



/v3/api-docs

Explore

Java Spring Boot REST API Microservice v1.0 OAS3

/v3/api-docs

Java Spring Boot REST API for Stock Market Brokers

Low Jie Wei - Website
Send email to Low Jie Wei

Servers

http://localhost:8080

Authorize

User Controller To sign up for an account and log in with the provided login credentials. Provides endpoint for withdrawing and depositing cash into the created account.

PUT

/api/withdraw

Withdraw from user account

To collect profits from user's accounts.

Parameters

Try it out

No parameters

Request body required

application/json

Example Value

Schema

```
{
  "amount": 1000
}
```

Responses

Code	Description	Links
200	Successfully withdrew amount from user's account	No links
	<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value</div> <div>Schema</div> <div><pre>{ "balance": 2000 }</pre></div>	
400	BAD REQUEST	No links
404	NOT FOUND (when user does not exist)	No links

PUT

/api/deposit

Deposit to user account

To load up funds to user's accounts to purchase stocks.

Parameters

Try it out

No parameters

Request body required

application/json

Example Value

Schema

```
{
  "amount": 1000
}
```

Responses

Code	Description	Links
200	Successfully deposited amount into user's account	No links
	<div>Media type<div>application/json</div>Controls Accept header.</div> <div>Example ValueSchema<div><pre>{ "balance": 2000 }</pre></div></div>	
404	NOT FOUND (when user does not exist)	No links

POST

/register

Register user

^

🔒

Adds the user to the database. Must have unique email and password.

Parameters

Try it out

No parameters

Request body required

application/json

Example Value

Schema

```
{
  "username": "leonlow",
  "email": "leonlow@email.com",
  "password": "Password0!"
}
```

Responses

Code	Description	Links
201	Sign Up was successful!	No links
	<div>Media type</div>	

 |

Code

Description

Links

application/json

Controls Accept header.

Example ValueSchema

```
{  "username": "leonlow",  "email": "leonlow@email.com"}  
```

400

BAD REQUEST

No links

409

CONFLICT

No links

POST

/login

Login user

⌵🔒

Stock Controller

Connecting to Yahoo Finance API to retrieve real time daily stock data.

⌵

GET

/yahoo-api/stocks/{stockTicker}

Get stock data

⌵🔒

Retrieves the price of the stock and other fundamental details of the business.

Parameters

Try it out

Name

Description

stockTicker★ required

string
(path)

stockTicker

Responses

Code

Description

Links

200

Successfully retrieved stock details from Yahoo Finance API

No links

Media type

application/json

Controls Accept header.

Example ValueSchema

```
{  "stockName": "Apple Inc.",  "stockTicker": "AAPL",  "previousDayClose": 167.63,  "stockPrice": 166.65,  "dayHigh": 167.87,  "dayLow": 165.56,  "dayChangeInPercent": -0.58,  "get_52WeekChangeInPercent": -5.39,  "get_52WeekHigh": 176.15,  "yearDividendInPercent": 0.54286225}  
```

Order Controller

To submit an order to purchase or sell a stock.

⌵

GET

/api/orders

Get All Orders

⌵🔒

Retrieves all the orders of the user.

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Successfully retrieved all the orders of the user. <div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div>Example Value Schema</div><div><pre>[{ "orderId": 1, "stockTicker": "AAPL", "orderType": "BUY", "noOfShares": 60, "cost": 143.21, "orderDate": "2023-04-22T05:12:25.348Z" }]</pre></div></div>	No links

POST /api/orders Submit Order

^ 🔒

Submits a order to BUY or SELL a stock.

Parameters

Try it out

No parameters

Request body required

application/json

Example Value Schema

```
{
  "stockTicker": "AAPL",
  "orderType": "BUY",
  "noOfShares": 60,
  "cost": 149.86
}
```

Responses

Code	Description	Links
201	Successfully created an order when the stock is bought or sold. <div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div>Example Value Schema</div><div><pre>{ "stockTicker": "AAPL", "orderType": "BUY", "noOfShares": 60, "cost": 149.86 }</pre></div></div>	No links
400	BAD REQUEST	No links

Code

Description

Links

404

NOT FOUND - Order Not Found

No links

500

INTERNAL SERVER ERROR - Unable to connect to Yahoo Finance API.

No links

GET

/api/orders/{orderId}

Get Order

^

🔒

Retrieves a specific order of the user based on order id.

Parameters

Try it out

Name

Description

orderId

★ required

integer(\$int32)

(path)

orderId

Responses

Code

Description

Links

200

Successfully retrieved the specified order of the user.

No links

Media type

application/json

Controls Accept header.

