

EVE Market viewer

0.01

Generated by Doxygen 1.9.5

1 Homepage	1
1.0.1 Quick Start Notes:	1
1.0.2 Example usage	1
2 # Program usage	3
2.1 Commands	3
3 Namespace Index	5
3.1 Package List	5
4 Class Index	7
4.1 Class List	7
5 Namespace Documentation	9
5.1 eve_market Namespace Reference	9
6 Class Documentation	11
6.1 eve_market.ContractInterface Class Reference	11
6.1.1 Detailed Description	11
6.1.2 Constructor & Destructor Documentation	12
6.1.2.1 ContractInterface()	12
6.1.3 Member Function Documentation	12
6.1.3.1 GetBidStrings()	12
6.1.3.2 getContractItemString()	12
6.1.3.3 HandleContracts()	13
6.1.3.4 HandleMyContracts()	13
6.2 eve_market.MainEsiInterface Class Reference	13
6.2.1 Detailed Description	14
6.2.2 Constructor & Destructor Documentation	15
6.2.2.1 MainEsiInterface()	15
6.2.3 Member Function Documentation	15
6.2.3.1 CheckAuthorization()	15
6.2.3.2 GenerateStateString()	15
6.2.3.3 HandleAuthorize()	16
6.2.3.4 HandleHelp()	16
6.2.3.5 HandleLogout()	16
6.2.3.6 IsIdField()	16
6.2.3.7 StringFromSlice()	17
6.3 eve_market.MarketInterface Class Reference	17
6.3.1 Detailed Description	18
6.3.2 Constructor & Destructor Documentation	18
6.3.2.1 MarketInterface()	19
6.3.3 Member Function Documentation	19
6.3.3.1 HandleAssets()	19

6.3.3.2 HandleDefaults()	19
6.3.3.3 HandleHistory()	20
6.3.3.4 HandleMyOrders()	20
6.3.3.5 HandleOrderHistory()	20
6.3.3.6 HandleOrders()	21
6.3.3.7 HandleTransactions()	21
6.3.3.8 HandleWallet()	21
6.3.3.9 SortItems()	21
6.3.3.10 SortOrders()	22
6.4 eve_market.Printer Class Reference	22
6.4.1 Detailed Description	23
6.4.2 Constructor & Destructor Documentation	23
6.4.2.1 Printer()	23
6.4.3 Member Function Documentation	23
6.4.3.1 PrintContracts()	23
6.4.3.2 PrintLine() [1/2]	24
6.4.3.3 PrintLine() [2/2]	24
6.4.3.4 PrintObjList< T >()	24
6.4.3.5 PrintTableHeader()	25
6.5 eve_market.Program Class Reference	25
6.5.1 Detailed Description	25
6.5.2 Member Function Documentation	25
6.5.2.1 ParseInput()	26
6.6 eve_market.UniverseInterface Class Reference	26
6.6.1 Detailed Description	27
6.6.2 Constructor & Destructor Documentation	27
6.6.2.1 UniverseInterface()	27
6.6.3 Member Function Documentation	27
6.6.3.1 ContractIdToName()	27
6.6.3.2 FindRegion()	28
6.6.3.3 HandleInfo()	28
6.6.3.4 IdToName() [1/2]	28
6.6.3.5 IdToName() [2/2]	29
6.6.3.6 NameTold()	29
6.6.3.7 PrintItemInfo()	29
6.6.3.8 SearchName()	30
Index	31

Chapter 1

Homepage

Welcome to the documentation of this program. Please read the notes below.

1.0.1 Quick Start Notes:

- You can run the program by clicking [eve_market](#) in the root directory
- Type "help" (without the quotes) to see a list of available commands
- Parts of the program may not be working if a character isn't authorized.

1.0.2 Example usage

```
orders "vedmak" "The Forge"
```

- Displays current orders of type Vedmak in the region The Forge

```
history "vedmak"
```

- Display price history of Vedmak in the default region (The Forge)

```
info "Nuclear M"
```

- Displays information about type Nuclear M (a kind of ammunition)

Chapter 2

Program usage

2.1 Commands

`authorize`

- Used to authorize a character

`logout`

- Removes the current character authorization

`set region "{region_name}"`

- Sets the default region to the given region

`set station "{station_name}"`

- Sets the default station the the given station

`orders "{type_name}" ["{region_name}"] [-a] [-de] [-d] [-v] [-p]`

- Displays current buy and sell orders for type {type_name} in region {region_name}, or in the default one if no region name is provided.
- Sell orders are by default price ascending, buy orders price price descending.
- Additional options: -a : Sort type ascending -de : Sort type descending -d : Sort by date -v : Sort by volume remaining -p : Sort by price

