ECE 568 Final Project Web Interface Document

Group # 7

Quick Demo

Suppose website is on http://rurich.zwithc.cn. A valid API call looks like:
http://rurich.zwithc.cn/api/v0.1.0/stockresource/GOOG?type=high
which will return GOOG's highest price in the last ten days. And the return looks like:
callback({"symbol":"GOOG","value":{"max":"1094.1600"}}),
where callback() is used for JSONP to solve the JS cross domain request.

The APIs in Stock Information and Comment are on http://rurich.zwithc.cn while APIs in Pridiction Engine are hosted on http://zwithc.cn:5001/.

Stock Information

Get Lastest Prices

URL: /api/v0.1.0/stockresource
HTTP Method: GET
Description:
Get the list of all companies in the database along with their latest stock price
Return: callback(JSON)
callback([{"price":1039.7800,"symbol":"GOOG","volume":0},{"price":69.7800,"symbol":"AABA","volume":0},{"price":175.2700,"symbol":"FB","volume":0},{"price":94.6200,"symbol":"MSFT","volume":0},{"price":30.4100,"symbol":"TWTR","volume":0}, …])

Get Stock Price

URL: /api/v0.1.0/stockresource/*symbol*
HTTP Method: GET
Example: GET /api/v0.1.0/stockresource/*GOOG*
Description:
Get the year/month/day-wide stock price for plotting the figure
Parameters:

    type: *year*, *month* or *day*

Return: callback(JSON)
callback({"symbol":"GOOG","data":{"volume":["2558385","3029471","3369275","2726830","2680400","2275076","2484651", …],"price":["1053.2100","1005.1000","1004.5600","1031.7900","1006.4700","1013.4100","1025.1400", … "],"time":["Mon Mar 26 00:00:00 UTC 2018","Tue Mar 27 00:00:00 UTC 2018", …]}})

Get highest, average or lowest prices

URL: /api/v0.1.0/stockresource/*symbol*
HTTP Method: GET
Example: GET /api/v0.1.0/stockresource/*GOOG*?type=*high*
Description:
Get the highest stock price of any company in the last ten days.
Get the average stock price of any company in the latest one year.
Get the lowest stock price of any company in the latest one year.
Parameters:
    type: *high*, *avg* or *low*
Return: callback(JSON)

```
callback({
      "symbol":"GOOG",
      "value":{
            "max":"1094.1600"
            }
})
```

URL: /api/v0.1.0/stockresource/*symbol*?type=loweravg
HTTP Method: GET
Example: GET /api/v0.1.0/stockresource/*GOOG*?type=loweravg
Description:
Get the list of companies who have the average stock price lesser than the lowest of any of the selected company in the latest one year.
Return: callback(JSON)

```
callback({
      "Stock List":[
            "MSFT","AABA","AAPL","BAC","TWTR","JPM","JNJ","FB"
            ]
})
```

Comment

Post comment
URL: /api/v0.1.0/stockresource/comment
HTTP Method: POST
Description:
Post the comment with the username, stock symbol and the timestamp into the database

Get comment
URL Method: /api/v0.1.0/stockresource/*symbol*?type=comment
HTTP: GET
Example: GET /api/v0.1.0/stockresource/*GOOG*?type=comment
Description:
Get the latest 5 comments of this stock
Return: callback(JSON)

```
callback({
      "comment":[{
                  "symbol":"GOOG",
                  "comment":"AABBCC",
                  "timestamp":"2018-04-25T21:35:35.546Z",
                  "username":"Test4"
            },
            {
                  "symbol":"GOOG",
                  "comment":"I Want To Fall In Love AQUARIUM",
                  "timestamp":"2018-04-21T21:16:20.567Z",
                  "username":"Aqours"
            }]
})
```

Prediction Engine

Return Error Code:

100: Missing parameters
101: Invalid parameters content
102: Invalid parameters type
103: Invalid Symbol
104: Not on the trading day

For example, api/v0.1.0/vr?symbol=GOOG will return

```
{
  "error": {
    "errorCode": 100,
    "errorInfo": "Missing parameters",
    "missingParameters": [
      "timestamp"
    ]
  },
  "time": "2018-04-27T08:32:29.551326+00:00",
  "type": "error"
}
```

Price Predictor

URL: /api/v0.1.0/predict
HTTP Method: GET
Example: GET /api/v0.1.0/predict?symbol=*GOOG*&term=*short*&timestamp=*2018-04-20*
Parameters:

    symbol: the name of the stock like GOOG
    term: *short* or *Long*
    timestamp: ISO 8601 timestamp, e.g. 2018-04-20 or 2018-04-20T06:40:39.643098%2B00:00. “+” should be encoded by “%2B” in URL.

