SpacePyTraders

Release 0.0.5

Zac Hooper

Jun 23, 2021

CONTENTS:

CHAPTER

ONE

CLIENT MODULE

The client is what provides easy pythonic interaction with the Space Traders API. Interact with the API with better human readable code rather than convoluted requests.

class SpacePyTraders.client.**Marketplace**(*username*, *token=None*)

THIS CLASS IS BEING DEPRECATED All containing methods have been removed or are being deprecated.

get_marketplace(*symbol*, *raw_res=False*, *throttle_time=10*)

THIS METHOD IS BEING DEPRECATED This method has moved to the Location class. Please now use *locations.get_marketplace()*.

Get the marketplace for the location provided

Parameters **symbol** (*str*) – The symbol for the location eg: OE-PM

Returns A dict containing details of the location and a JSON list of the items available in the marketplace

Return type dict

exception SpacePyTraders.client.**ServerException**(*data: Field(name=None, type=None, default=<dataclasses._MISSING_TYPE object at 0x7f61b9f6ce90>, default_factory=<class 'dict'>, init=True, repr=True, hash=None, compare=True, metadata=mappingproxy({}), _field_type=None)*, *message: str = 'Server Error. Pausing before trying again'*)

exception SpacePyTraders.client.**ThrottleException**(*data: Field(name=None, type=None, default=<dataclasses._MISSING_TYPE object at 0x7f61b9f6ce90>, default_factory=<class 'dict'>, init=True, repr=True, hash=None, compare=True, metadata=mappingproxy({}), _field_type=None)*, *message: str = 'Throttle limit was reached. Pausing to wait for throttle'*)

exception SpacePyTraders.client.**TooManyTriesException**(*message: str = 'Has failed too many times to make API call.'*)

class SpacePyTraders.client.**Users**(*username*, *token=None*)
THIS CLASS IS BEING DEPRECATED All methods have been removed or are being deprecated.

get_your_info(*raw_res=False*, *throttle_time=10*)
THIS METHOD IS BEING DEPRECATED

This method has now moved to the Account class. Please now use *account.info()*

Get your user info

Returns dict containing your user data

Return type dict

SpacePyTraders.client.**make_request**(*method*, *url*, *headers*, *params*)
Checks which method to use and then makes the actual request to Space Traders API

Parameters

- **method** (*str*) – The HTTP method to use
- **url** (*str*) – The URL of the request
- **headers** (*dict*) – the request headers holding the Auth
- **params** (*dict*) – parameters of the request

Returns Returns the request

Return type Request

Exceptions: Exception: Invalid method - must be GET, POST, PUT or DELETE

1.1 Api

class SpacePyTraders.client.**Api**(*username*, *token=None*)

__init__(*username*, *token=None*)
Initialize self. See help(type(self)) for accurate signature.

generate_token()
Trys to create a new user and return their token

Parameters **username** (*str*) – Username to user

Returns Token if user valid else None

Return type str

1.2 Client

class SpacePyTraders.client.**Client**(*username*, *token=None*)

generic_api_call(*method*, *endpoint*, *params=None*, *token=None*, *warning_log=None*, *raw_res=False*, *throttle_time=10*)
Function to make consolidate parameters to make an API call to the Space Traders API. Handles any throttling or error returned by the Space Traders API.

Parameters

- **method** (*str*) – The HTTP method to use. GET, POST, PUT or DELETE
- **endpoint** (*str*) – The API endpoint
- **params** (*dict, optional*) – Any params required for the endpoint. Defaults to None.
- **token** (*str, optional*) – The token of the user. Defaults to None.
- **raw_res** (*bool, default = False*) – Returns the request response's JSON by default. Can be set to True to return the request response.
- **throttle_time** (*int, default = 10*) – Sets how long the wait time before attempting call again. Default is 10 seconds

Returns depends on the return from the API but likely JSON

Return type Any

1.3 Account

class SpacePyTraders.client.**Account**(*username*, *token=None*)

info(*raw_res=False*, *throttle_time=10*)
Get's the user's info

Parameters

- **raw_res** (*bool, optional*) – Get the actual response from requests. Defaults to False.
- **throttle_time** (*int, optional*) – change the throttle time to wait. Defaults to 10.

Returns Returns a dictionary containing the user's details

Return type dict

1.4 Flight Plans

class SpacePyTraders.client.**FlightPlans**(*username*, *token=None*)

get_active_flight_plans(*symbol*, *raw_res=False*, *throttle_time=10*)
THIS METHOD IS BEING DEPRECATED This method has been moved to the systems class. Please now use *systems.get_active_flight_plans()* to get the active flight plans in the system given.

