

Application Manual – Stock Price Monitor

The application is divided into 3 modules

- 1) A centralized database which contains Company and Stock History Tables
- 2) A Web module which provides end points to add, delete companies and view Company and Stock History Information
- 3) A Batch job that will load the company stock information to the database every 5 minutes

Database – MySQL

Company Table – Holds the company information

Field	Type	Null	Key	Default
symbol	varchar(255)	NO	PRI	NULL
created	datetime	NO		NULL
name	varchar(255)	NO		NULL

Symbol - The symbol of the company

Created - The date the company is added to the database

Name - The name of the company

Stock History Table – Holds the stock information that is loaded every 5 minutes

Field	Type	Null	Key
symbol	varchar(255)	NO	PRI
time	datetime	NO	PRI
stock_price	double	NO	

Symbol - The symbol of the company which is also the foreign key reference to the 'symbol' column in Company Table

Time - The time the entry was added to the database

Stock_Price - The stock price of a company

Web API

The Web API provides a REST interface to perform CRUD operations on the companies and stock price history

Note – The application will be deployed at /StockPriceMonitor.

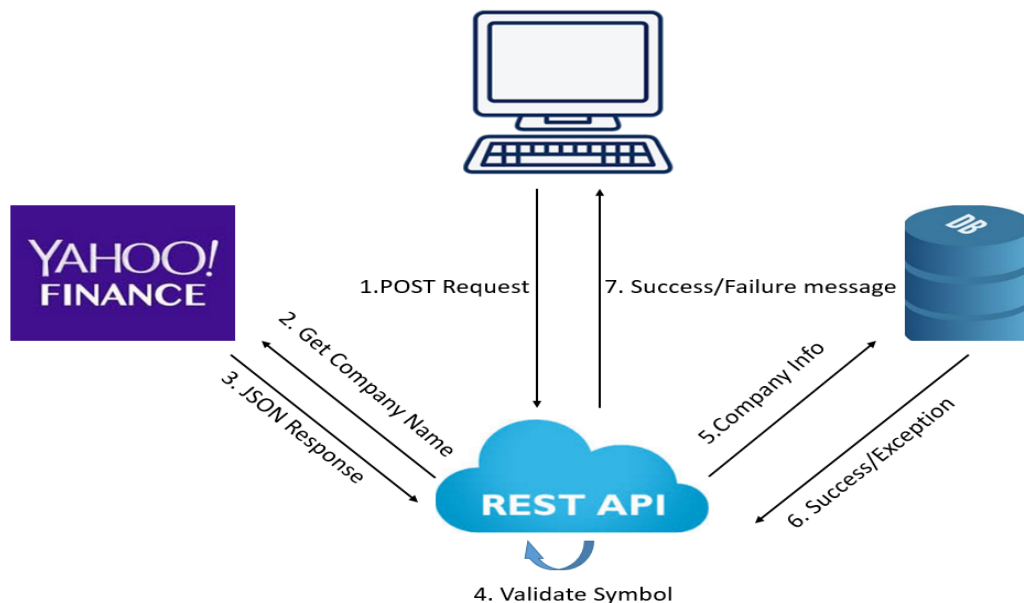
Example end point - <http://localhost:8080/StockPriceMonitor/rest/companies/>

Services

Add a company

End Point - /rest/companies/{company-symbol}

Request Type - POST



The user will make a Http POST request with the above URL, the REST API will then try to get the company information from Yahoo Finance. If the company symbol is invalid an `InvalidSymbolException` is thrown, otherwise the company information is persisted.