`wallet`

- Displays the current account balance of the authorized character transactions
- Displays the transaction history of the authorized character

`my_orders [-a] [-de] [-d] [-v] [-p]`

- Displays the active market orders of the authorized character
- Additional options: -a : Sort type ascending -de : Sort type descending -d : Sort by date -v : Sort by volume remaining -p : Sort by price

`my_order_history [-a] [-de] [-d] [-v] [-p]`

- Displays all completed market orders of the authorized character
- Additional options: -a : Sort type ascending -de : Sort type descending -d : Sort by date -v : Sort by volume remaining -p : Sort by price

`info "{type_name}"`

- Displays information about the given type history "{type_name}" [{"region_name}"]
- Displays price and volume history of type {type_name} in region {region_name}, or the default region if no region name specified.

`assets`

- Displays all assets of the authorized character contracts page_number [{"region_name}"]
- Displays page page_number of public contracts (or all contracts available to the authorized character) in the given region, or the default one if no region name is provided.

`my_contracts`

- Displays all contracts of the authorized character.

`exit`

- Exits the program.

`help`

- Displays the help

Chapter 3

Namespace Index

3.1 Package List

Here are the packages with brief descriptions (if available):

eve_market	9
--------------------------------------	---

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

eve_market.ContractInterface	Main interaction point for the Contract part of ESI.	11
eve_market.MainEsiInterface	Main interface for communicating with the ESI API. Holds references to specialized interfaces, has some helper functions and takes care of authentication of the user.	13
eve_market.MarketInterface	Class that handles the interaction with the Market part of ESI, as well ass the handling of assets.	17
eve_market.Printer	Class that handles printing of various data.	22
eve_market.Program	Entry point of the program.	25
eve_market.UniverseInterface	Class that handles the interaction with the Universe part of ESI, and provides various helpful functions to other parts of the program.	26

Chapter 5

Namespace Documentation

5.1 eve_market Namespace Reference

Classes

- class [ContractInterface](#)
Main interaction point for the Contract part of ESI.
- class [MainEsiInterface](#)
Main interface for communicating with the ESI API. Holds references to specialized interfaces, has some helper functions and takes care of authentication of the user.
- class [MarketInterface](#)
Class that handles the interaction with the Market part of ESI, as well as the handling of assets.
- class [Printer](#)
Class that handles printing of various data.
- class [Program](#)
Entry point of the program.
- class [UniverseInterface](#)
Class that handles the interaction with the Universe part of ESI, and provides various helpful functions to other parts of the program.

Chapter 6

Class Documentation

6.1 eve_market.ContractInterface Class Reference

Main interaction point for the Contract part of ESI.

Public Member Functions

- [ContractInterface](#) ([MainEsiInterface](#) @interface, TextWriter textWriter)
Basic constructor. Only assigns the instances from arguments to fields.
- void [HandleContracts](#) (string[] tokens)
Handles the "contracts" command. Displays a given page of contracts in the input region. If no region is given, the default region is used.
- void [HandleMyContracts](#) (string[] tokens)
Handles the "my_contracts" command. Displays all contracts of the authorized character.
- List< List< string > > [getContractItemString](#) (long contractId, bool isPublic=false)
Helper function. For a given contract, constructs a string of the items in the given contract, with a specific formatting. If listPublic is false, it is assumed that the contract is part of the authorized characters contracts, otherwise it's public.
- List< string > [GetBidStrings](#) (int contractId)
Helper function. Constructs a string containing all the bids from of the given auction contract. Returns an empty lists if no bids are found.

Public Attributes

- TextWriter **output**
Text output stream
- [MainEsiInterface](#) **mainInterface**
Reference to a MainInterface instance

6.1.1 Detailed Description

Main interaction point for the Contract part of ESI.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 ContractInterface()

```
eve_market.ContractInterface.ContractInterface (
    MainEsiInterface @ interface,
    TextWriter textWriter )
```

Basic constructor. Only assigns the instances from arguments to fields.

Parameters

<i>interface</i>	Reference to an (already created) MainEsiInterface instance
<i>textWriter</i>	Text output stream

6.1.3 Member Function Documentation

6.1.3.1 GetBidStrings()

```
List< string > eve_market.ContractInterface.GetBidStrings (
    int contractId )
```

Helper function. Constructs a string containing all the bids from of the given auction contract. Returns an empty lists if no bids are found.