Get all the currently active flight plans in the system given. This is for all global accounts

Parameters **symbol** (*str*) – Symbol of the system. OE or XV

Returns dict containing a list of flight plans for each system as the key

Return type dict

get_flight_plan(*flightPlanId*, *raw_res=False*, *throttle_time=10*)
Get the details of a currently active flight plan

Parameters **flightPlanId** (*str*) – ID of the flight plan

Returns dict containing the details of the flight plan

Return type dict

new_flight_plan(*shipId*, *destination*, *raw_res=False*, *throttle_time=10*)
Submit a new flight plan for a ship

Parameters

- **shipId** (*str*) – ID of the ship to fly
- **destination** (*str*) – Symbol of the locatino to fly the ship to

1.5 Game

class SpacePyTraders.client.**Game**(*username*, *token=None*)

get_game_status(*raw_res=False*, *throttle_time=10*)
Check to see if game is up

1.6 Loans

class SpacePyTraders.client.**Loans**(*username*, *token=None*)

get_loans_available(*raw_res=False*, *throttle_time=10*)
THIS METHOD IS BEING DEPRECATED This method has now been moved to the types class. Please now use *types.get_loans_available()*.

Gets the list of loans available

Returns dict containing a list of loans

Return type dict

get_user_loans(*raw_res=False*, *throttle_time=10*)
Gets the list of loans available

Returns dict containing a list of loans

Return type dict

pay_off_loan(*loanId*, *raw_res=False*, *throttle_time=10*)
Pays of the loan with ID provided

Parameters **loanId** (*str*) – ID of the loan to pay off

Returns Success or fail message

Return type dict

request_loan(*type*, *raw_res=False*, *throttle_time=10*)
Request a new loan

Parameters **type** (*str*) – The type of loan - e.g. STARTUP

Returns The loan taken

Return type dict

1.7 Locations

class SpacePyTraders.client.**Locations**(*username*, *token=None*)

get_location(*symbol*, *raw_res=False*, *throttle_time=10*)
Get info on a location with the provided Symbol

Parameters **symbol** (*str*) – The symbol for the location eg: OE-PM

Returns A dict containing info about a location

Return type dict

get_marketplace(*symbol*, *raw_res=False*, *throttle_time=10*)
Get the marketplace for the location provided

Parameters **symbol** (*str*) – The symbol for the location eg: OE-PM

Returns A dict containing details of the location and a JSON list of the items available in the marketplace

Return type dict

get_ships_at_location(*symbol*, *raw_res=False*, *throttle_time=10*)
Get the ships docked at a location

Parameters **symbol** (*str*) – The symbol for the location eg: OE-PM

Returns A dict containing a JSON list of the ships docked at the location.

Return type dict

get_system_locations(*symbol*, *type=None*, *raw_res=False*, *throttle_time=10*)
THIS METHOD IS BEING DEPRECATED This method has now moved to the systems class. Please now use *systems.get_system_locations()*.

Get locations in the defined system

Parameters **symbol** (*str*) – The symbol for the system eg: OE

Returns A dict containing a JSON list of the locations in the system

Return type dict

1.8 Marketplace

class SpacePyTraders.client.**Marketplace**(*username*, *token=None*)
THIS CLASS IS BEING DEPRECATED All containing methods have been removed or are being deprecated.

get_marketplace(*symbol*, *raw_res=False*, *throttle_time=10*)
THIS METHOD IS BEING DEPRECATED This method has moved to the Location class. Please now use *locations.get_marketplace()*.

Get the marketplace for the location provided

Parameters **symbol** (*str*) – The symbol for the location eg: OE-PM

Returns A dict containing details of the location and a JSON list of the items available in the marketplace

Return type dict

1.9 PurchaseOrders

class SpacePyTraders.client.**PurchaseOrders**(*username*, *token=None*)

new_purchase_order(*shipId*, *good*, *quantity*, *raw_res=False*, *throttle_time=10*)
Makes a purchase order to the location the ship is currently located at.

Parameters

- **shipId** (*str*) – ID of the ship to load the goods onto
- **good** (*str*) – Symbol of the good to purchase
- **quantity** (*int*) – How many units of the good to purchase
- **raw_res** (*bool, default = False*) – Returns the request response's JSON by default. Can be set to True to return the request response.
- **throttle_time** (*int, default = 10*) – Sets how long the wait time before attempting call again. Default is 10 seconds

Returns A dict containing the user's remaining credits, the ships updated cargo and the order just made.