Responses:

HTTP 201, if the information was added successfully

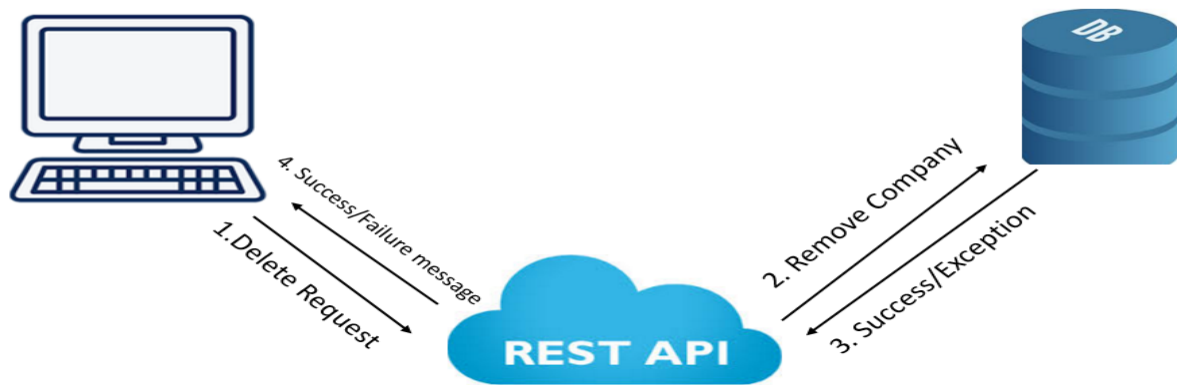
HTTP 400, if the symbol was invalid

HTTP 500, if there was an exception while retrieving the data from Yahoo Finance or while persisting the company information

Delete Company

End Point - /rest/companies/{company-symbol}

Request Type - DELETE



The user will make a delete request for the above URL and send the company symbol as the path parameter. The REST API will try to delete this company from the database.

Responses:

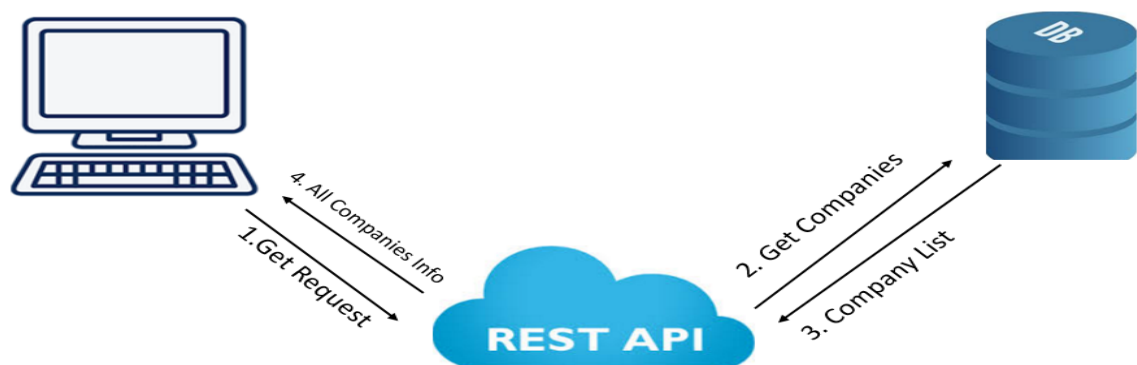
HTTP 200, if the information was deleted successfully

HTTP 500, if there was an exception while deleting the company information

Get All Companies

End Point - /rest/companies

Request Type – GET



Responses:

HTTP 200, The list of all companies and their information

HTTP 500, if there was an exception while retrieving the company information

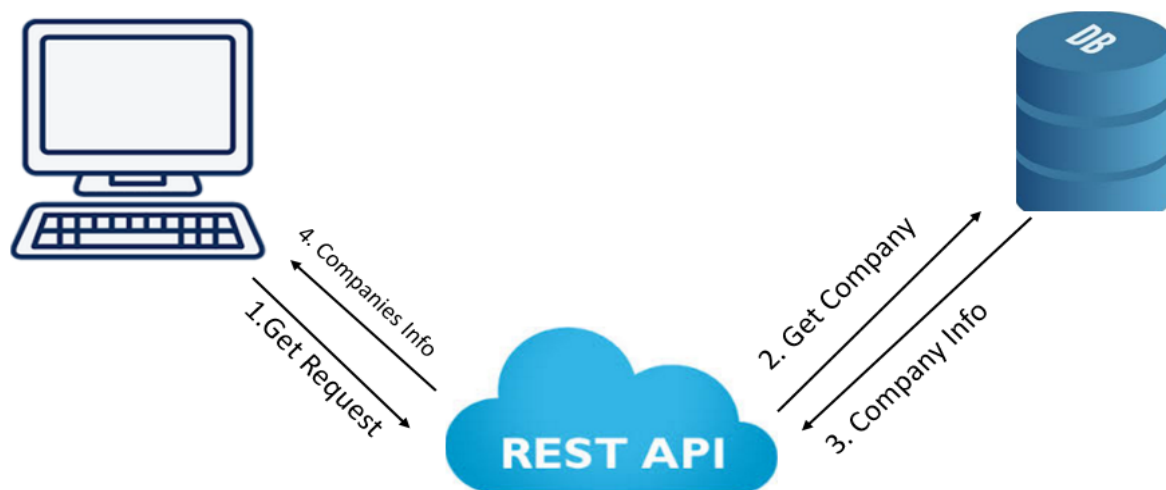
Example –

```
[
  {
    "symbol": "appl",
    "name": "APPELL PETE CORP",
    "created": "2016-11-23 50:50"
  },
  {
    "symbol": "ppl",
    "name": "PayPal Inc",
    "created": "2016-11-22 04:11"
  },
  {
    "symbol": "yhoo",
    "name": "Yahoo Inc",
    "created": "2016-11-22 03:57"
  }
]
```

Get Company Info

End Point - /rest/companies/{company-symbol}

Request Type – GET



Responses:

HTTP 200, The company information

HTTP 500, if there was an exception while retrieving the company information

Example – /rest/companies/yhoo

```
{  
  "symbol": "yhoo",  
  "name": "Yahoo Inc",  
  "created": "2016-11-22 03:57"  
}
```

Company and Latest Stock Price

End Point - /rest/stocks

Request Type – GET



Responses:

HTTP 200, The company information and the last updated stock price of the company

HTTP 500, if there was an exception while retrieving the stock information

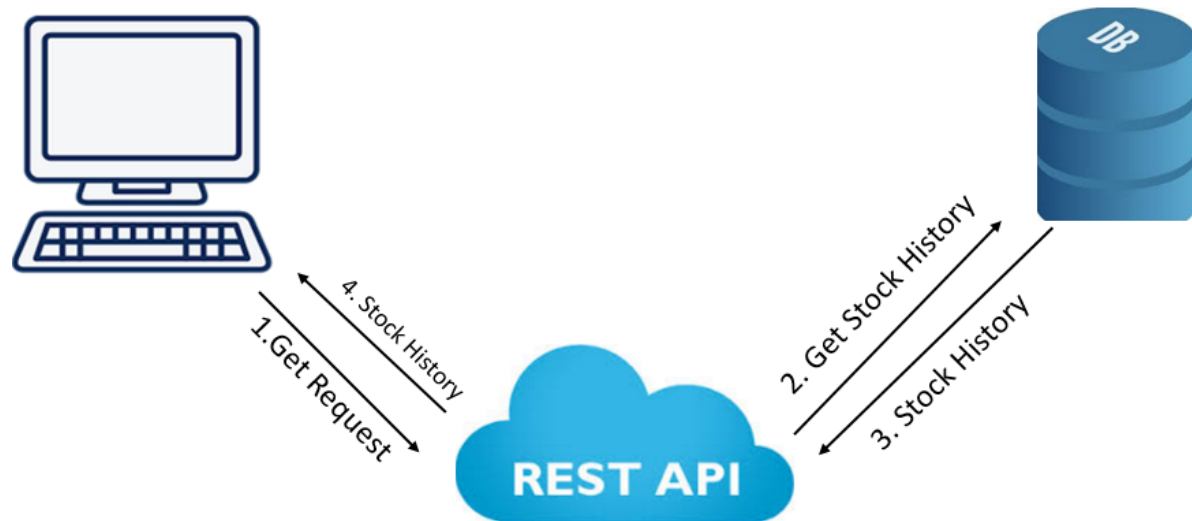
Example – /rest/stocks/

```
[  
  {  
    "name": "APPELL PETE CORP",  
    "symbol": "appl",  
    "last_updated_price": 0  
  },  
  {  
    "name": "PayPal Inc",  
    "symbol": "ppl",  
    "last_updated_price": 32.81  
  },  
  {  
    "name": "Yahoo Inc",  
    "symbol": "yhoo",  
    "last_updated_price": 40.96  
  }  
]
```

