AdviceSlip Middleman API

Design Doc.

Paweł Sauer

Overview

This project has been started an exercise in ASP.NET API data requisition and provision among other things. Apart from serving educational reasons, it aims to provide the same data as the API it addresses (Advice Slip API), yet in a more elegant and standardized manner.

Objective

The objective of the project is twofold: to acquire new knowledge, and improve what could be seen as imperfections in the data source API.

Features:

As required by the Simple API Test:

- Use Git to keep track of your project
  - Git has been used as the projects version control, with the remote repo hosted on GitHub.

- Use the Advice Slip API to display a random advice slip on a simple HTML page
  - Two sample HTML pages have been created to fit this requirement. The Index page displays a random advice slip, as well as the Search page that allows the user to search for advice slips.
  - Both pages call the locally hosted Middleman API, not the original Advice Slip API.

- Use the Advice Slip API to add the ability to search for advice slips and display them in a list
  - Search utilises the Advice Slip API's search function to provide results. These however are displayed in a similar way to other JSON produced by the Middleman API
  - Search is available both on the Search webpage that is part of the project as well as the API itself at /api/search/{query}

- The Advice Slip API should not be called from the HTML page. Integrate the API into your own custom API/MVC project. Your HTML page should call your own API which in turn calls the Advice Slip API
  - Both HTML pages call the Middleman API using js, not the original Advice Slip API.

- Your project should handle API errors gracefully
  - Error messages from the original api are returned analogically to the original API, however in a form that is common in the entire Middleman API

- Your project should log any errors and any information that might be useful to debug said errors.
    - Logging has been introduced and uses various message levels to ease filtering data.

- Provide a sample HTTP Request directly towards your API
    - Calling GET on https://localhost:5001/api/ will yield a random advice slip
        - Example call: curl https://localhost:5001/api/

Bonus features:

- Implement an authentication mechanism into your API
    - Will be implemented at a later date

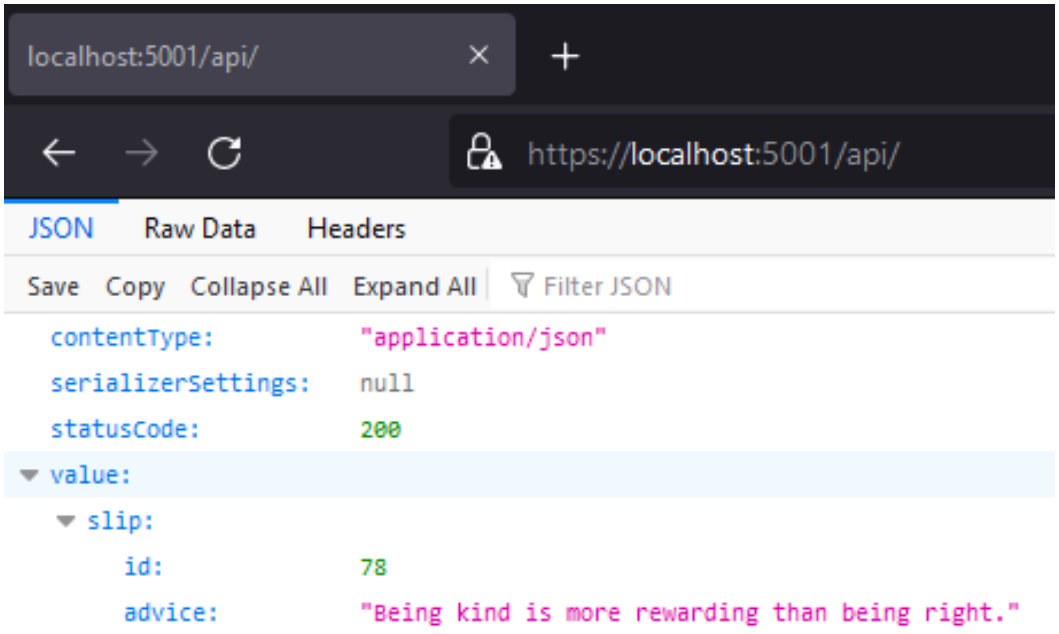
- Add API documentation to enable third-parties to integrate your API
    - API documentation consists of a readme.md file and a PDF design doc (this file)

- Use an SQL database to store an audit of all Advice Slips requested. Add the ability to list all audit records in your HTML page
    - Will be implemented at a later date

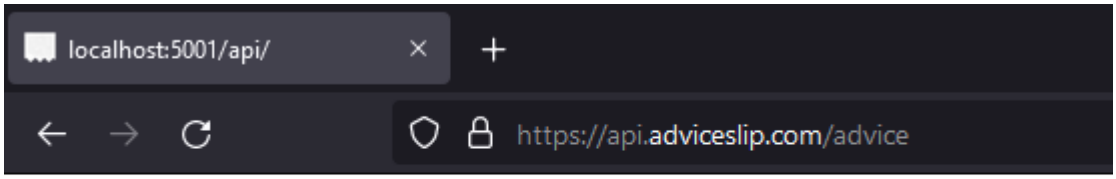
- Add automated tests to your project
    - Unit/Integration tests have been added to the project. Moq has been used to supplement some tests.

Additions/Changes:

The way data from the original API has been changed compared to the original API. This is best seen when using a web browser:



Middleman API



Advice Slip API

The software also attempts to fix JSONs from the original API that were missing a trailing closing bracket. This seems to have been fixed in the original API.

Apart from the required random slip and search functionality, the Middleman API also supports getting data via slip ID at /api/{ID}

Further plans:

Further plans include implementation of the two missing extra features (history and authentication) but also:

- Code refactoring
- Traffic control
- Extended automated testing
- Extended logging