

# Agent Envelopes in the Fetch.ai uAgents Framework

## Introduction

Within the Fetch.ai uAgents Framework, effective communication between [Agents](#) is crucial for the execution of complex tasks and exchange of information. To this aim, the `Exchange` protocol is a standardized communication protocol within the Framework that facilitates such interactions. A key component of this protocol are `Envelopes`; these serve as structured containers for messages exchanged between agents.

In particular, envelopes provide a secure and reliable way to exchange messages by encapsulating them along with essential metadata. This metadata includes details about the sender and recipient, the session in which the communication occurs, and other information needed to ensure the integrity and authenticity of the messages. By using envelopes, we ensure that messages are correctly routed, verified, and processed, thereby enabling straightforward interoperability among multiple agents and related functions.

i Check out the [Exchange protocol](#) resource for a better comprehension on this topic.

## What Are Envelopes?

Envelopes are specialized structures used to encapsulate messages in the `Exchange` protocol. The structure of an envelope can be likened to a physical envelope containing a letter, where the outer envelope holds information such as the sender's and recipient's addresses, while the inner letter (i.e., the payload) carries the actual content.

They play a critical role in the communication process by:

1. Packaging the message
2. : Envelopes contain the message payload, ensuring that all necessary information is transmitted together.
3. Providing metadata
4. : Envelopes include metadata such as sender and recipient addresses, session information, and expiration times. This metadata helps in identifying and routing the message correctly.
5. Ensuring security and integrity
6. : Envelopes can include digital signatures and other security features to authenticate the sender and ensure the message has not been tampered with.

## Structure of an Envelope

The `Envelope` class can be defined as follows:

```
@dataclass class
```

```
Envelope ( BaseModel ): version :
```

```
int
```

## Envelope version

```
sender :
```

```
str :
```

## Bech32-encoded public address of the sender

```
target :
```

```
str :
```

## Bech32-encoded public address of the target recipient

```
session : UUID4
```

## UUID of the session

schema\_digest :

str

## Digest of the schema used for the payload

protocol\_digest : Optional [ str ]

## Optional protocol digest for custom protocols

payload : Optional [ str ]

## JSON payload encoded as a base64 string

expires : Optional [ int ]

## Unix timestamp in seconds indicating expiration

nonce : Optional [ int ]

## Optional nonce

signature : Optional [ str ]

## Bech32-encoded signature

**i** Envelopes are quite similar to blockchain transactions .

## Sending Envelopes

Envelopes are encoded in JSON format and sent via HTTP 1.1 POST requests to specific endpoints of receiving agents or services. Currently, the protocol supports only one standardized endpoint:

HTTP 1.1 POST /submit

This endpoint expects data that adheres to a JSON-compatible format, typically using the MIME content type `application/json` .

In essence, Envelopes in the uAgents Framework act as secure wrappers for messages, providing essential information for their secure and reliable transmission between Agents within the Framework itself.

Last updated on August 12, 2024

### Was this page helpful?

You can also leave detailed feedback [on Github](#)

[Agents Asynchronous Loops](#) [Verify messages with Agents](#)

On This Page

- [Introduction](#)
- [What Are Envelopes?](#)
- [Structure of an Envelope](#)
- [Sending Envelopes](#)
- [Edit this page on github \(opens in a new tab\)](#)