Parameters

<i>contractId</i>	Auction contract ID
-------------------	---------------------

Returns

List of bid strings

6.1.3.2 getContractItemString()

```
List< List< string > > eve_market.ContractInterface.getContractItemString (
    long contractId,
    bool isPublic = false )
```

Helper function. For a given contract, constructs a string of the items in the given contract, with a specific formatting. If listPublic is false, it is assumed that the contract is part of the authorized characters contracts, otherwise it's public.

Parameters

<i>contract↵ Id</i>	ID of the contract
<i>isPublic</i>	Whether the contract is a public or character one

Returns

List of item info strings

6.1.3.3 HandleContracts()

```
void eve_market.ContractInterface.HandleContracts (
    string[] tokens )
```

Handles the "contracts" command. Displays a given page of contracts in the input region. If no region is given, the default region is used.

Parameters

<i>tokens</i>	Command line tokens
---------------	---------------------

6.1.3.4 HandleMyContracts()

```
void eve_market.ContractInterface.HandleMyContracts (
    string[] tokens )
```

Handles the "my_contracts" command. Displays all contracts of the authorized character.

Parameters

<i>tokens</i>	Command line tokens (included for uniformity of API, not used)
---------------	--

The documentation for this class was generated from the following file:

- C:/Users/Samo/Repos/eve_market/ContractInterface.cs

6.2 eve_market.MainEsiInterface Class Reference

Main interface for communicating with the ESI API. Holds references to specialized interfaces, has some helper functions and takes care of authentication of the user.

Public Member Functions

- [MainEsiInterface](#) (TextWriter writer)
Basic constructor, creates the instances of other interfaces and the [Printer](#), and also creates the EsiClient used for communication with ESI. If there is an auth token from a previous session, uses it again.
- bool [IsIdField](#) (string field)
Helper function which returns true if a given json field is an ID, and thus the corresponding name has to be resolved.
- string [GenerateStateString](#) (int length=10)
Generates a state string of given length, used for OAuth.
- void [HandleHelp](#) (string[] tokens)
Handles the "help" command. Prints the text inside the help.txt file.
- string [StringFromSlice](#) (string[] tokens, int offset, int size, char sep=' ')
Helper function which creates a string from a given slice of the "tokens" string array.
- bool [CheckAuthorization](#) ()
Checks whether a character is authorized. Returns the value of IsAuthorized and
- async void [HandleLogout](#) (string[] tokens)
Logs the authorized character out (if there is any)
- async void [HandleAuthorize](#) (string[] tokens)
Handles the authorization of a user. Takes care of the OAuth authorization process, and refers the end-user to the auth page.

Public Attributes

- EsiClient **Client**
ESI API Client instance
- TextWriter **Output**
Stream to which output will be written
- [MarketInterface](#) **marketInterface**
[MarketInterface](#) instance, used for accessing the Market portion of ESI
- [UniverseInterface](#) **universeInterface**
[UniverseInterface](#) instance, used for accessing the Universe portion of ESI
- [ContractInterface](#) **contractInterface**
[ContractInterface](#) instance, used for accessing the Universe portion of ESI
- [Printer](#) **printer**
[Printer](#) instance, used for nice printing of various data gathered from ESI

Properties

- bool **IsAuthorized** [get]
Bool which is set to true if a character is authorized
- SsoToken **AuthToken** [set]
Auth Token obtained from the server
- AuthorizedCharacterData **AuthData** [get, set]
Information about the authorized character

6.2.1 Detailed Description

Main interface for communicating with the ESI API. Holds references to specialized interfaces, has some helper functions and takes care of authentication of the user.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 MainEsiInterface()

```
eve_market.MainEsiInterface.MainEsiInterface (
    TextWriter writer )
```

Basic constructor, creates the instances of other interfaces and the [Printer](#), and also creates the EsiClient used for communication with ESI. If there is an auth token from a previous session, uses it again.

Parameters

<i>writer</i>	Output text stream
---------------	--------------------

6.2.3 Member Function Documentation

6.2.3.1 CheckAuthorization()

```
bool eve_market.MainEsiInterface.CheckAuthorization ( )
```

Checks whether a character is authorized. Returns the value of IsAuthorized and

Returns

6.2.3.2 GenerateStateString()

```
string eve_market.MainEsiInterface.GenerateStateString (
    int length = 10 )
```

Generates a state string of given length, used for OAuth.

Parameters

<i>length</i>	length of the state string
---------------	----------------------------

Returns

State string

6.2.3.3 HandleAuthorize()

```
async void eve_market.MainEsiInterface.HandleAuthorize (
    string[] tokens )
```

Handles the authorization of a user. Takes care of the OAuth authorization process, and refers the end-user to the auth page.

