



Course Exercise

Angular + .NET

version 1.0

Table of Contents

Scope.....	2
In General.....	2
Mockups.....	2
Functionality	3
Dashboard.....	3
Symbols.....	4
API.....	4
Database	5
Users	5
Symbols.....	5
Watchlists.....	5

Scope

The scope of this document is to describe a sample application that can be developed during the “Angular - .NET” course. During the development, each team should use best practices learned in the course.

In General

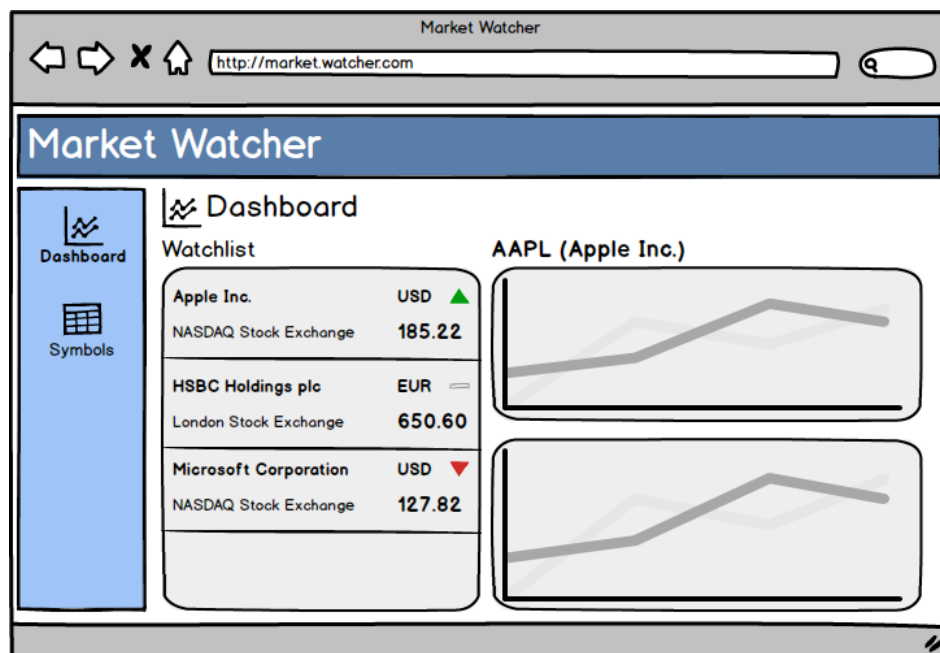
The application will include a **back-end** server and a **front** application. An external API will be also used to allow access to financial information. This API is available to the following link:

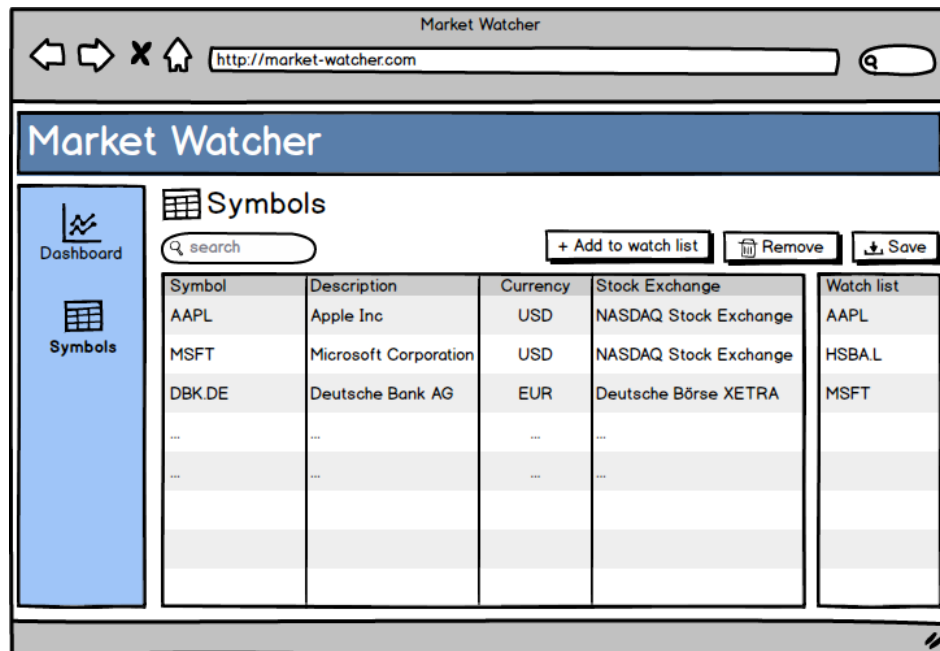
<https://www.worldtradingdata.com>

You can subscribe and use that API to collect financial information described below.

