SEP_SMFLSpring2021 Wiki

Table of contents

User

Register User

Registering a user to the database.

- **URL**

`/user/register`

- **Method:**

  `POST`

- **URL Params**

  None

- **Data Params**

  `username`: String, user's desired username
  `email`: String, user's email
  `password`: String, user's desired password

- **Success Response:**

  - **Code:** 200
    Content: `{ {savedUser} }`

- **Error Response:**

  - **Code:** 422 UNPROCESSABLE ENTRY
    Content: `{ error : "Invalid email/password" }`

- **Sample Call:**

```
{
    "username": "username123", <br>
    "email": "email123@gmail.com", <br>
    "password": "password123"
}
```

Login User

Authenticates the user and assigns the user a token.

- **URL**

  `/user/login`

- **Method:**

  `POST`

- **URL Params**

  None

- **Data Params**

  `username`: String, username
  `password`: String, password

- **Success Response:**

  - **Code:** 200
    Content: `{ {token} }`

- **Error Response:**

  - **Code:** 401 UNAUTHORIZED
    Content: `{ error : "Incorrect username/password" }`

- **Sample Call:**

```
{
    "username": "username123",
    "password": "password123"
}
```

Get League by User

Gets all leagues that a user is a part of. Returns an empty array if there are no leagues found.

- **URL**

  `/user/:username/league`

- **Method:**

  `GET`

- **URL Params**

  `username`: String, username

- **Success Response:**

  - **Code:** 200
    Content: `{ {list of League objects} }`

- **Error Response:**

  - **Code:** 404 NOT FOUND
    Content: `{ error : "League(s) not found" }`

League

Create League

Allows an authenticated user to create a league and returns league json object. Initializes an empty portfolio and makes user the league manager by default.

- **URL**

  `/league/create`

- **Method:**

  `POST`

- **Headers**

  Bearer token in authorization header. (Ex: Authorization: 'Bearer ' + token)

- **URL Params**

  None

- **Data Params**

  Required

  `leagueName`: String, desired league name
  `settings`: Object, object that contains settings params
  `settings.balance`: Number, balance that everyone in the league starts out with
  `settings.endDate`: Date, date at which the league expires and is disbanded
  `settings.maxPlayers`: Number, max players that are allowed in the league

  Other

  `leagueKey`: String, specifies a key that is essentially used a password to join a league. defaults to empty string.
  `settings.aiPlayer`: String, number of ai players that will be added to the league.
  `settings.public`: Bool, league visibility; whether a league key is required or not. defaults to false. When this is false, `leagueKey` must be set.
  `settings.commissionPercent`: Number, specifies commission percentage which is deducted as part of a transaction. defaults to 0.
  `settings.tradeLimit`: Number, number of trades allowed within a trading day. defaults to 0.

- **Success Response:**

  - **Code:** 200
    Content: `{ {savedLeague} }`

- **Error Response:**

  - **Code:** 422 UNPROCESSABLE ENTRY
    Content: `{ error : "Incorrect params" }`

- **Sample Call:**

```
{
    "leagueName": "My Awesome League",
    "leagueKey": "secretkey123",
    "settings": {
        "balance": 500,
        "aiPlayer": 0,
        "endDate": "2022-04-23T18:25:43",
        "maxPlayers": 5,
    }
}
```

Join League

Allows an authenticated user to join a league and returns league json object. Initializes an empty portfolio with each new user.

- **URL**

  `/league/join`

- **Method:**

  `POST`

- **Headers**

  Bearer token in Authorization header. (See above)

- **URL Params**

  None

- **Data Params**

  Required

  `leagueName`: String, league name
  `leagueKey`: String, associated league key, if applicable, otherwise exclude this param

- **Success Response:**

  - **Code:** 200
    Content: `{ 'Successfully joined league! ({leagueName}) }`

- **Error Response:**

  - **Code:** 404 NOT FOUND
    Content: `{ error: 'League not found'}`

  - **Code:** 422 UNPROCESSABLE ENTRY
    Content: `{ error : 'Cannot join league that you are already a part of! ({leagueName}) }`

    - **Code:** 400 BAD REQUEST
      Content: `{ error : 'Cannot join league since it is full! ({leagueName}) }`

- **Sample Call:**

```
{
    "leagueName": "My Awesome League",
    "leagueKey": "secretkey123"
}
```

Leave League

Allows an authenticated user to leave a league and returns league json object. Removes associated portfolio in said league.