Parameters

<i>tokens</i>	Command line tokens. Included for the uniformity of API
---------------	---

6.2.3.4 HandleHelp()

```
void eve_market.MainEsiInterface.HandleHelp (
    string[] tokens )
```

Handles the "help" command. Prints the text inside the help.txt file.

Parameters

<i>tokens</i>	Command line tokens. Only included for uniformity of API
---------------	--

6.2.3.5 HandleLogout()

```
async void eve_market.MainEsiInterface.HandleLogout (
    string[] tokens )
```

Logs the authorized character out (if there is any)

Parameters

<i>tokens</i>	Command line tokens. Only included for the uniformity of API
---------------	--

6.2.3.6 IsIdField()

```
bool eve_market.MainEsiInterface.IsIdField (
    string field )
```

Helper function which returns true if a given json field is an ID, and thus the corresponding name has to be resolved.

Parameters

<i>field</i>	Name of the field
--------------	-------------------

Returns

True if

6.2.3.7 StringFromSlice()

```
string eve_market.MainEsiInterface.StringFromSlice (
    string[] tokens,
    int offset,
    int size,
    char sep = ' ' )
```

Helper function which creates a string from a given slice of the "tokens" string array.

Parameters

<i>tokens</i>	Array of strings
<i>offset</i>	Beginning of the slice
<i>size</i>	Length of the slice
<i>sep</i>	Separator to be added between tokens

Returns

String of joined tokens starting at "offset" with a length of "size", separated by "sep"

The documentation for this class was generated from the following file:

- C:/Users/Samo/Repos/eve_market/MainEsiInterface.cs

6.3 eve_market.MarketInterface Class Reference

Class that handles the interaction with the Market part of ESI, as well as the handling of assets.

Public Types

- enum [SortOrder](#)
Enum choosing the sort order of various items
- enum [SortBy](#)
Enum for choosing the field by which orders should be sorted.

Public Member Functions

- [MarketInterface](#) ([MainEsiInterface](#) @interface, `TextWriter textWriter`)
Basic constructor. Only assigns the instances from arguments to fields.
- void [HandleDefaults](#) (`string[] tokens`)
Handles the "set" command. Sets default values for region or station used when displaying market listings.
- void [HandleTransactions](#) (`string[] tokens`)
Handles the "transactions" command. Displays all of the recent transactions of the authorized characters
- void [SortOrders](#) (`List< Order > orders`, [SortBy](#) sortByType, [SortOrder](#) sortOrder)
Sorts orders based on various fields defined by SortBy enum and in ascending or descending order
- void [SortItems](#) (`List< Item > items`, [SortBy](#) sortByType, [SortOrder](#) sortOrder)
Sorts a list of items according to the given field in the given order
- void [HandleHistory](#) (`string[] tokens`)
Prints type history in a given region.
- void [HandleAssets](#) (`string[] tokens`, [SortBy](#) sortByType=[SortBy.Station](#), [SortOrder](#) sortOrder=[SortOrder.↔](#)
Descending)
Handles the "assets" command. Displays all assets of the authorized character as a table.
- void [HandleOrders](#) (`string[] tokens`, [SortBy](#) sortByType=[SortBy.Price](#), [SortOrder](#) sortOrder=[SortOrder.↔](#)
Descending)
Handles the "orders" command. Finds all orders of the given type in the given location. If not location is given, the default region will be used
- void [HandleMyOrders](#) (`string[] tokens`, [SortBy](#) sortByType=[SortBy.Date](#), [SortOrder](#) sortOrder=[SortOrder.↔](#)
Descending)
Handles the "my_orders" command. Gathers orders from the API, sorts them and prints them out to output
- void [HandleOrderHistory](#) (`string[] tokens`, [SortBy](#) sortByType=[SortBy.Date](#), [SortOrder](#) sortOrder=[SortOrder.↔](#)
Order.Descending)
Handles the "my_order_history" command. Gathers the order history of a given character, potentially sorts it and prints it out.
- async void [HandleWallet](#) (`string[] tokens`)
Gathers information about the authorized characters' wallet and prints it out to output

Public Attributes

- `TextWriter output`
Text output stream
- [MainEsiInterface](#) `mainInterface`
Reference to a [MainEsiInterface](#) instance
- long `defaultRegionId` = 10000002
Id of the default region ("The Forge" currently)
- long `defaultStationId` = 60003760
Id of the default station ("Jita IV - Moon 4 - Caldari Navy Assembly Plant")

6.3.1 Detailed Description

Class that handles the interaction with the Market part of ESI, as well as the handling of assets.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 MarketInterface()

```
eve_market.MarketInterface.MarketInterface (
    MainEsiInterface @ interface,
    TextWriter textWriter )
```

Basic constructor. Only assigns the instances from arguments to fields.