Return type dict

1.10 SellOrders

class SpacePyTraders.client.**SellOrders**(*username*, *token=None*)

new_sell_order(*shipId*, *good*, *quantity*, *raw_res=False*, *throttle_time=10*)
Makes a sell order to the location the ship is currently located at.

Parameters

- **shipId** (*str*) – ID of the ship to offload the goods from
- **good** (*str*) – Symbol of the good to sell
- **quantity** (*int*) – How many units of the good to sell

1.11 Ships

class SpacePyTraders.client.**Ships**(*username*, *token=None*)

buy_ship(*location*, *type*, *raw_res=False*, *throttle_time=10*)
Buys a ship of the type provided and at the location provided. Certain ships can only be bought from specific locations. Use get_available_ships to see full list.

Parameters

- **location** (*str*) – symbol of the location the ship to buy is
- **type** (*str*) – type of ship you want to buy e.g. GR-MK-III

get_available_ships(*type=None*, *raw_res=False*, *throttle_time=10*)
THIS METHOD IS BEING DEPRECATED This method has now moved to the types class. Please now use *types.ships()*.

Get the avialable ships to purchase across all systems

Parameters **type** (*str, optional*) – Filter the list of ships to the class level. eg 'MK-II' (Note: those are capital i's). Defaults to None.

Returns A dict containing a JSON list of ships that are available.

Return type dict

API LINK: https://api.spacetraders.io/#api-ships-ships

get_ship(*shipId*, *raw_res=False*, *throttle_time=10*)
Get info on the ship

Parameters **shipId** (*str*) – The shipId of the ship you want to get info on

Returns A dict containing the info about the ship

Return type dict

API LINK: https://api.spacetraders.io/#api-ships-GetShip

get_user_ships(*raw_res=False*, *throttle_time=10*)
Get a list of all the ships you own

Returns A JSON list of the ships you own. Each item is a return from the get_ship_info endpoint.

Return type dict

API Link: https://api.spacetraders.io/#api-ships-GetShips

jettinson_cargo(*shipId*, *good*, *quantity*, *raw_res=False*, *throttle_time=10*)
Jettison (delete) some cargo from a ship

Parameters

- **shipId** (*str*) – The shipId of the ship you want to jettison cargo from
- **good** (*str*) – The symbol of the good you want to jettison. Eg. FUEL
- **quantity** (*int*) – How many units of the good you want to jettison

Returns If successful a dict is returned with the remaining quantitiy of the good on the ship

Return type dict

API Link: https://api.spacetraders.io/#api-ships-JettisonCargo

scrap_ship(*shipId*, *raw_res=False*, *throttle_time=10*)
Scraps the shipId for a small amount of credits. Ships need to be scraped at a location with a Shipyard. Known Shipyards: - OE-PM-TR

Parameters **shipId** (*str*) – ID of the ship to scrap

Returns True if the ship was scrapped

Return type bool

Raises **Exception** – If something went wrong during the scrapping process

transfer_cargo(*fromShipId*, *toShipId*, *good*, *quantity*, *raw_res=False*, *throttle_time=10*)
Move cargo from own ship to another that are in the same location

Parameters

- **fromShipId** (*str*) – The shipId of the ship you want to transfer the cargo FROM
- **toShipId** (*str*) – The shipId of the ship you want to transfer the cargo TO
- **good** (*str*) – The symbol of the good you want to transfer. Eg. FUEL
- **quantity** (*int*) – How many units of the good you want to transfer

Returns A dict is returned with two keys "fromShip" & "toShip" each with the updated ship info for the respective ships

Return type dict

API Link: https://api.spacetraders.io/#api-ships-TransferCargo

1.12 Structures

class SpacePyTraders.client.**Structures**(*username*, *token=None*)

create_new_structure(*location*, *type*, *raw_res=False*, *throttle_time=10*)
Create a new structure on the location provided. Note that only certain structures can be built at specific locations

Parameters

- **location** (*str*) – symbol of the location to build the structure
- **type** (*str*) – type of structure you want to build

deposit_goods(*structureId*, *shipId*, *good*, *quantity*, *user_owned=True*, *raw_res=False*, *throttle_time=10*)
Deposit goods from a ship to a structure. The ship must be at the location the structure has been built.