- **URL**

  `/league/leave`

- **Method:**

  `POST`

- **Headers**

  Bearer token in Authorization header. (See above)

- **URL Params**

  None

- **Data Params**

  Required

  `leagueName`: String, league name

- **Success Response:**

    - **Code:** 200
      Content: `{ 'Successfully left league! ({leagueName}) }`

- **Error Response:**

    - **Code:** 404 NOT FOUND
      Content: `{ error: 'League not found'}`

- **Sample Call:**

```
{
    "leagueName": "My Awesome League",
}
```

Disband League

Allows an authenticated league manager to delete a league entirely and returns a confirmation message.

- **URL**

  `/league/disband`

- **Method:**

  `POST`

- **Headers**

  Bearer token in Authorization header. (See above)

- **URL Params**

  None

- **Data Params**

  Required

  `leagueName`: String, league name

- **Success Response:**

  - **Code:** 200
    Content: `{ 'Successfully disbanded league! ({leagueName}) }`

- **Error Response:**

  - **Code:** 404 NOT FOUND
    Content: `{ error: 'League not found'}`

  - **Code:** 401 UNAUTHORIZED
    Content: `{ Cannot disband league. User is not the league manager}`

  - **Code:** 422 UNPROCESSABLE ENTRY
    Content: `{ Cannot disband league cleanly. Unknown Error occurred. {err}'}`

- **Sample Call:**

```
{
    "leagueName": "My Awesome League",
}
```

Get All Leagues

Get all league objects in the database.

- **URL**

  `/league/find/all`

- **Method:**

  `GET`

- **URL Params**

  None

- **Data Params**

  None

- **Success Response:**

  - **Code:** 200
    Content: `{ {all the leagues} }`

- **Error Response:**

  - **Code:** 404 NOT FOUND
    Content: `{ error: 'League not found'}`

Get League Names

Get all league names from the database.

- **URL**

  `/league/find/names`

- **Method:**

  `GET`

- **URL Params**

  None

- **Data Params**

  None

- **Success Response:**

  - **Code:** 200
    Content: `{ {all the league names} }`

- **Error Response:**

  - **Code:** 404 NOT FOUND
    Content: `{ error: 'League not found'}`

Get League by Name

Get a league by specifying a league name. Returns a league object.

- **URL**

  `/league/find/:leagueName`

- **Method:**

  `GET`

- **URL Params**

  `leagueName`: String, league name

- **Success Response:**

  - **Code:** 200
    Content: `{ {league object} }`

- **Error Response:**

  - **Code:** 404 NOT FOUND
    Content: `{ error: 'League not found'}`

Trade

Make Trade

Execute a trade in a league of the user's choice. Returns relevant portfolio.

- **URL**

  `/trade/submit`

- **Method:**

  `POST`

- **URL Params**

  None

- **Data Params**

  None

  Required

`leagueName`: String, league name to perform trade in
`orderType`: String, type of order to execute. For now, you can specify marketBuy and marketSell.
`quantity`: Number, quantity of shares to buy
`tickerSymbol`: String, ticker to specify which share to buy
`expiryDate`: Date, date and time when the outstanding order should expire

- **Success Response:**

  - **Code:** 200
    Content: `{ {portfolio object} }`

- **Error Response:**

  - **Code:** 400 BAD REQUEST
    Content: `{ error: 'Insufficient cash for transaction'}`

  - **Code:** 404 NOT FOUND
    Content: `{ error: 'League not found'}`

- **Sample Call:**

```
{
    "leagueName": "My Awesome League",
    "orderType": "marketBuy",
    "quantity": 1,
    "tickerSymbol": "GME",
    "expiryDate": "2022-04-23T18:25:43"
}
```

Stock Services

Request Current Price

Returns the latest price data available for a ticker

- **URL**

  `/equity/current/:ticker`

- **Method:**

  `GET`

- **URL Params**

  None

- **Data Params**

  None

- **Success Response:**
    - **Code:** 200
      Content: `{ "price": (number) }`
- **Error Response:**
    - **Code:** 400 BAD REQUEST
      Content: `{ "error" : {err} }`
- **Sample Call:**

  `/equity/current/twtr`

```
{
  "price": 65.27
}
```

Request Intraday Price

Returns a ticker's price data for the day (or last available day in case of weekends)

- **URL**

  `/equity/intraday/:ticker`

- **Method:**

  `GET`

- **URL Params**

  None

- **Data Params**

  None

- **Success Response:**
    - **Code:** 200
      Content:

      ```
      {
        {time1}: (number),
        {time2}: (number),
        ...
      }
      ```