Description:
    Get the next day's predict price of *symbol*. Short term prediction will use the latest 50 prices while Long term uses the latest 252 prices (number of trading days in a year). Using Bayes, SVR and DNN models at the same time, return three prices together.
Return: JSON
```
{
  "result": {
    "note": "ONLY FOR TESTING!",
    "predictPrice": 1084.548221444521,
    "predictor": [
      {
        "name": "bayes",
        "price": 1095.6156217837324
      },
      {
        "name": "Support Vector Regression",
        "price": 1083.2693684775647
      },
      {
        "name": "Deep Neural Network",
        "price": 1074.7596740722656
      }
    ],
    "symbol": "GOOG",
    "timestamp": "2018-04-20T06:40:39.643098+00:00"
  },
  "time": "2018-04-27T07:49:30.637156+00:00",
  "type": "result"
}
```

Indicators - VR
URL: /api/v0.1.0/vr
HTTP Method: GET
Example: GET /api/v0.1.0/vr?symbol=*GOOG*&timestamp=*2018-04-20*
Parameters:
    symbol: the name of the stock like GOOG
    timestamp: ISO 8601 timestamp, e.g. 2018-04-20 or 2018-04-20T06:40:39.643098%2B00:00. "+" should be encoded by "%2B" in URL.
    period: [optional, default = 24] Determine the period of VR, e.g. if period = 24, it will calculate the value of 25-day (24+1 day) VR.
Description:
    Calculate the VR indicator value of *symbol* at *timestamp* of the last *period* + 1 days.
Return: JSON

```
{
  "result": {
    "data": 75.07711536845262,
    "indicator": "VR",
    "symbol": "GOOG",
    "timestamp": "2018-04-20T06:40:39.643098+00:00"
  },
  "time": "2018-04-27T07:55:21.920108+00:00",
  "type": "result"
}
```

Indicators - EMA
URL: /api/v0.1.0/ema
HTTP Method: GET
Example: GET /api/v0.1.0/ema?symbol=*GOOG*&timestamp=*2018-04-20*
Parameters:
    symbol: the name of the stock like GOOG
    timestamp: ISO 8601 timestamp, e.g. 2018-04-20 or 2018-04-20T06:40:39.643098%2B00:00. “+” should be encoded by “%2B” in URL.
    period: [optional, default = 10, must > 10] Determine the period of VR, e.g. if period = 10, it will calculate the value of 11-day (10+1 day) VR.
Description:
    Calculate the EMA indicator value of *symbol* at *timestamp* of the last *period* + 1 days.
Return: JSON

```
{
  "result": {
    "data": 1040.7800000000002,
    "indicator": "EMA",
    "symbol": "GOOG",
    "timestamp": "2018-04-20T00:00:00+00:00"
  },
  "time": "2018-04-27T07:59:22.547874+00:00",
  "type": "result"
}
```

Indicators - MACD
URL: /api/v0.1.0/macd
HTTP Method: GET
Example: GET /api/v0.1.0/macd?symbol=*GOOG*&timestamp=*2018-04-20*
Parameters:
    symbol: the name of the stock like GOOG

timestamp: ISO 8601 timestamp, e.g. 2018-04-20 or 2018-04-20T06:40:39.643098%2B00:00. “+” should be encoded by “%2B” in URL.

Description:

Calculate the MACD indicator value of *symbol* at *timestamp*.

Return: JSON

```
{
  "result": {
    "data": -8.215950644495251,
    "indicator": "MACD",
    "symbol": "GOOG",
    "timestamp": "2018-04-20T00:00:00+00:00"
  },
  "time": "2018-04-27T08:04:09.669459+00:00",
  "type": "result"
}
```