Parameters

<i>interface</i>	Reference to an (already created) MainEsiInterface instance
<i>textWriter</i>	Text output stream

6.3.3 Member Function Documentation

6.3.3.1 HandleAssets()

```
void eve_market.MarketInterface.HandleAssets (
    string[] tokens,
    SortBy sortByType = SortBy.Station,
    SortOrder sortOrder = SortOrder.Descending )
```

Handles the "assets" command. Displays all assets of the authorized character as a table.

Parameters

<i>tokens</i>	Command line tokens
<i>sortByType</i>	Which field to sort by
<i>sortOrder</i>	Ascending/Descending

6.3.3.2 HandleDefaults()

```
void eve_market.MarketInterface.HandleDefaults (
    string[] tokens )
```

Handles the "set" command. Sets default values for region or station used when displaying market listings.

Parameters

<i>tokens</i>	Tokens from the input, including the original command
---------------	---

6.3.3.3 HandleHistory()

```
void eve_market.MarketInterface.HandleHistory (
    string[] tokens )
```

Prints type history in a given region.

Parameters

<i>tokens</i>	Command line tokens
---------------	---------------------

6.3.3.4 HandleMyOrders()

```
void eve_market.MarketInterface.HandleMyOrders (
    string[] tokens,
    SortBy sortByType = SortBy.Date,
    SortOrder sortOrder = SortOrder.Descending )
```

Handles the "my_orders" command. Gathers orders from the API, sorts them and prints them out to output

Parameters

<i>tokens</i>	Contains tokens from the command line.
<i>sortByType</i>	Enum which determines the field by which to sort the orders
<i>sortOrder</i>	Enum for ascending/descending sort order

6.3.3.5 HandleOrderHistory()

```
void eve_market.MarketInterface.HandleOrderHistory (
    string[] tokens,
    SortBy sortByType = SortBy.Date,
    SortOrder sortOrder = SortOrder.Descending )
```

Handles the "my_order_history" command. Gathers the order history of a given character, potentially sorts it and prints it out.

Parameters

<i>tokens</i>	
<i>sortByType</i>	
<i>sortOrder</i>	

6.3.3.6 HandleOrders()

```
void eve_market.MarketInterface.HandleOrders (
    string[] tokens,
    SortBy sortByType = SortBy.Price,
    SortOrder sortOrder = SortOrder.Descending )
```

Handles the "orders" command. Finds all orders of the given type in the given location. If not location is given, the default region will be used

Parameters

<i>tokens</i>	
<i>sortByType</i>	
<i>sortOrder</i>	

6.3.3.7 HandleTransactions()

```
void eve_market.MarketInterface.HandleTransactions (
    string[] tokens )
```

Handles the "transactions" command. Displays all of the recent transactions of the authorized characters

Parameters

<i>tokens</i>	Command line tokens (included for uniformity of API, not used)
---------------	--

6.3.3.8 HandleWallet()

```
async void eve_market.MarketInterface.HandleWallet (
    string[] tokens )
```

Gathers information about the authorized characters' wallet and prints it out to output

Parameters

<i>tokens</i>	Included for the uniformity of API, not used
---------------	--

6.3.3.9 SortItems()

```
void eve_market.MarketInterface.SortItems (
    List< Item > items,
```

```
SortBy sortByType,
SortOrder sortOrder )
```

Sorts a list of items according to the given field in the given order

Parameters

<i>items</i>	List of items to sort
<i>sortByType</i>	Field to sort by
<i>sortOrder</i>	Order to sort by

6.3.3.10 SortOrders()

```
void eve_market.MarketInterface.SortOrders (
    List< Order > orders,
    SortBy sortByType,
    SortOrder sortOrder )
```

Sorts orders based on various fields defined by SortBy enum and in ascending or descending order

Parameters

<i>orders</i>	
<i>sortByType</i>	
<i>sortOrder</i>	

The documentation for this class was generated from the following file:

- C:/Users/Samo/Repos/eve_market/MarketInterface.cs

6.4 eve_market.Printer Class Reference

Class that handles printing of various data.

