Algo-trading market Client LLD

## **MarketRequest class**

#### Functionality

Provides parsing for JSON requests and calls the RequestOperations class for performing actions of the request object.

#### Request JSON fields

* All Requests:
  + “type” - request type, one of the following : {“buy”, “sell”, “queryBuySell”, “queryUser”, “queryMarket”, “cancelBuySell”}
* Buy/Sell requests:
  + "commodity" - commodity ID.
  + “amount” - buy/sell amount.
  + “price” - buy/sell price.
* Query buy/sell request:
  + “id” - request ID to query.
* Query user request:
  + No arguments
* Query market request:
  + "commodity" - commodity ID.
* Cancel buy/sell request:
  + “id” - request ID to query.

**Files:**

* Data layer:
* *DataEntries*

AllDataRequest.cs

AllMarketRequest.cs

ItemAskBid.cs

MarketBuySell.cs: Stores the buy or sell response from the server.

MarketCommodityOffer.cs: Stores Ask price and Bid price.

MarketItemQuery.cs: Stores the Data of active buy or sell request.

MarketUserData.cs: Stores the current state of the user.

MarketUserRequests.cs

* *Loggers*

Debugger.cs

HistoryLogger.cs

* *MarketRequests*

BuyRequest.cs: Stores the buy request related information.

SellRequest.cs: Stores the sell request related information.

CancelBuySellRequest.cs: Stores a cancel request related information.

QuerybuySellRequest.cs: Stores an information about ongoing request.

QueryMarketRequest.cs: Stores the information about the market.

QueryUserRequest.cs: Stores the information about the user.

QueryUserRequests.cs

QueryAllMarketRequest.cs

IMarketClient.cs

MarketClientClass.cs: In charge of communicating with the server.

SimpleHTTPClient.cs

* Logic layer:
* Presentation layer:

Presentation.cs: In charge of communicating with the user, deal with illegal input, communicates with the logic layer- and represent the results to the user.

Details about:

Terminology, Trading framework and Actors can be found here:

<https://goo.gl/h5MyZu>