Parameters

- **structureId** (*str*) – ID of the structure to deposit the goods into
- **shipId** (*str*) – ID of the ship to take the goods from
- **good** (*str*) – symbol of the good to deposite. Eg: FUEL
- **quantity** (*str*) – How many units of the good to deposit
- **user_owned** (*bool*) – Determines which endpoint to use: deposit to user structure or any structure

Returns dict containing the updated info of the ship and structure

Return type dict

Possible Endpoints:

- https://api.spacetraders.io/#api-structures-DepositMyGoods
- https://api.spacetraders.io/#api-structures-DepositStructure

get_structure(*structureId*, *user_owned=True*, *raw_res=False*, *throttle_time=10*)
Get the info about a structure.

the *user_owned* argument will determine whether the *my* endpoint is used or note. Possible Endpoints:

- https://api.spacetraders.io/#api-structures-GetStructure
- https://api.spacetraders.io/#api-structures-GetMyStructure

Parameters

- **structureId** (*str*) – ID of the structure to deposit the goods into
- **user_owned** (*bool*) – Determines if the queried for structure is user owned or not

Returns dict containing the info of the strucutre

Return type dict

get_users_structures(*raw_res=False*, *throttle_time=10*)
Get the info about a structure

Returns dict containings a JSON list of the structures the user owns

Return type dict

transfer_goods(*structureId*, *shipId*, *good*, *quantity*, *raw_res=False*, *throttle_time=10*)
Transfer goods from a structure to a ship. The ship must be docked at the location the structure has been built.

Parameters

- **structureId** (*str*) – ID of the structure to deposit the goods into
- **shipId** (*str*) – ID of the ship to take the goods from
- **good** (*str*) – symbol of the good to deposite. Eg: FUEL
- **quantity** (*str*) – How many units of the good to deposit

Returns dict containing the updated info of the ship and structure

Return type dict

1.13 Systems

class SpacePyTraders.client.**Systems**(*username*, *token=None*)

get_active_flight_plans(*symbol*, *raw_res=False*, *throttle_time=10*)
Get all the currently active flight plans in the system given. This is for all global accounts

Parameters **symbol** (*str*) – Symbol of the system. OE or XV

Returns dict containing a list of flight plans for each system as the key

Return type dict

get_available_ships(*symbol*, *raw_res=False*, *throttle_time=10*)
Get the ships listed for sale in the system defined

Parameters **symbol** (*str*) – The symbol for the system eg: OE

Returns A dict containing a list of the available ships for sale

Return type dict

get_system(*symbol*, *raw_res=False*, *throttle_time=10*)
Get info on the definined system

Parameters **symbol** (*str*) – The symbol for the system eg: OE

Returns A dict with info about the system

Return type dict

get_system_docked_ships(*symbol*, *raw_res=False*, *throttle_time=10*)
Get docked ships in the defined system

Parameters **symbol** (*str*) – The symbol for the system eg: OE

Returns A dict containing a JSON list of the docked ships in the system

Return type dict

get_system_locations(*symbol*, *raw_res=False*, *throttle_time=10*)
Get locations in the defined system

Parameters **symbol** (*str*) – The symbol for the system eg: OE

Returns A dict containing a JSON list of the locations in the system

Return type dict

get_systems(*raw_res=False*, *throttle_time=10*)
[ENDPOINT CURRENTLY BROKEN - DEVS FIXING]

Get info about the systems and their locations.

Returns dict containing a JSON list of the different systems

Return type dict

1.14 Users

class SpacePyTraders.client.**Users**(*username*, *token=None*)
THIS CLASS IS BEING DEPRECATED All methods have been removed or are being deprecated.

get_your_info(*raw_res=False*, *throttle_time=10*)
THIS METHOD IS BEING DEPRECATED

This method has now moved to the Account class. Please now use *account.info()*

Get your user info

Returns dict containing your user data

Return type dict

1.15 Types

class SpacePyTraders.client.**Types**(*username*, *token=None*)

goods(*raw_res=False*, *throttle_time=10*)
Get's all the available goods in the game

Parameters

- **raw_res** (*bool, optional*) – Returns the actual request response. Defaults to False.

- **throttle_time** (*int, optional*) – Change how long to wait if throttled. Defaults to 10.

Returns A dict containing a list of all the goods in the game

Return type dict

loans(*raw_res=False*, *throttle_time=10*)
Get's all the available loans in the game

Parameters

- **raw_res** (*bool, optional*) – Returns the actual request response. Defaults to False.
- **throttle_time** (*int, optional*) – Change how long to wait if throttled. Defaults to 10.