Company Stock History

End Point - /rest/stocks/{company-symbol}

Request Type – GET



Responses:

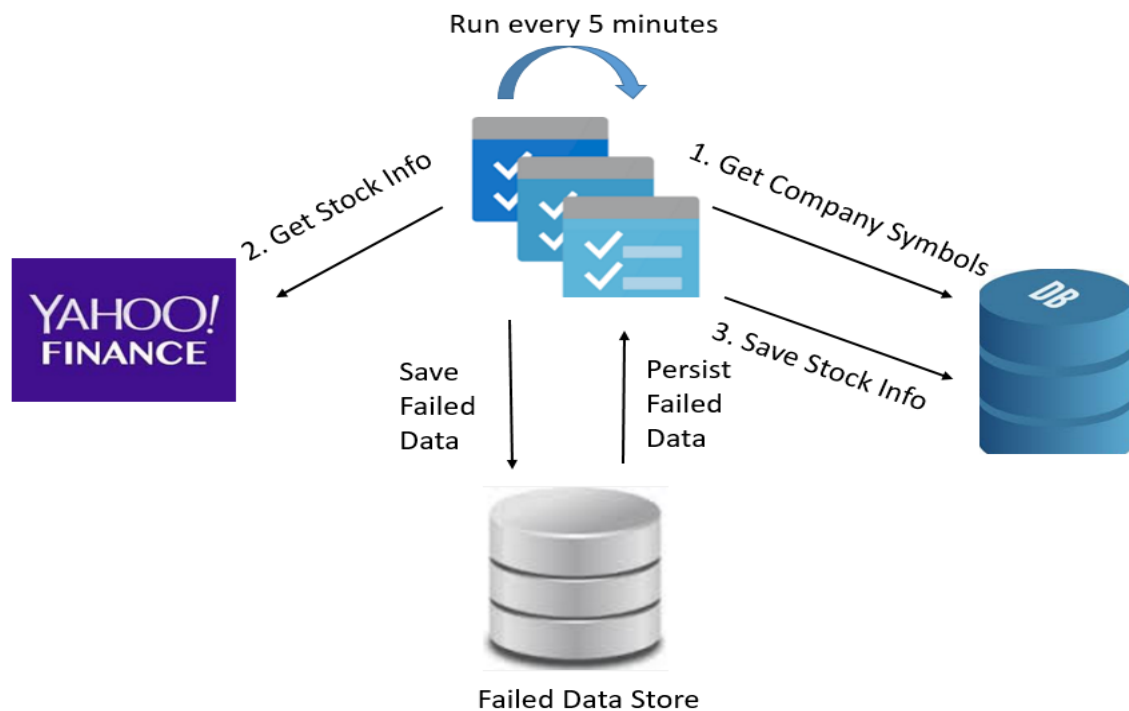
HTTP 200, The company's historical stock information

HTTP 500, if there was an exception while retrieving the stock information

Example – /rest/stocks/yhoo

```
▼ {
  "symbol": "yhoo",
  ▼ "stockHistory": [
    ▼ {
      "time": "2016-11-23 20:01",
      "stockPrice": 40.88
    },
    ▼ {
      "time": "2016-11-23 25:00",
      "stockPrice": 40.865
    },
    ▼ {
      "time": "2016-11-23 30:00",
      "stockPrice": 40.845
    },
    ▼ {
      "time": "2016-11-23 35:00",
      "stockPrice": 40.865
    },
    ▼ {
      "time": "2016-11-23 40:00",
      "stockPrice": 40.88
    },
  ],
}
```

Batch Job



The batch job is scheduled to run every 5 minutes during the week days from 9 am to 4 pm, since this is the time the stock market is open.

Every 5 minutes the batch job starts a task which will first collect the list of company symbols that are monitored from the database. Then, the latest stock price for all these companies is obtained from the yahoo finance provider and is persisted in the database.

While saving the data, if any record fails due to some connection issues, we save this data in an internal data store which is a data structure in the application. This data is later persisted in the subsequent tasks.