Example Value

Schema

{

"orderId": 1,

"stockTicker": "AAPL",

"orderType": "BUY",

"noOfShares": 60,

"cost": 143.21,

"orderDate": "2023-04-22T05:12:25.350Z"

}

Watchlist Controller

Allow user to save stocks into their watchlist to monitor prices or other details related to the stock.

^

POST

/api/watchlist/{stockTicker}

Add Watchlist

^

🔒

Adds a stock the the watchlist of the user.

Parameters

Try it out

Name

Description

stockTicker

★ required

string

(path)

stockTicker

Responses

Code	Description	Links
201	Successfully added the stock to the watchlist of the user. <div>Media type <div>application/json</div><div>Controls Accept header.</div></div> <div>Example Value Schema</div> <div><pre>{ "userId": 1, "stockName": "JPMorgan Chase & Co.", "stockTicker": "JPM", "price": 141.01, "previousDayClose": 138.76, "get_52WeekHigh": 167.87, "get_52WeekLow": 102.32, "marketCapInBillions": 123.21, "annualDividendYield": 4.23}</pre></div>	No links
404	NOT FOUND - Stock Ticker is invalid	No links
409	CONFLICT - Stock is already in watchlist	No links
500	INTERNAL SERVER ERROR - Unable to connect to Yahoo Finance API	No links

DELETE

/api/watchlist/{stockTicker}

Remove Watchlist

^

🔒

Removes a stock from the watchlist of the user.

Parameters

Try it out

Name	Description
<div>stockTicker * required</div> <div>string</div> <div>(path)</div>	<div>stockTicker</div>

Responses

Code	Description	Links
204	Successfully deleted the stock from the watchlist of the user.	No links

GET

/api/watchlist

Get Watchlist

^

🔒

Retrieves the watchlist of the user for all the stocks

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Successfully retrieves the watchlist for all the stocks. Media type <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <pre>[{ "userId": 1, "stockName": "JPMorgan Chase & Co.", "stockTicker": "JPM", "price": 141.01, "previousDayClose": 138.76, "get_52WeekHigh": 167.87, "get_52WeekLow": 102.32, "marketCapInBillions": 123.21, "annualDividendYield": 4.23 }]</pre>	No links
404	NOT FOUND - Watchlist is empty	No links

Portfolio Controller To retrieve the portfolio of the user to view stocks that have been purchased.

GET

/api/portfolio

Get Portfolio

Retrieves the entire portfolio of stocks of the user.

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	<div>Successfully retrieved the portfolio of the user.</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "stockName": "Apple Inc.", "stockTicker": "AAPL", "noOfShares": 60, "cost": 134.54, "price": 154.65, "pnlinPercentage": 0, "pnlinDollars": 0 }]</pre></div>	No links
404	<div>NOT FOUND - portfolio does not exist</div>	No links

GET	/api/portfolio/{portfolioId}	Get Portfolio Stock	⌵ 🔒

Retrieves the details of a particular stock in the portfolio of the user.

Parameters

Try it out

Name	Description
portfolioId ★ required integer (\$int32) (path)	<div>portfolioId</div>

Responses

Code	Description	Links
200	Successfully retrieved the details of the stock of the user. Media type <div>application/json</div> Controls Accept header. Example Value Schema <pre>{ "stockName": "Apple Inc.", "stockTicker": "AAPL", "noOfShares": 60, "cost": 134.54, "price": 154.65, "pnlinPercentage": 0, "pnlinDollars": 0}</pre>	No links
404	NOT FOUND - stock does not exist in user's portfolio.	No links

