



Select a definition

default

# Api Documentation <sup>1.0</sup>

[ Base URL: localhost:8080/ ]

<http://localhost:8080/v2/api-docs>

Api Documentation

[Terms of service](#)[Apache 2.0](#)

## basic-error-controller Basic Error Controller



GET /error error

HEAD /error error

POST /error error

PUT /error error

DELETE /error error

OPTIONS /error error

PATCH /error error

## message-controller REST API related to Web Socket Messages



GET /chat/messages/{target} Get list of Messages sent to {target}

## stock-controller REST APIs related to the stock entity



## user-controller REST APIs for the user/admin entity



### user

**PUT****/banuser/{uid}/byadmin/{aid}**

Bans a user based on their uid, but ensures that this is being done by an admin with aid

**POST****/login** create a new login request with the request body**PUT****/unban/{uid}/byadmin/{aid}**

Unbans the user with the uid, but must be done by an admin called by aid

**GET****/user/{id}** Gets the list of purchased stocks for a specific user**GET****/userByName/{username}** Get user response from their username {username}**GET****/users** Gets a list of all the users in the userRepository**POST****/users** create new user based on request body**GET****/users/{id}** Gets the user and their information by their id**PUT****/users/{id}** Update the user {id} with information from request body**DELETE****/users/{id}** Remove User {id}

### userbuyandsell

**GET****/buy/{id}/user/{uid}/amt/{amount}**

Has the user purchase a stock, will show the money spent from the transaction

**GET****/sell/{id}/user/{uid}/{numStocks}**

Has the user sell a stock, returns their money that they earned from the transaction

**PUT****/users/{userId}/stocks/{stockId}/{numPurchasing}**

Assigns a stock to the user, essentially having them purchase it just a little different

### friendgroup



**GET** **/chat/messages/friendgroup** Get list of all friend group objects

**GET** **/friendgroup** Get list of all friend group objects

**GET** **/friendgroup/{groupName}** create list of users in friend group {groupName}

**POST** **/friendgroup/{groupName}** create new friend group named {groupName}

**PUT** **/friendgroup/{groupName}/{userID}** Add User {userId} to friend group {groupName}

**DELETE** **/friendgroup/{groupName}/{userID}** Remove User {userId} from friend group {groupName}

## stock



**POST** **/newstocks/{uid}** Creates a new stock based on a stock json object, admins have permissions to do this

**GET** **/stock/{id}** Gets all users single stock based on its id value from the stockRepository

**GET** **/stockchange/{id}** Gets the current price from a stock based on its id

**GET** **/stocks** Gets a list of all the stocks in the stockRepository

**POST** **/stocks** Creates a new stock that gets added to the stockRepository

**PUT** **/stocks** Updates every stock in the stockRepository from the AlphaVantage API, based on current data

**GET** **/stocks/{id}** Gets a stock to return based on its id

**PUT** **/stocks/{id}** Updates a stock from the AlphaVantage API based on the stock id

**DELETE** **/stocks/{id}** Deletes a stock from the stockRepository based on its id

**DELETE** **/stocks/{sid}/{uid}** Deletes a stock based on its id, if the admin wants to have that stock deleted

**GET****/stocks/history  
/{id}**

Gets the price history of a stock from AlphaVantage API based on the id of the stock

**GET****/stocks/news  
/{id}**

Gets the current news for a stock from AlphaVantage API based on the id of that stock

**GET****/stocksUpdate  
/{symbol}**

Gets the live stock information from AlphaVantage to apply to one of our stocks so that current information is displayed.

**PUT****/update/{uid}** Updates all the stocks if the admin user calls upon the stocks to be updated**Models**