

Simple Nancy Web Api Project

Goal

The purpose of this exercise is check the candidate's ability to create a standalone RESTful API web application using the Nancy framework for .Net.

Acceptance criteria

We expect a Visual Studio 2017 solution with the requested features that will be developed and sent back as a zip file. The final result will be analysed with the following criteria in mind:

Architecture and design. We expect the candidate to make the decision on the necessary components architecture which follows c#/ .net best practice recommendations.

Code. Clean and readable code style is important part of this exercise.

Unit testing. The candidate should provide unit tests where appropriate.

Attention to detail. We expect that all features listed in the description section are delivered and any side effects are thought through and tested.

Creativity and Research ability. You are expected in relatively short period of time to research and use a fully-functioning API framework to customise the standard functionality in most efficient way. Nice styling is not required, but will be a bonus.

Assignment

You will be creating a Visual Studio solution using the Nancy API framework for .Net. This framework allows you to quickly develop a restful api which can process http web requests and respond with json data. The data will be supplied by the New York Times newspaper web api.

Creating the solution

Visual Studio 2017

New Project

- Visual C#

- Web

- ASP.NET Core Web Application Project

First tab - .NET Framework

Second tab - ASP.NET Core 2.1

Use "Empty" template

You will need the following nuget packages
Microsoft.AspNetCore
Microsoft.AspNetCore.Owin
Nancy

Some help on initial setup of NancyApi
<https://www.excella.com/insights/getting-started-nancyfx-and-asp-net-core>
<http://aartekservices.com/blogs/development/how-to-create-nancy-api/>

Data

Please use the "top stories" api made available by the New York Times.

You can use the following apikey - c4378f88eeea41b7859dba3bc44aab66

https://developer.nytimes.com/top_stories_v2.json

<https://api.nytimes.com/svc/topstories/v2/{section}.json?apikey=c4378f88eeea41b7859dba3bc44aab66>

eg. for the "home" section articles you would use...

<https://api.nytimes.com/svc/topstories/v2/home.json?apikey=c4378f88eeea41b7859dba3bc44aab66>

Tasks

Create a "Module" class - inheriting from NancyModule and implement the api methods in the list below.

Note - each method should respond with either a list or a single instance of the following ViewModel classes in a json format.

```
public class ArticleView
{
    public string Heading { get; set; }

    public DateTime Updated { get; set; }

    public string Link { get; set; }
}

public class ArticleGroupByDateView
{
    public string Date { get; set; }

    public int Total { get; set; }
}
```

Api Requests/Methods

1. Get[""] - this should return a simple json string indicating that the api is functional.
2. Get["/list/{section}"] - this should return an array of json representing a list of articles filtered by a "section" parameter. Example values for the "section" parameter might be "home", "world", "opinion" etc. This value should be used directly in the ny times api request url. The list should be a list of a strongly typed object/class/model named ArticleView (see above). ie. you will have to transform the results of your query to the nytimes api into a list of ArticleView objects.
3. Get["/list/{section}/first"] - get the first item from the previous list
4. Get["/list/{section}/{updatedAt}"] - get the list from point 2 and then filtered by "updatedAt". This is a date string with the format "yyyy-MM-dd".
5. Get["/article/{shortUrl}"] - get a single ArticleView object from the "home" section filtered by "shortUrl". Ideally the "shortUrl" should be in the format "XXXXXXX".
6. Get["/group/{section}"] - this should return a list of a different type – ArticleGroupByDateView (see above). This is basically a list of dates with the total number of articles contained for each date. ie. If the {section} contains 10 articles across 2 different dates then your list will contain 2 items.

Api Response

The json response to an API request for a list should look something like the example below (for 2 items). A request for a single article should obviously just contain the json for 1 item.

I would recommend that you use a tool such as Postman to check the application http requests and responses.

Please make sure that the appropriate json responses are available both within the browser environment (Chrome is fine) as well as external tools such as Postman.

eg. for ArticleView list responses

```
[{"heading":"December Jobs Report Highlights Economy's Strength Despite Market Tumult","updated":"2019-01-04T14:48:58.0000000+00:00","link":"https://nyti.ms/2RwYf01"}, {"heading":"This Expert Called the Market Plunge. Here's What He Sees in 2019.","updated":"2019-01-04T10:00:09.0000000+00:00","link":"https://nyti.ms/2Ruoe8f"}, {"heading":"Stocks Rise as Jobs Report Adds Fuel to Global Rebound","updated":"2019-01-04T14:33:38.0000000+00:00","link":"https://nyti.ms/2RxCtcz"}, {"heading":"McConnell Faces Pressure From Republicans to Stop Avoiding Shutdown Fight","updated":"2019-01-04T04:04:16.0000000+00:00","link":"https://nyti.ms/2Ru66vf"}, {"heading":"Government Shutdown Leaves Workers Reeling: 'We Seem to Be Pawns'","updated":"2019-01-04T01:03:20.0000000+00:00","link":"https://nyti.ms/2GTJnVy"}]
```

eg. for ArticleGroupByDateView list responses

```
[{"date":"2019-01-04","total":34}, {"date":"2019-01-03","total":7}, {"date":"2019-01-02","total":4}, {"date":"2018-12-07","total":1}]
```