
AmongUsIO

Technofab

Apr 05, 2021

CONTENTS:

1	Introduction	3
1.1	Prerequisites	3
1.2	Installing	3
2	API Reference	5
2.1	Client	5
2.2	Exceptions	7
3	Indices and tables	9
	Index	11



INTRODUCTION

1.1 Prerequisites

You will need Python 3.6 or higher because this uses [f-Strings](#) (3.6+)

1.2 Installing

1.2.1 PyPi

```
pip install amongus  
# or  
python -m pip install amongus
```

1.2.2 Test-PyPi (automatic dev build)

```
pip install --index-url https://test.pypi.org/simple/ amongus  
# or  
python -m pip install --index-url https://test.pypi.org/simple/ amongus
```

1.2.3 Development Version

```
git clone https://gitlab.com/TECHNOFAB/AmongUsIO  
cd amongusio  
pip install .
```


API REFERENCE

2.1 Client

```
class amongus.Client (name:      str,      color:      amongus.enums.PlayerAttributes.Color  
                    = 0,      hat:      amongus.enums.PlayerAttributes.Hat    = 0,  
                    skin:      amongus.enums.PlayerAttributes.Skin    = 0,      pet:  
                    amongus.enums.PlayerAttributes.Pet = 0, spectator: bool = False)
```

The main client used to interact with the Among Us servers

name

The current name of the user

Type str

stopped

If the client is stopped (= connection closed)

Type bool

lobby_code

The current game lobby code (normally 6 chars long)

Type str

region

The current region to which the client is connected to

Type str

color

Color of the character

Type PlayerAttributes.Color

hat

Hat of the character

Type PlayerAttributes.Hat

skin

Skin of the character

Type PlayerAttributes.Skin

pet

Pet of the character

Type PlayerAttributes.Pet

spectator

Whether the Client should behave like a normal player or just “spectate” and remain invisible

Type bool

game

The current game

Type Game

players

Players in the current game

Type List[Player]

run (*args, **kwargs) → Any

Helper function which runs `Client.start()`

All arguments will be passed to `Client.start()`, this will block until the connection is closed from either side

add_listener (event: str, func: Callable) → None

This adds a listener to the eventbus

Parameters

- **event** (str) – The event to listen/subscribe to
- **func** (Callable) – The callback which will be run when the event happens

Raises **TypeError** – The callback is not a coroutine

remove_listener (func: Callable) → None

Removes an event listener, doesn’t do anything when it doesn’t exist

Parameters **func** (Callable) – The callback which should be removed as a listener

event (name: Optional[Union[str, Callable]] = None) → Callable

Decorator for `Client.add_listener()`

Parameters **name** (str) – Optional; The event name to listen on, if not given the function name will be used

await start (region: Optional[str] = None, custom_server: Optional[str] = None, port: int = 22023, gameVersion: tuple = (2021, 3, 5)) → Any

Starts the client, connecting to the server and sleeping until disconnect

Parameters

- **region** (str) – Optional; The region where the lobby is hosted, see `amongus.regions.regions`
- **custom_server** (str) – Optional; A custom address to connect to, either this or region has to be given. Example: `10.1.1.1:22023` or `10.1.1.1`
- **port** (int) – Optional; Port of the server to connect to, defaults to 22023
- **gameVersion** (tuple) – Optional; The version of the game running on the server

Raises

- **ConnectionException** – Server disconnected, see `ConnectionException.reason` for the reason and `ConnectionException.custom_reason` if the reason is “Custom”
- **AmongUsException** – Invalid region or custom_server could not be parsed

await join_lobby (*lobby_code: str*) → bool

Joins an existing lobby

Parameters **lobby_code** (*str*) – 6 or 4 digit lobby code from Among Us

Raises **AmongUsException** – Invalid lobby code

await find_games (*mapId: amongus.enums.GameSettings.SearchMap = <SearchMap.All: 7>, impostors: int = 0, language: amongus.enums.GameSettings.Keywords = <Keywords.All: 0>*) → amongus.game.GameList

Returns the currently open games/lobbies

Parameters

- **mapId** (*GameSettings.SearchMap*) – The wanted map
- **impostors** (*int*) – Amount of impostors (0-3, 0 being Any)
- **language** (*GameSettings.Keywords*) – Which language the chat should be

Returns GameList

Raises **AmongUsException** – Amount of impostors is not between 1 and 3

await stop (*force: bool = False*) → None

Stops the client

Informs the server before disconnecting if force is False

Parameters **force** (*bool*) – Will just close the connection if True, otherwise it will inform the server first

await send_chat (*message: str*) → None

Sends a chat message to the server

Parameters **message** (*str*) – The message to send

await move (*position: Tuple[int, int], velocity: Tuple[int, int]*) → None

Moves the player to the given position

Parameters

- **position** (*Tuple[int, int]*) – A tuple of x, y coordinates to move to
- **velocity** (*Tuple[int, int]*) – A tuple of x, y coordinates with the velocity/relative position

2.2 Exceptions

class amongus.**AmongUsException**

Base Exception from which all exceptions of this package derive

class amongus.**ConnectionException** (*message: str, reason: int, **kwargs*)

Exception which gets raised when something went wrong with the connection

message

Message which explains more about the error

Type str

reason

A DisconnectReason to be able to programmatically check what went wrong

Type int

custom_reason

If reason is `DisconnectReason.Custom`, this will contain the custom disconnect reason sent by the Among Us server

Type str

INDICES AND TABLES

- `genindex`

INDEX

A

`add_listener()` (*amongus.Client method*), 6
`AmongUsException` (*class in amongus*), 7

C

`Client` (*class in amongus*), 5
`color` (*amongus.Client attribute*), 5
`ConnectionException` (*class in amongus*), 7
`custom_reason` (*amongus.ConnectionException attribute*), 7

E

`event()` (*amongus.Client method*), 6

F

`find_games()` (*amongus.Client method*), 7

G

`game` (*amongus.Client attribute*), 6

H

`hat` (*amongus.Client attribute*), 5

J

`join_lobby()` (*amongus.Client method*), 6

L

`lobby_code` (*amongus.Client attribute*), 5

M

`message` (*amongus.ConnectionException attribute*), 7
`move()` (*amongus.Client method*), 7

N

`name` (*amongus.Client attribute*), 5

P

`pet` (*amongus.Client attribute*), 5
`players` (*amongus.Client attribute*), 6

R

`reason` (*amongus.ConnectionException attribute*), 7
`region` (*amongus.Client attribute*), 5
`remove_listener()` (*amongus.Client method*), 6
`run()` (*amongus.Client method*), 6

S

`send_chat()` (*amongus.Client method*), 7
`skin` (*amongus.Client attribute*), 5
`spectator` (*amongus.Client attribute*), 5
`start()` (*amongus.Client method*), 6
`stop()` (*amongus.Client method*), 7
`stopped` (*amongus.Client attribute*), 5