Returns A dict containing a list of all the loans in the game

Return type dict

ships(*raw_res=False*, *throttle_time=10*)
Get's all the available ships in the game

Parameters

- **raw_res** (*bool, optional*) – Returns the actual request response. Defaults to False.
- **throttle_time** (*int, optional*) – Change how long to wait if throttled. Defaults to 10.

Returns A dict containing a list of all the ships in the game

Return type dict

structures(*raw_res=False*, *throttle_time=10*)
Get's all the available structures in the game

Parameters

- **raw_res** (*bool, optional*) – Returns the actual request response. Defaults to False.
- **throttle_time** (*int, optional*) – Change how long to wait if throttled. Defaults to 10.

Returns A dict containing a list of all the structures in the game

Return type dict

1.16 Warp Jumps

class SpacePyTraders.client.**WarpJump**(*username*, *token=None*)

attempt_jump(*shipId*, *raw_res=False*, *throttle_time=10*)
Attempts sending a ship through a warp jump

Parameters

- **raw_res** (*bool, optional*) – Returns the actual request response. Defaults to False.
- **throttle_time** (*int, optional*) – Change how long to wait if throttled. Defaults to 10.

Returns A dict containing a list of all the goods in the game

Return type dict

CHAPTER

TWO

MODELS MODULE

Models provides common objects in the Space Trader Universe. Access a ships speed with dot notation rather than convoluted JSON manipulation.

test

```
from models import Ship
...
ship = Ship(api.ships.get_ship('12345'))
print(ship.manufacturer)
>>> Jackshaw
```

class SpacePyTraders.models.**Cargo**(*good: str*, *quantity: int*, *totalVolume: int*)

class SpacePyTraders.models.**Good**(*symbol: str*, *volumePerUnit: int*, *pricePerUnit: int*, *spread: int*, *purchasePricePerUnit: int*, *sellPricePerUnit: int*, *quantityAvailable: int*)

class SpacePyTraders.models.**Loan**(*id: str*, *due: str*, *repaymentAmount: int*, *status: str*, *type: str*)

class SpacePyTraders.models.**Location**(*symbol: str*, *type: str*, *name: str*, *x: int*, *y: int*, *allowsConstruction: bool*, *structures: Field(name=None, type=None, default=<dataclasses._MISSING_TYPE object at 0x7f61b9f6ce90>, default_factory=<class 'list'>, init=True, repr=True, hash=None, compare=True, metadata=mappingproxy({}), _field_type=None)*, *messages: list = None*)

class SpacePyTraders.models.**Marketplace**(*symbol: str*, *type: str*, *name: str*, *x: int*, *y: int*, *allowsConstruction: bool*, *structures: Field(name=None, type=None, default=<dataclasses._MISSING_TYPE object at 0x7f61b9f6ce90>, default_factory=<class 'list'>, init=True, repr=True, hash=None, compare=True, metadata=mappingproxy({}), _field_type=None)*, *messages: list = None*, *marketplace: list = <factory>*)

get_good(*symbol*)

Returns a Good object for the symbol provided

Parameters **symbol** (*str*) – Symbol of the good Eg: "FUEL"

Returns Good object for the symbol given

Return type *Good*

class SpacePyTraders.models.**Ship**(*id: str*, *manufacturer: str*, *kind: str*, *type: str*, *location: str*, *speed: int*, *plating: int*, *weapons: int*, *maxCargo: int*, *spaceAvailable: int*, *cargo: Field(name=None, type=None, default=<dataclasses._MISSING_TYPE object at 0x7f61b9f6ce90>, default_factory=<class 'list'>, init=True, repr=True, hash=None, compare=True, metadata=mappingproxy({}), _field_type=None)*, *flightPlanId: str = None*, *x: int = None*, *y: int = None*)

class SpacePyTraders.models.**System**(*locations: Field(name=None, type=None, default=<dataclasses._MISSING_TYPE object at 0x7f61b9f6ce90>, default_factory=<class 'list'>, init=True, repr=True, hash=None, compare=True, metadata=mappingproxy({}), _field_type=None)*)

get_location(*symbol*)

Returns a Location object for the symbol provided

Parameters **symbol** (*str*) – Symbol of the location Eg: "OE-PM"

Returns Location object for the symbol given

Return type *Location*

class SpacePyTraders.models.**User**(*username: str*, *credits: int*, *ships: Field(name=None, type=None, default=<dataclasses._MISSING_TYPE object at 0x7f61b9f6ce90>, default_factory=<class 'list'>, init=True, repr=True, hash=None, compare=True, metadata=mappingproxy({}), _field_type=None)*, *loans: Field(name=None, type=None, default=<dataclasses._MISSING_TYPE object at 0x7f61b9f6ce90>, default_factory=<class 'list'>, init=True, repr=True, hash=None, compare=True, metadata=mappingproxy({}), _field_type=None)*)

The basic user object. Great way to store and access a user's credits, ships and loans.

