

Registering in the Almanac Contract

Introduction

The [Almanac ↗](#) contract requires agents to register using their Agent Address and pay a fee for registration to be found by other agents on the network.

Walk-through

1. First of all, create a Python script and name it by running: `touch registration.py`
2. Then, import the `Agent`
3. `class` from `theuagents`
4. `library` to create our agent, and the `fund_agent_if_low`
5. `class` from `theuagents.setup`
6. `module`. This function will check if you have enough tokens to register in the Almanac contract, if not it will add testnet tokens to your `Fetch Network` address
7. `.` Then, create an agent, `alice`
8. `,` providing `name`
9. `,` `port`
10. `,` `seed`
11. `,` `and endpoint`
12. `:`

```
registration.py from uagents . setup import fund_agent_if_low from uagents import Agent , Context , Protocol
```

alice

```
Agent ( name = "alice" , port = 8000 , seed = "alice secret phrase" , endpoint = [ "http://127.0.0.1:8000/submit" ], )
```

```
fund_agent_if_low (alice.wallet. address ())
```

```
@alice . on_interval (period = 3 ) async
```


```
def
```

```
hi ( ctx : Context): ctx . logger . info ( f "Hello" )
```

`alice . run ()` There's a few things happening in this script; we initialize Alice with an endpoint. An endpoint is the address in which other agents can send messages to where Alice will be listening. As highlighted in [Registration and Endpoints Weighting ↗](#), agents can communicate by querying the Almanac and retrieving an endpoint from the recipient agent. Therefore, we need to specify the service endpoints when defining our agents.

We also have to define `alice.run()`. This `run()` function runs the agent, but more importantly this registers the agent to the Almanac when code is initialized.

Once you run your script, your agent will start the registration process automatically, the balance of the agent's wallet will be checked and funded if needed. Finally, it will try to register on the Almanac contract. Then, we will be ready to start to a remote communication with other agents registered within the Almanac contract.

i For further information on how to set up your uAgents and register them into the Almanac to allow for remote communication, visit the [Communicating with other uAgents](#) guide. 

Was this page helpful?

[Using agents storage function Communicating with other agents](#)