- **Error Response:**

    - **Code:** 400 BAD REQUEST
      Content: `{ "error" : {err} }`

- **Sample Call:**

  `/equity/intraday/twtr`

```
{
  "09:30": 65.18,
  "09:31": 65.068,
  "09:32": 64.926,
  "09:33": 65.235,
  "09:34": 65.28,
  "09:35": 65.198,
  ...
}
```

Request Historical Data

Returns the price data for the given ticker and time frame

- **URL**

  `/equity/historical/:ticker?timeframe={timeframe}`

  timeframe must be in `['5d', '1m', '6m', 'ytd', '1y', '5y']`

- **Method:**

  `GET`

- **URL Params**

  None

- **Data Params**

  None

- **Success Response:**

    - **Code:** 200
      Content:

      ```
      {
        {time1}: (number),
        {time2}: (number),
        ...
      }
      ```

- **Error Response:**

  - **Code:** 400 BAD REQUEST
    Content: `{ Invalid timeframe }`

  - **Code:** 400 BAD REQUEST
    Content: `{ "error" : {err} }`

- **Sample Call:**

  `/equity/historical/twtr?timeframe=1m`

```
{
  "2021-02-23": 73.17,
  "2021-02-24": 71.92,
  "2021-02-25": 74.59,
  "2021-02-26": 77.06,
  "2021-03-01": 77.63,
  "2021-03-02": 73.67,
  "2021-03-03": 70.86,
  "2021-03-04": 66.75,
  "2021-03-05": 66.95,
  "2021-03-08": 63.48,
  "2021-03-09": 67.52,
  "2021-03-10": 64.83,
  "2021-03-11": 68.46,
  "2021-03-12": 68.1,
  "2021-03-15": 70.28,
  "2021-03-16": 69.25,
  "2021-03-17": 70.2,
  "2021-03-18": 66.72,
  "2021-03-19": 66.23,
  "2021-03-22": 65.21
}
```

Request Ticker Statistics

Returns statistics for an equity

- **URL**

  `/equity/statistics/:ticker`

- **Method:**

  `GET`

- **URL Params**

  None

- **Data Params**

None

- **Success Response:**

  - **Code:** 200
    Content:
    (Currently includes other extraneous fields, just ignore for now)

    ```
    {
      "peRatio":         (number),
      "dividend":        (number),
      "tickerSymbol":    (string),
      "__v":             (number),
      "askPrice":        (number),
      "avgVolume":       (number),
      "beta":            (number),
      "bidPrice":        (number),
      "currentPrice":    (number),
      "dayHigh":         (number),
      "dayLow":          (number),
      "earningsDate":    (date),
      "eps":             (number),
      "equityName":      (string),
      "exDividend":      (number),
      "marketCap":       (number),
      "openPrice":       (number),
      "percentChange":   (number),
      "previousClose":   (number),
      "time":            (date),
      "volume":          (number),
      "week52High":      (number),
      "week52Low":       (number)
    }
    ```

- **Error Response:**

  - **Code:** 400 BAD REQUEST
    Content: `{ "error" : {err} }`

- **Sample Call:**

  `/equity/statistics/twtr`

```
{
  "peRatio": -45.83157108280366,
  "dividend": 0,
  "_id": "6058085f8ecf8b81f934319a",
  "tickerSymbol": "twtr",
  "__v": 0,
  "askPrice": 81,
```

```
    "avgVolume": 22957837,
    "beta": 1.236305599208382,
    "bidPrice": 65.01,
    "currentPrice": 65.07,
    "dayHigh": null,
    "dayLow": null,
    "earningsDate": "2021-02-09T00:00:00.000Z",
    "eps": -1.44,
    "equityName": "Twitter Inc",
    "exDividend": null,
    "marketCap": 52047523742,
    "openPrice": null,
    "percentChange": -0.00215,
    "previousClose": 65.21,
    "time": "2021-03-23T15:51:21.891Z",
    "volume": 218993,
    "week52High": 80.75,
    "week52Low": 22.36
  }
```

Validate Ticker

Returns whether or not a ticker is a valid tracked ticker

- **URL**

  `/equity/tickers/:ticker`

- **Method:**

  `GET`

- **URL Params**

  None

- **Data Params**

  None

- **Success Response:**

  - **Code:** 200
    Content: `{ "exists": (boolean) }`

- **Error Response:**

  - **Code:** 503 SERVICE UNAVAILABLE
    Content: `{ "error" : "Unable to access database" }`

- **Sample Call:**

  `/equity/tickers/twtr`

```
{
  "exists": true
}
```