Public Member Functions

- [Printer](#) ([MainEsiInterface](#) esiInterface, [TextWriter](#) writer)
Basic constructor. Only assigns the instances from arguments to fields.
- void [PrintLine](#) (int length)
Prints a line of a given length to the output stream.
- void [PrintLine](#) (List< string > fields, int width)
Prints a line that adapts to the number of fields, and the width corresponding to one field.
- void [PrintTableHeader](#) (List< string > fields, int width)
Prints the header of a table, which contains the given fields, with a cell width of "width".
- void [PrintContracts](#) (List< [ESI.NET.Models.Contracts.Contract](#) > contracts, bool listPublic=false)
Prints the contracts from arguments. Warning, contracts have a lot of variations and this function is quite extensive in its printing. Use listPublic if you're trying to print public contracts.
- void [PrintObjList](#)< T > (List< T > objList, int width, int rows, List< string > fields, List< string > fieldDesc)
Prints a list of json-convertible objects as a table, with a given cell width and number of rows. Only prints the fields named in the "fields". If a field is an ID field, also resolves it's name before printing.

Public Attributes

- **TextWriter output**
Text output stream
- [MainEsiInterface](#) **mainEsiInterface**
Reference to a MainIntreface instance.

6.4.1 Detailed Description

Class that handles printing of various data.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 Printer()

```
eve_market.Printer.Printer (
    MainEsiInterface esiInterface,
    TextWriter writer )
```

Basic constructor. Only assigns the instances from arguments to fields.

Parameters

<i>esiInterface</i>	Reference to an (already created) MainEsiInterface instance
<i>writer</i>	Text output stream

6.4.3 Member Function Documentation

6.4.3.1 PrintContracts()

```
void eve_market.Printer.PrintContracts (
    List< ESI.NET.Models.Contracts.Contract > contracts,
    bool listPublic = false )
```

Prints the contracts from arguments. Warning, contracts have a lot of variations and this function is quite extensive in its printing. Use listPublic if you're trying to print public contracts.

Parameters

<i>contracts</i>	List of contracts to print
<i>listPublic</i>	Whether public contract is being printed

6.4.3.2 PrintLine() [1/2]

```
void eve_market.Printer.PrintLine (
    int length )
```

Prints a line of a given length to the output stream.

Parameters

<i>length</i>	Length of the line
---------------	--------------------

6.4.3.3 PrintLine() [2/2]

```
void eve_market.Printer.PrintLine (
    List< string > fields,
    int width )
```

Prints a line that adapts to the number of fields, and the width corresponding to one field.

Parameters

<i>fields</i>	List of fields
<i>width</i>	Width for one field

6.4.3.4 PrintObjList< T >()

```
void eve_market.Printer.PrintObjList< T > (
    List< T > objList,
    int width,
    int rows,
    List< string > fields,
    List< string > fieldDesc )
```

Prints a list of json-convertible objects as a table, with a given cell width and number of rows. Only prints the fields named in the "fields". If a field is an ID field, also resolves it's name before printing.

Template Parameters

<i>T</i>	JsonObject-convertible type
----------	-----------------------------

Parameters

Parameters

<i>objList</i>	List of items to print
<i>width</i>	Table cell width
<i>rows</i>	Number of rows. If it's less than the length of "objList", prints only the first "rows" items.
<i>fields</i>	Keys of the fields to print.

6.4.3.5 PrintTableHeader()

```
void eve_market.Printer.PrintTableHeader (
    List< string > fields,
    int width )
```

Prints the header of a table, which contains the given fields, with a cell width of "width".

Parameters

<i>fields</i>	List of field names
<i>width</i>	Width of a cell in the table

The documentation for this class was generated from the following file:

- C:/Users/Samo/Repos/eve_market/Printer.cs

6.5 eve_market.Program Class Reference

Entry point of the program.

Static Public Member Functions

- static void [ParseInput](#) (TextReader streamReader, TextWriter writer, [MainEsiInterface](#) apiInterface)
Simple entry point function to the program. Turns the command line input into tokens and calls a function based on the value of the first token.

6.5.1 Detailed Description

Entry point of the program.

6.5.2 Member Function Documentation

6.5.2.1 ParseInput()

```
static void eve_market.Program.ParseInput (
    TextReader streamReader,
    TextWriter writer,
    MainEsiInterface apiInterface ) [static]
```

Simple entry point function to the program. Turns the command line input into tokens and calls a function based on the value of the first token.

Parameters

<i>streamReader</i>	Input stream
<i>writer</i>	Output stream
<i>apiInterface</i>	MainEsiInterface instance

The documentation for this class was generated from the following file:

- C:/Users/Samo/Repos/eve_market/Program.cs

6.6 eve_market.UniverseInterface Class Reference

Class that handles the interaction with the Universe part of ESI, and provides various helpful functions to other parts of the program.

