Getting Started with FET Token for Agent development

This is currently a work in progress and may be subject to further updates and revisions.

Introduction

The FET token was created to facilitate payments in an AI driven world. Where Agents send micro payments to one another which traditional currencies do not support. The FET token is the currency of theArtificial Superintelligence (ASI) Alliance , unitingFetch.ai ,Ocean Protocol , andSingularityNET . FET token enabled decentralized AI services, governance, and computational resources.

Whether enabling microtransactions for autonomousAgents , supporting network operations through staking, or unlocking access to AI tools, the FET token plays a central role into advancing the infrastructure of decentralized AI functionalities. Alrighty, let's get started with the FET token.

Understanding the FET Token

The FET token is central to enabling the agentic-driven future on the Fetch.ai mainnet and within the broader ASI ecosystem. Its key utilities include:

- Network Operations
- : FET tokens secure the Fetch.ai mainnet through staking and enable decentralized governance.
- Agent-Based Systems
- : FET is used to register, and interact with autonomous AI agents in Fetch.ai's ecosystem.
- Decentralized Transactions
- : FET enables low-cost micropayments and smart contract interactions.
- AI Marketplace Access
- : Use FET for accessing AI tools, datasets, and compute resources.

Acquiring FET Tokens

Exchanges are the simplest way to get FET tokens.

Visit a trusted exchange such as Coinbase or Binance. You may need to create an account, which will involve you passingKYC(opens in a new tab) . Once your account is enabled, you can buy FET token. You can buy this with most global currencies, just check on the exchange which currencies they accept.

Storing FET

Ideally you would be sending this FET to your agent to transact on the network. However, if you'd like greater security or accessibility..

For short-term storage goals, the ASI Wallet is an excellent choice; it provides direct functionalities for daily activities such as staking and interacting with Agents.

For long-term storage goals, you can enhance security by integrating your ASI Wallet with a hardware wallet like Ledger, offering robust protection for your assets. Follow this guidehere if you wish to set up the ASI Alliance extension wallet with a Ledger hardware wallet.

How to transfer tokens to your Agents

You can download and use the uAgents library to create autonomous Agents capable of interacting with other Agents in a decentralized environment. Check out this guidehere to get started with Agents development.

In order to get your Agent up and running within the Fetch.ai ecosystem, you will need to retrieve the Agent's address and fund it with FET tokens to make it correctly register within the network.

ℹ When creating your account, it is crucial to securely store youseed phrase . The seed phrase is essential for accessing your Agent's identity and controlling any funds it holds. Treat it with the highest level of security to prevent unauthorized access!

Getting your Agent address to send tokens to your Agent

The following Python script demonstrates how to create and initialize an Agent using theuagents andcosmpy libraries, to connect it to the Fetch.ai Mainnet, and retrieve its address and balance information:

agent_address_and_balance.py from uagents import Agent , Context import cosmpy

from cosmpy . aerial . client import LedgerClient , NetworkConfig

agent

Agent (name = "alice" , seed = "" , port = 8000 , test = False , endpoint = ["http://localhost:8000/submit"])

@agent . on_event ("startup") async

def

introduce_agent (ctx : Context): ctx . logger . info (f "ASI network address: { agent.wallet. address () } ") ledger_client =

LedgerClient (NetworkConfig. fetch_mainnet ()) address :

str

= agent . wallet . address () balances = ledger_client . query_bank_all_balances (address) ctx . logger . info (f "Balance of addr: { balances } ")

if

name

==

"**main**" : agent . run () You must update the seed value, and store it safely. Losing this value will lose you your tokens.

In the code example above an Agent namedalice is initialized with a specifiedname ,port ,endpoint , andseed parameters. When the Agent starts up, it logs the wallet address and queries the balance using theLedgerClient connected to the Fetch.ai Mainnet. Finally, the script runs the Agent, which processes thestartup event and retrieves the balance, allowing the Agent to interact with the Fetch.ai network.

Once you run the above Agent script, you will see your Agent address and balance printed out. You will see something similar to the following output:

INFO: [alice]: Registration on Almanac API successful INFO: [alice]: Registering on almanac contract... INFO: [alice]: Registering on almanac contract...complete INFO: [alice]: Agent inspector available at https://agentverse.ai/inspect/?uri=http%3A//127.0.0.1%3A8000&address=agent1qdxdrwqek4pt9xt8kggcxus0zm54d4vgdznrs6y5acn26paphervwfj7pdd INFO: [alice]: Starting server on http://0.0.0.0:8000 (Press CTRL+C to quit) INFO: [alice]: ASI network address:fetch1ujr7wyuvza7uwnkr3usv53hjlwjvu8s7l06vzf INFO: [alice]: Balance of addr: [] You can now use this address to transfer your purchased FET tokens from the exchange to this Agent's address. This should be as simple as withdrawing native FET, by selecting the Fetch.ai Mainnet network when withdrawing. Some exchanges do not support Native FET, and you will need to use thetoken bridge(opens in a new tab) , luckily we havea guide for that too

Check out the following resources for more information on Agents and how these integrate within the Fetch Ecosystem and perform operations using FET tokens:

- Agents - uAgents Framework
- The Agentverse
- AI Engine
- DeltaV

Last updated on January 20, 2025

Was this page helpful?

You can also leave detailed feedbackon Github

Secret Management APIs How to convert native FET to and from ERC-20 FET

On This Page