Agents

Agents are entities designed to assist users in interacting with Smart Contracts by managing the proof verification of verifiable ML models and executing these contracts using Ape's framework.

Agents serve as intermediaries between users and Smart Contracts, facilitating seamless interaction with verifiable ML models and executing associated contracts. They handle the verification of proofs, ensuring the integrity and authenticity of data used in contract execution.

Creating an agent

To create an agent, first you need to have an endpoint already deployed and anape account created locally. If you have not yet deployed an endpoint, please refer to theendpoints documentation. To create the ape account, you can use theape accounts generate command:

```
Copy ape accounts generate Enhance the security of your account by adding additional random input: Show mnemonic? [Y/n]: n Create Passphrase to encrypt account: Repeat for confirmation: SUCCESS: A new account '0x766867bB2E3E1A6E6245F4930b47E9aF54cEba0C' with HDPath m/44'/60'/0'/0/0 has been added with the id ''
```

The passphrase must be kept secret and secure, as it is used to encrypt the account and is required to access it. The account name is used to identify the account and along with the passphrase to perform transactions in the smart contracts. To create an agent, users can employ thecreate command. This command facilitates the creation of an agent, allowing users to interact with deployed endpoints and execute associated contracts.

During the creation you will be asked to select an account to create the agent. The account is used to sign the transactions in the smart contracts.

```
Copy

    giza agents create --model-id--version-id --name --description

[giza][2024-04-10 11:50:24.005] Creating agent ✅ [giza][2024-04-10 11:50:24.006] Using endpoint id to create agent, retrieving model id and version id [giza][2024-04-10 11:50:53.480] Select an existing account to create the agent. [giza][2024-04-10 11:50:53.480] Available accounts are: ┏━━━━━━━━━━━━━━┓ ┃ Accounts ┃ ┡━━━━━━━━━━━━━━┩ │ my_account │ └──────────────┘ Enter the account name: my_account { "id": 1, "name": , "description": , "parameters": { "model_id": , "version_id": , "endpoint_id": , "alias": "my_account" }, "created_date": "2024-04-10T09:51:04.226448", "last_update": "2024-04-10T09:51:04.226448" }
```

An Agent can also be created using the--endpoint-id flag, which allows users to specify the endpoint ID directly.

```
Copy

    giza agents create --endpoint-id--name--description
```

Listing agents

The list command is designed to retrieve information about all existing agents and the parameters of them.

```
Copy

    giza agents list [giza][2024-04-10 12:30:05.038] Listing agents ✅ [ { "id": 1, "name": "Agent one", "description": "Agent to handle liquidity pools", "parameters": { "model_id": 1, "version_id": 1, "endpoint_id": 1, "account": "awesome_account", }, "created_date": "2024-04-09T15:07:14.282177", "last_update": "2024-04-10T10:06:36.928941" }, { "id": 2, "name": "Agent two", "description": "Agent to handle volatility", "parameters": { "model_id": 1, "version_id": 2, "endpoint_id": 2, "account": "another_awesome_account" }, "created_date": "2024-04-10T09:51:04.226448", "last_update": "2024-04-10T10:12:18.975737" } ]
```

`` ``` ``

Retrieving an Agent

For retrieving detailed information about a specific agent, users can utilize theget command. This command allows users view the details of a specific agent:

```
Copy

    giza agents get --agent-id 1 { "id": 1, "name": "Agent one", "description": "Agent to handle liquidity pools", "parameters": { "model_id": 1, "version_id": 1, "endpoint_id": 1, "account": "awesome_account", }, "created_date": "2024-04-09T15:07:14.282177", "last_update": "2024-04-10T10:06:36.928941" }
```

Updating an Agent

To update an agent, users can use theupdate command. This command facilitates the modification of an agent, allowing users to update the agent's name, description, and parameters.

```
Copy

    giza agents update --agent-id 1 --name "Agent one updated" --description "Agent to handle liquidity pools updated" --parameters chain=ethereum:mainnet:geth

{ "id": 1, "name": "Agent one updated", "description": "Agent to handle liquidity pools updated", "parameters": { "model_id": 1, "version_id": 1, "endpoint_id": 1, "chain": "ethereum:mainnet:geth", "account": "awesome_account", }, "created_date": "2024-04-10T09:51:04.226448", "last_update": "2024-04-10T10:37:28.285500" }
```

The parameters can be updated using the--parameters flag, which allows users to specify the parameters to be updated.

```
Copy

    giza agents update --agent-id 1 --parameters chain=ethereum:mainnet:geth --parameters account=awesome_account
```

The --parameters flag can be used multiple times to update multiple parameters and expects a key-value pair separated by an equal sign,parameter_key=parameter_value .

Delete an Agent

For deleting an Agent, users can use thedelete command. This command will erase any related data to the agent.

```
Copy

    giza agents delete --agent-id 1 [giza][2024-04-10 12:40:33.959] Deleting agent 1 ✅ [giza][2024-04-10 12:40:34.078] Agent 1 deleted ✅
```

More Information

For more information about agents, and their usage in AI Actions, please refer to theAgents documentation.

Previous Endpoints Next Cairo

Last updated3 days ago