Public Member Functions

- [UniverseInterface](#) ([MainEsiInterface](#) @interface, TextWriter textWriter)
Basic constructor. Only assigns the instances from arguments to fields.
- void [PrintItemInfo](#) (string itemName)
Finds the closest matching item and prints its description
- void [HandleInfo](#) (string[] tokens)
Handles the "info" command.
- long[] [SearchName](#) (string query, SearchCategory category)
Find names of types/objects which contain a given string query and returns the list of matching ids
- long [FindRegion](#) (long id, SearchCategory type)
Finds the region of a given object lower in the hierarchy
- long [NameToId](#) (string name, SearchCategory category)
For a given name of object and its type, finds its ID. Doesn't require exact names, will find the closest match and returns its ID. If no matches are found, returns -1
- void [ContractIdToName](#) (List< long > ids, string fieldName)
Function that resolves the ids in a given contract, categories are determined based on the name of the field to which these ids belong. Results are cached and are available later from the [IdToName\(\)](#) function.
- List< string > [IdToName](#) (List< long > ids)
Finds the names corresponding to these IDs, caches them and returns them as a list. Make sure all IDs are unique and of the same type. It is better to later query the [IdToName\(long id\)](#) method for the results, instead of using this list.
- string [IdToName](#) (long id)
Overload for singular ids. If id is in the cache, returns its cached value. Otherwise calls [IdToName\(\)](#) with a List of length 1 containing this id.

Public Attributes

- TextWriter **output**
Text output stream
- [MainEsiInterface](#) **mainInterface**
Reference to a [MainEsiInterface](#) instance

6.6.1 Detailed Description

Class that handles the interaction with the Universe part of ESI, and provides various helpful functions to other parts of the program.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 UniverseInterface()

```
eve_market.UniverseInterface.UniverseInterface (
    MainEsiInterface @ interface,
    TextWriter textWriter )
```

Basic constructor. Only assigns the instances from arguments to fields.

Parameters

<i>interface</i>	Reference to an (already created) MainEsiInterface instance
<i>textWriter</i>	Text output stream

6.6.3 Member Function Documentation

6.6.3.1 ContractIdToName()

```
void eve_market.UniverseInterface.ContractIdToName (
    List< long > ids,
    string fieldName )
```

Function that resolves the ids in a given contract, categories are determined based on the name of the field to which these ids belong. Results are cached and are available later from the [IdToName\(\)](#) function.

Parameters

<i>ids</i>	List of ids to resolve
<i>fieldName</i>	Name of the field to which these ids belong

6.6.3.2 FindRegion()

```
long eve_market.UniverseInterface.FindRegion (
    long id,
    SearchCategory type )
```

Finds the region of a given object lower in the hierarchy

Parameters

<i>id</i>	ID of the object
<i>type</i>	Type of the object

Returns

ID of the region to which this object belongs

6.6.3.3 HandleInfo()

```
void eve_market.UniverseInterface.HandleInfo (
    string[] tokens )
```

HAndles the "info" command.

Parameters

<i>tokens</i>	
---------------	--

6.6.3.4 IdToName() [1/2]

```
List< string > eve_market.UniverseInterface.IdToName (
    List< long > ids )
```

Finds the names corresponding to these IDs, caches them and returns them as a list. Make sure all IDs are unique and of the same type. It is better to later query the [IdToName\(long id\)](#) method for the results, instead of using this list.

Parameters

<i>ids</i>	List of ids to resolve
------------	------------------------

Returns

Name corresponding to the given ID

6.6.3.5 IdToName() [2/2]

```
string eve_market.UniverseInterface.IdToName (
    long id )
```

Overload for singular ids. If id is in the cache, returns it's cached value. Otherwise calls [IdToName\(\)](#) with a List of length 1 containing this id.

Parameters

<i>id</i>	Id to resolve
-----------	---------------

Returns

Resolved name

6.6.3.6 NameToId()

```
long eve_market.UniverseInterface.NameToId (
    string name,
    SearchCategory category )
```

For a given name of object and it's type, finds its ID. Doesn't require exact names, will find the closest match and returns its ID If no matches are found, returns -1

Parameters

<i>name</i>	Name of the object
<i>category</i>	Enum describing the category of the object

Returns

ID of the given query type/object

6.6.3.7 PrintItemInfo()

```
void eve_market.UniverseInterface.PrintItemInfo (
    string itemName )
```

Finds the closest matching item and prints its description

Parameters

<i>itemName</i>	Command line tokens
-----------------	---------------------

6.6.3.8 SearchName()

```
long[] eve_market.UniverseInterface.SearchName (
    string query,
    SearchCategory category )
```

Find names of types/objects which contain a given string query and returns the list of matching ids

Parameters

<i>query</i>	
<i>category</i>	Enum describing the category of the object

Returns

A list of ids of the matches

The documentation for this class was generated from the following file:

- C:/Users/Samo/Repos/eve_market/UniverseInterface.cs

Index

- CheckAuthorization
 - eve_market.MainEsilInterface, [15](#)
- ContractIdToName
 - eve_market.UniverseInterface, [27](#)
- ContractInterface
 - eve_market.ContractInterface, [12](#)
- eve_market, [9](#)
- eve_market.ContractInterface, [11](#)
 - ContractInterface, [12](#)
 - GetBidStrings, [12](#)
 - getContractItemString, [12](#)
 - HandleContracts, [13](#)
 - HandleMyContracts, [13](#)
- eve_market.MainEsilInterface, [13](#)
 - CheckAuthorization, [15](#)
 - GenerateStateString, [15](#)
 - HandleAuthorize, [16](#)
 - HandleHelp, [16](#)
 - HandleLogout, [16](#)
 - IsIdField, [16](#)
 - MainEsilInterface, [15](#)
 - StringFromSlice, [17](#)
- eve_market.MarketInterface, [17](#)
 - HandleAssets, [19](#)
 - HandleDefaults, [19](#)
 - HandleHistory, [19](#)
 - HandleMyOrders, [20](#)
 - HandleOrderHistory, [20](#)
 - HandleOrders, [20](#)
 - HandleTransactions, [21](#)
 - HandleWallet, [21](#)
 - MarketInterface, [18](#)
 - SortItems, [21](#)
 - SortOrders, [22](#)
- eve_market.Printer, [22](#)
 - PrintContracts, [23](#)
 - Printer, [23](#)
 - PrintLine, [24](#)
 - PrintObjList< T >, [24](#)
 - PrintTableHeader, [25](#)
- eve_market.Program, [25](#)
 - ParseInput, [25](#)
- eve_market.UniverseInterface, [26](#)
 - ContractIdToName, [27](#)
 - FindRegion, [28](#)
 - HandleInfo, [28](#)
 - IdToName, [28](#), [29](#)
 - NameTold, [29](#)
 - PrintItemInfo, [29](#)
 - SearchName, [30](#)
 - UniverseInterface, [27](#)
- FindRegion
 - eve_market.UniverseInterface, [28](#)
- GenerateStateString
 - eve_market.MainEsilInterface, [15](#)
- GetBidStrings
 - eve_market.ContractInterface, [12](#)
- getContractItemString
 - eve_market.ContractInterface, [12](#)
- HandleAssets
 - eve_market.MarketInterface, [19](#)
- HandleAuthorize
 - eve_market.MainEsilInterface, [16](#)
- HandleContracts
 - eve_market.ContractInterface, [13](#)
- HandleDefaults
 - eve_market.MarketInterface, [19](#)
- HandleHelp
 - eve_market.MainEsilInterface, [16](#)
- HandleHistory
 - eve_market.MarketInterface, [19](#)
- HandleInfo
 - eve_market.UniverseInterface, [28](#)
- HandleLogout
 - eve_market.MainEsilInterface, [16](#)
- HandleMyContracts
 - eve_market.ContractInterface, [13](#)
- HandleMyOrders
 - eve_market.MarketInterface, [20](#)
- HandleOrderHistory
 - eve_market.MarketInterface, [20](#)
- HandleOrders
 - eve_market.MarketInterface, [20](#)
- HandleTransactions
 - eve_market.MarketInterface, [21](#)
- HandleWallet
 - eve_market.MarketInterface, [21](#)
- IdToName
 - eve_market.UniverseInterface, [28](#), [29](#)
- IsIdField
 - eve_market.MainEsilInterface, [16](#)
- MainEsilInterface
 - eve_market.MainEsilInterface, [15](#)
- MarketInterface
 - eve_market.MarketInterface, [18](#)

NameTold
 eve_market.UniverseInterface, [29](#)

ParselInput
 eve_market.Program, [25](#)

PrintContracts
 eve_market.Printer, [23](#)

Printer
 eve_market.Printer, [23](#)

PrintItemInfo
 eve_market.UniverseInterface, [29](#)

PrintLine
 eve_market.Printer, [24](#)

PrintObjList< T >
 eve_market.Printer, [24](#)

PrintTableHeader
 eve_market.Printer, [25](#)

SearchName
 eve_market.UniverseInterface, [30](#)

SortItems
 eve_market.MarketInterface, [21](#)

SortOrders
 eve_market.MarketInterface, [22](#)

StringFromSlice
 eve_market.MainEsiInterface, [17](#)

UniverseInterface
 eve_market.UniverseInterface, [27](#)