Schemas



```
UserAmountDTO {
  description: Data Transfer Object for user to deposit/withdraw funds from the account.

  amount      number($double)
               writeOnly: true
               example: 1000
               Amount specified by the user to deposit or withdraw to/from the account

  balance★    number($double)
               readOnly: true
               example: 2000
               Shows the balance in the user's account.
}
```



```

UserRegisterDTO {
  description:
    Data Transfer Object for register user with username, email and password fields.

  username*
    string
    example: leonlow

    Username for the newly registered user for an account.

  email*
    string($email)
    example: leonlow@email.com

    Email for the newly registered user for an account.

  password
    string
    writeOnly: true
    example: Password0!

    Password for the newly registered user for an account.
}

UserLoginDTO {
  description:
    Data Transfer Object for user login with email and password fields.

  email*
    string($email)
    example: leonlow@email.com

    The user's email address used for authentication

  password*
    string
    example: Password0!

    The user's password used for authentication
}

UserAccessTokenDTO {
  description:
    Data Transfer Object for a user access token, which is returned upon successful authentication.

  accessToken* string
    example:
    eyJhbGciOiJIUzI1NiJ9.eyJpYXQiOiJlbnRvbnQzNDMsImV4cCI6MTY4MjA1MTU0MywidXN1cklkIjoxLCJlbWVpbCI6Imxlb25sb3dAZW1haWwY29tIn0.OLCfu
    L2cfo76_XccZYOF5uwWxPhJA

    Access Token of the authenticated user upon login.
}

WatchlistDTO {
  description:
    Data Transfer Object for the watchlist of the user, which returns the list of stocks added into the
    watchlist by the user.

  userId*
    integer($int64)
    example: 1

    Unique Identifier of the user that is a foreign key to the 'accounts' table.

  stockName*
    string
    example: JPMorgan Chase & Co.

    Name of the stocks inside the watchlist of the user.

  stockTicker*
    string
    example: JPM

    Stock ticker of the stocks inside the watchlist of the user.

  price*
    number($double)
    example: 141.01

    Price of the stock inside the watchlist

  previousDayClose*
    number($double)
    example: 138.76

    Previous day close of the stock inside the watchlist

  get_52WeekHigh*
    number($double)
    example: 167.87

    52 Week High of the stock inside the watchlist

  get_52WeekLow*
    number($double)
    example: 102.32

    52 Week Low of the stock inside the watchlist

  marketCapInBillions*
    number($double)
    example: 123.21

    Market Capitalisation of the business in billions

  annualDividendYield*
    number($double)
    example: 4.23

    Annual dividend yield payout by the business in percentage
}

```

```

OrderSubmitDTO {
  description:      Data Transfer Object for submitting the necessary fields needed to create an order by the user.

  stockTicker*     string
                    example: AAPL

                    Stock ticker of the stock the user wants to buy/sell.

  orderType*       string
                    example: BUY

                    Order type refers to the action which can only be 'BUY' or 'SELL'

                    Enum:
                        Array [ 2 ]
  noOfShares*      integer($int32)
                    example: 60

                    No of shares the user wants to BUY/SELL in the order.

  cost*            number($double)
                    example: 149.86

                    Price of the stock the user is willing to BUY/SELL in the order.
}

StockDataDTO {
  description:      Data Transfer Object used to communicate with Yahoo Finance API to retrieve relevant stock data.

  stockName*       string
                    example: Apple Inc.

                    Stock name retrieved from Yahoo Finance API

  stockTicker*     string
                    example: AAPL

                    Stock ticker used in path parameter to retrieve relevant stock data from Yahoo Finance API

  previousDayClose* number
                    example: 167.63

                    Previous day close of the stock retrieved from Yahoo Finance API

  stockPrice*      number
                    example: 166.65

                    Stock Price of the stock retrieved from Yahoo Finance API

  dayHigh*         number
                    example: 167.87

                    Day High of the stock retrieved from Yahoo Finance API

  dayLow*          number
                    example: 165.56

                    Day Low of the stock retrieved from Yahoo Finance API

  dayChangeInPercent* number
                    example: -0.58

                    Day Change In Percent of the stock retrieved from Yahoo Finance API

  get_52WeekChangeInPercent* number
                    example: -5.39

                    52 Week Change In Percent of the stock retrieved from Yahoo Finance API

  get_52WeekHigh*  number
                    example: 176.15

                    52 Week High of the stock retrieved from Yahoo Finance API

  yearDividendInPercent* number
                    example: 0.54286225

                    Annual Dividend of the stock in percentage retrieved from Yahoo Finance API
}

```

```

PortfolioDTO {
  description: Data Transfer Object of the portfolio which contains all the stocks the user has purchased.

  stockName* string
    example: Apple Inc.
    Name of the stocks inside the portfolio of the user.

  stockTicker* string
    example: AAPL
    Stock ticker of the stocks inside the portfolio of the user.

  noOfShares* integer($int32)
    example: 60
    Number of shares of the stocks inside the portfolio of the user.

  cost* number($double)
    example: 134.54
    Cost of the stock purchased by the user.

  price* number($double)
    example: 154.65
    Current price of the stock in the market

  addedDate string($date-time)
    writeOnly: true
    Added date of the stock in the portfolio.

  updatedDate string($date-time)
    writeOnly: true
    Updated date of the stock in the portfolio.

  pnlinPercentage number($double)
  pnlinDollars number($double)
}

OrderDTO {
  description: Data Transfer Object of the orders associated with the user, returns the history of all the orders.

  orderId* integer($int64)
    example: 1
    Unique identifier of the user which is a foreign key to the 'accounts' table.

  stockTicker* string
    example: AAPL
    Stock ticker of the stock associated with the order.

  orderType* string
    example: BUY
    Order type of the stock associated with the order.

  noOfShares* integer($int32)
    example: 60
    Number of shares transacted in the order for the stock.

  cost* number($double)
    example: 143.21
    Price purchased/sold for each order.

  orderDate* string($date-time)
    Order date for each order.
}

```