Mockups

The following mock-ups shown how the application might look. The team is free to provide its own layout, but it is important to cover the required functionality.





Functionality

The application starts with a login page (**not shown** in the mock-ups) and after a success login it will display the dashboard. The main layout consists of the **header**, the **side menu bar** and the **routing area** which changes according to selected menu item.

2 different pages should be available and selectable from the side menu bar:

1. The **Dashboard** and
2. The **Symbols**

The selected menu item is shown as a **title** on the top of the routing page.

Dashboard

The dashboard will have 3 different components: one **watchlist** and two **charts**.

The **watchlist** will be per user which means that after the login, the front needs to request from the server the symbols that will be shown in it. The list will allow up to **5** symbols. Each item in the watch list will display the following information:

- **Description**
- **Stock Exchange**
- **Currency**
- **Price** (this will be retrieved from the external API)
- An arrow icon that reveals if the symbol price is better (*up arrow with green color*), worst (*down arrow with red color*) or equal (*horizontal line in gray*) with its previous price.

In order to compare prices and define change, use fields **price** and **close_yesterday** returned from the API.

When selecting (*clicking*) a symbol in the list, then **2** different calls will be sent to the server. The 1st will return the intraday data of that symbol and the 2nd the historical data. The data returned from the calls will be plotted in the 2 charts next to the watch list, on top the intraday chart and then the historical chart. By default, the first symbol in the list will be selected and plotted.

Symbols

The Symbols page will have 2 tables: **one** for the symbols and another **one** for the symbols found in the user watch list.

- By selecting a symbol from the **symbols list**, the user will be able to add it in the watch list. If the watch list is full (5 symbols) then a message should appear to notify him/her.
- By selecting a symbol from the **watch list** the user will be able to remove it from the list.
- By pressing the Save button, the list will be saved for that user.

A search control will allow **filtering the symbols** in the list. Filter will be applied in all fields.

API

External API

The external API (www.worldtradingdata.com) will help to collect financial data. After subscription, an API key will be available which should be used as a query string in each call.

The required calls on that API are:

Fill watch list with prices

```
https://api.worldtradingdata.com/api/v1/stock?symbol=AAPL,MSFT,HSBA.L
&api_token=123456789
```

Use the **price** field and the **close_yesterday** for comparison to draw the right arrow.

Collect intraday data for a symbol

```
https://intraday.worldtradingdata.com/api/v1/intraday?symbol=AAPL&range=1&interval=1&
api_token=123456789
```

Please note that you need to plot a **line chart** using the “**close**” field returned in the data. **Dates** will be used in the x-Axis and values in the y-Axis.

Collect historical data for a symbol

```
https://api.worldtradingdata.com/api/v1/history?symbol=AAPL&sort=newest&api_t
oken=123456789
```

Please note that you need to plot a **line chart** using the “close” field returned in the data. Dates will be used in the x-Axis and values in the y-Axis.

Server API

The API that need to be developed in the .NET server includes the following:

Login

You need to implement an API call to authenticate your user. The id of the authorized user will be used to manage the watch list.

Watch list

Get user watch list

You need to implement an API call to retrieve the symbols of a user watch list. Only one watch list is required in this sample application. Keep in mind that the list may be empty.

Save user watch list

You need to implement an API call to save the symbols in a user watch list. Keep in mind that the list may be empty.

Database

For this course we need to have a small database (or 3 files) to store our data. The required tables are described below.

Users

The users should have the ability for self-registration & remind password.

Symbols

For this application we need to keep a table with all available symbols. An excel file is provided with those data. Management of symbols is not required.

Symbol	Description	Currency	Stock Exchange Long
AAPL	Apple Inc.	USD	NASDAQ Stock Exchange
MSFT	Microsoft Corporation	USD	NASDAQ Stock Exchange

Watchlists

Consider adding some symbols for each available user in order to initially have data for those after the login.

UserId	Symbol
--------	--------

1	AAPL
1	MSFT
1	HSBA.L
2	AAPL
2	MSFT
2	HSBA.L