Howl Documentation

Release 1.0.0

Hawkins

November 06, 2016

Table of Contents

Howl is a python-based bot framework for the web game Initium. Making bots with Howl is easy! Check out the following pages to begin.

CHAPTER 1

General and Module Indices

- genindex
- modindex

Contents:

1.1 Initium Module

Initium.py is essentially the core framework for the Howl project. This file contains all the methods included in the framework, however, modules are encouraged to contain their own functions in addition to the core framework functions found here.

class `initium.`**`initium`**

Bases: `object`

This is the core Howl framework object.

All bot modules should inerhit from this class by using:

```
>>> class Bot(initium.webdriver, initium.initium): # Class definition line for Bot implementing
```

`get_chat_messages`(*chat_tab='Private'*)

This function opens the specified chat tab and collects all messages and authors in the chat.

Args: chat_tab (string) – Global, Location, Party, Group, or Private. Default is "Private"

Returns: messages (array) – An array of dictionaries containing keys 'time', 'author', 'text', 'channel'.

One way to use this function is:

messages = Bot.update_messages("Global")

`get_gold`()

This function returns the current gold in the bot's inventory in the Initium game.

Args: None: This function takes no arguments

Returns: Gold (string) – Either actual gold or "-1" if gold could quantitiy not be found.

One way to use this function is:

```
>>> print("Gold: " + Bot.get_gold())
Gold: 9,873
```

get_item_stats(*element=None*)

This function parses the popup HTML for relevant item stats. Optionally receives an argument *element* to click before loading stats, otherwise assumes an item popup is opened.

Args: element (selenium.WebElement) – Optionally click this element to load its item stats

Returns: stats (dictionary) – A mapping of string keys to their respective stats. Empty if no popup is open

One way to use this function is:

```
>>> item_stats = Bot.get_item_stats(item_web_element) # Fills the dictionary with all releva
```

get_location()

This function returns the current location of the bot in the Initium world.

Args: None: This function takes no arguments

Returns: Location (string) – Either actual location or "Unknown Location" if location could not be found.

One way to use this function is:

```
>>> print("Location: " + Bot.get_location())
Location: Aera
```

is_chat_tab_loaded(*chat_tab='Location'*)

Checks for existance of messages in the given chat tab.

Args: chat_tab (strng) – The chat tab to select. Default is 'Location'

Returns: result (boolean) – True if an item popup is loaded

is_item_popup_open()

Checks for existance of an item popup with stats visible.

Args: None - This function receives no arguments

Returns: result (boolean) – True if an item popup is loaded

login(*email*, *pw*)

This function logs the player in to the Initium world.

Args: email (string) – Username to log in with pw (string) – Password to log in with

Returns: None: This function return is void

reply(*text*, *player=None*, *delay=True*)

This replies to a player with a certain message in the Private chat tab. Multiple messages can be said by separating them with newline characters.

Do be wary of multiple messages at once tho, as too many sent too quickly can result in a temporary chat ban in-game.

Assumes player name already selected if player is not given.

Args: text (string) – Actual message to be said to player.

player (string) – Target player's name.

delay (boolean) – Delay between sending messages or not. Default is True

Returns: Nothing is returned by this function.

Examples of this function in use:

```
>>> Bot.reply("Using Howl is easy!", "[Dev] Rade") # "Using Howl is easy!" whispered to whic
>>> Bot.reply("This Message splits \n Into two chat messages.", "Bella") # "This message spl
>>> Bot.reply("Something here.") # "Something here" said in private chat, assuming player na
```

say (*text*, *chat_tab='Location'*, *delay=True*)

This function says a message in the specified chat tab. Multiple messages can be said by separating them with newline characters.

Do be wary of multiple messages at once tho, as too many sent too quickly can result in a temporary chat ban in-game.

Args: text (string) – Actual message to be said in chat.

chat_tab (string) – Global, Location, Party, Group, or Private (Note, Initium.reply should be used for private). Default is "Location"

delay (boolean) – Delay between sending messages or not. Default is True

Returns: Nothing is returned by this function.

Examples of this function in use:

```
>>> Bot.say("Using Howl is easy!", "Global") # "Using Howl is easy!" in Global tab
>>> Bot.say("This Message splits \n Into two chat messages.", "Location") # "This message sp
```

send_keypresses (*keys*)

Send keys to the page.

Args: keys (string) – String of keys to be pressed

Returns: Nothing is returned by this function

1.2 Owl Module

Owl.py is a prime example of how to use the Howl framework. Owl is a solid starting point for new modules. Simply take the file as a coookie-cutter template for making a new bot with Howl, as at the moment we do not have a traditional example bot.

Because Owl uses no new functions, we have nothing to document. Instead, the logging method will be documented here. Note this logger reappears in records.py as well.

class `owl.`**`Owl_Bot`**

Bases: `selenium.webdriver.chrome.webdriver.WebDriver`, *`initium.initium`*

`owl.`**`create_timed_rotating_log`**(*path*)

1.3 Records Module

Records.py is another quick example of a Howl bot. This bot is designed to track the best items seen in the game, and re-share them on request.

class `records.`**`Records`**

Bases: `selenium.webdriver.chrome.webdriver.WebDriver`, *`initium.initium`*

calculate_score_armor (*chance=0*, *reduction=0*, *penalty=0*, *shirt=False*)

calculate_score_weapon (*dice=0*, *sides=0*, *chance=0*, *mult=0*)

load_records (*file_path*)

remove_record (*file_path*, *rank*)

save_records (*item_name*, *item_id*, *item_score*)

score_item()

records.**create_timed_rotating_log**(*path*)

Index