AI/ML
12. : this category allows you to develop AI/ML agents.
13. You can choose to:
14.
    - Create an agent requesting data from an AI model.
15. Smart Services
16. : this category is for building smart agents for different services.agent
17. You can choose among:
18.
    - Weather data agent
19.
    - : this allows you to develop agents retrieving weather data from weather services API.
20.
    - Weather data provider and consumer agents
21.
    - : this use case allows you to create two different agents, one providing weather data and another one consuming such data.
22.
    - Weather data seller and buyer agents
23.
    - : this use case allows you to create two different agents, one selling weather data and another one buying such data.
24.
    - Weather oracle provider and consumer agents
25.
    - : this use case allows you to create two different agents, one updating an oracle contract with weather data and another one requesting such data to the oracle by paying a small fee.
26.
    - MongoDB Atlas integration
27.
    - : this use case shows how to use MongoDB Atlas as a backend for agent storage with a restaurant booking example.
28.
    - PlanetScale integration
29.
    - : this use case shows how to use PlanetScale as a backend for agent storage with a restaurant booking example.
30. DeltaV Integration
31. : this category allows you to develop Agents offering different Agent Functions (i.e., Services) which are then made available through theDeltaV ↗
32. chat based on a given template. These agents and related Functions need to register within theAgentverse ↗
33. first so for them to be available for queries from users operating and interacting DeltaV Agent. Checkout our dedicated guide for Agent Functions availablehere ↗
34. .
35. You can choose to:
36.
    - Stock Price Agent
37.
    - : create a DeltaV compatible agent requesting stock price of a share.
38.
    - Coin Toss Agent

39.
    - : create a DeltaV compatible agent tossing a coin for you.
40.
    - Dice Roll Agent
41.
    - : create a DeltaV compatible agent rolling a dice for you.
42.
    - EV Charger Agent
43.
    - : create a DeltaV compatible agent to find nearby EV charging stations via DeltaV chat.
44.
    - Geocode Agent
45.
    - : create a DeltaV compatible agent to find the latitude and longitude of a given address using DeltaV chat.
46.
    - Restaurant Agent
47.
    - : create a DeltaV compatible agent to find the nearby restaurants based on a location using DeltaV chat.

Let's assume you want to create your agent using theYour first agent template from theGetting Started window. Each template available presents a small number in the bottom right corner indicating the number of agents available for that particular template.

Whenever you choose one of the templates available to start building your agent, in the Agent Editor you will see some additional information appearing when clicking on the use case chosen, alongside with the baseline code needed to make it runnable. You can use this as a starting point for your ideas.

The output in this case would be the following:

Support for Local Agent Code

You can now import theAgent class and create your agent withmy_agent = Agent(...) . Note that this is simply a wrapper for your pre-loadedagent instance, so any custom configuration you add to the agent will be ignored. This is because the configuration, seed, name, and address for hosted agents is already set, and is not allowed to be overwritten.

Additionally, you can now useagent.run() method, but this is also not required. ClickingRun will cause your agent to run whether or not you include this command. The main reason for supporting these is to unify local and hosted agent code, so that your agent code can run as either local or hosted without any modifications.

Exceptions :

- Use ofBureau
- is not allowed as each agent project can contain only a single agent.
- For security reasons, imports are still restricted to those listed inAgentverse: allowed imports ↗
- . Soon we will be making available a new hosting environment which much fewer restrictions on the imports.
- **i**
- Checkout theAVCTL - Agentverse Command Line Interface ↗
- to better understand how to possibly interact with the Agentverse. AVCTL provides multiple features (e.g., authorization and hosting management) allowing developers to log in, set up projects, and deploy Agents on the Agentverse.

Register Domain Name

You can also provide aWeb3 Agent Name to your agent. To do so, just click on+ Register Web3 Name and fill in the dedicated field for this. This way, your agent will be given a unique name, making it easier for other agents to find and communicate with it. The only requirement isthe name being in lowercase with no whitespaces in between .

Was this page helpful?

Verify messages with Agents Agentverse: Dice Roll agent