Parameters

- **username** (*str*) – The username of the user
- **credits** (*int*) – How many credits does the user have
- **ships** (*list*) – A list of the ships the user owns
- **loans** (*list*) – A list of the loans the user has

Returns returns a user object

Return type *User*

SpacePyTraders.models.**build_ship**(*ship_dict*)

Handles the creation of a ship class. The ship dict contains a 'class' key which needs to be changed for the class creation. The ship may also be in transit and that needs to be handled accordingly

Parameters **ship_dict** (*dict*) – the dict version of a ship

Returns A ship object

Return type *Ship*

2.1 User

class SpacePyTraders.models.**User**(*username: str*, *credits: int*, *ships: Field(name=None, type=None, default=<dataclasses._MISSING_TYPE object at 0x7f61b9f6ce90>, default_factory=<class 'list'>, init=True, repr=True, hash=None, compare=True, metadata=mappingproxy({}), _field_type=None)*, *loans: Field(name=None, type=None, default=<dataclasses._MISSING_TYPE object at 0x7f61b9f6ce90>, default_factory=<class 'list'>, init=True, repr=True, hash=None, compare=True, metadata=mappingproxy({}), _field_type=None)*)

The basic user object. Great way to store and access a user's credits, ships and loans.

Parameters

- **username** (*str*) – The username of the user
- **credits** (*int*) – How many credits does the user have
- **ships** (*list*) – A list of the ships the user owns
- **loans** (*list*) – A list of the loans the user has

Returns returns a user object

Return type *User*

2.2 Ship

class SpacePyTraders.models.**Ship**(*id: str*, *manufacturer: str*, *kind: str*, *type: str*, *location: str*, *speed: int*, *plating: int*, *weapons: int*, *maxCargo: int*, *spaceAvailable: int*, *cargo: Field(name=None, type=None, default=<dataclasses._MISSING_TYPE object at 0x7f61b9f6ce90>, default_factory=<class 'list'>, init=True, repr=True, hash=None, compare=True, metadata=mappingproxy({}), _field_type=None)*, *flightPlanId: str = None*, *x: int = None*, *y: int = None*)

2.3 Cargo

class SpacePyTraders.models.**Cargo**(*good: str*, *quantity: int*, *totalVolume: int*)

2.4 Loan

class SpacePyTraders.models.**Loan**(*id: str*, *due: str*, *repaymentAmount: int*, *status: str*, *type: str*)

2.5 Location

class SpacePyTraders.models.**Location**(*symbol: str*, *type: str*, *name: str*, *x: int*, *y: int*, *allowsConstruction: bool*, *structures: Field(name=None*, *type=None*, *default=<dataclasses._MISSING_TYPE object at 0x7f61b9f6ce90>*, *default_factory=<class 'list'>*, *init=True*, *repr=True*, *hash=None*, *compare=True*, *metadata=mappingproxy({})*, *_field_type=None)*, *messages: list = None*)

2.6 Good

class SpacePyTraders.models.**Good**(*symbol: str*, *volumePerUnit: int*, *pricePerUnit: int*, *spread: int*, *purchasePricePerUnit: int*, *sellPricePerUnit: int*, *quantityAvailable: int*)

2.7 System

class SpacePyTraders.models.**System**(*locations: Field(name=None*, *type=None*, *default=<dataclasses._MISSING_TYPE object at 0x7f61b9f6ce90>*, *default_factory=<class 'list'>*, *init=True*, *repr=True*, *hash=None*, *compare=True*, *metadata=mappingproxy({})*, *_field_type=None)*)

get_location(*symbol*)

Returns a Location object for the symbol provided

Parameters **symbol** (*str*) – Symbol of the location Eg: "OE-PM"

Returns Location object for the symbol given

Return type *Location*

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

INDEX

Symbols

A

B

C

D

F

G