**Authors:** Justin Hathaway, Chloe Kuhar, Paige McKinney, Navreen Kaur

**Goal:**

This document outlines the different Moolah Financial Web APIs that are currently implemented in the C# backend. Information on the APIs will include which APIs we plan to implement (what features we want them to deliver), tools that are useful for API development, ways to invoke the different APIs, and example request and response data for each API. Feel free to make edits to this document to keep it up to date.

**APIs that we need to implement:**

Note: If you think of a specific API that we will need for our application, list it here (it terms of what functionality it needs to deliver).

* Top 2 Recommended Portfolios for a user
  + Display the top two portfolios that share the most tags with a given user

**Useful Tools:**

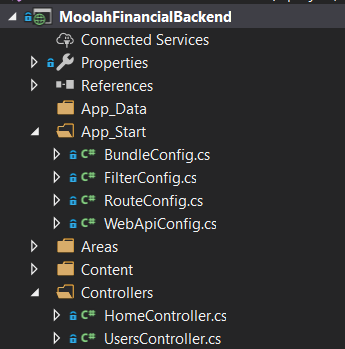
* Postman
  + Website to download tool: <https://www.getpostman.com/>
  + Allows you to enter a URL and a JSON request. It then outputs the JSON response that the API returns (useful for debugging & testing the C# backend).

**Calling APIs:**

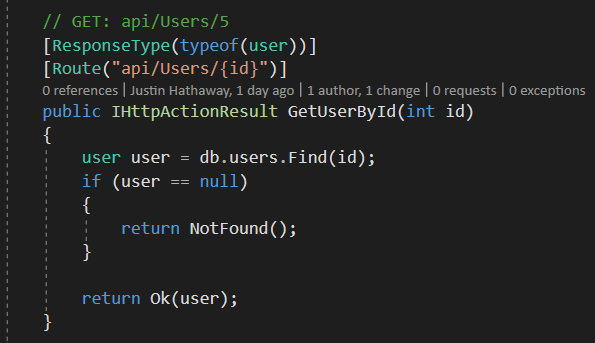
Base URL: <http://moolah-financial-api.azurewebsites.net/>

To call an API, we need to access the URL that invokes the API that we want to call while also passing in the parameters the API method needs to execute. All of our APIs are running on this site: <http://moolah-financial-api.azurewebsites.net/>. This site acts as the base URL from which we will call all our APIs (“http://moolah-financial-api.azurewebsites.net/” + [the specific URL for an API]. There are three different ways of finding a specific URL for an API:

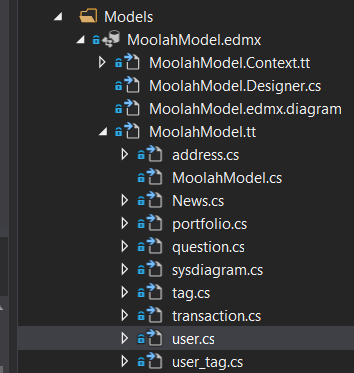
1. This document (if this is up to date)
2. Open the base URL (<http://moolah-financial-api.azurewebsites.net/>) and selecting the ‘API’ tab at the top of the page. This will display all of the APIs in their respective controllers. The left column shows what type of HTTP method the API uses (GET, POST, PUT, DELETE, etc). The column also displays the part of the URL you would need to append to the base URL.
3. Look in the C# code. Under the controllers folder, you can find controllers that handle some of the API functionality. If an API deals with a specific model, it will typically be held under that respective controller (if it deals with a user for example, it will be under the User Controller). The folder location is shown in the below image.



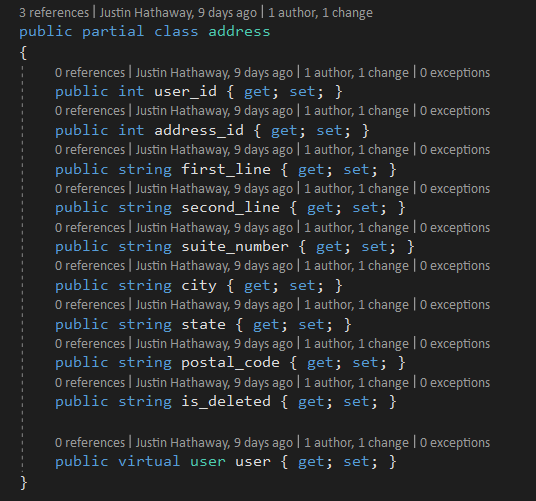
Opening a specific folder, we can see our different web APIs. Below is one API method that returns a specific user based on the user id that is passed in. ‘[ResponseType(typeof(user))]’ states that this method returns a User in JSON form. The user type is based on the Entity model found under (Models->MoolahModel.edmx->user.cs). ‘[Route(“api/Users/{id}”] specifies that the following API can be called with the url, http://moolah-financial-api.azurewebsites.net/api/users/{id} (with {id} being replaced by an integer).



Picture showing the location of the Entity Models that the Entity Controller is using:

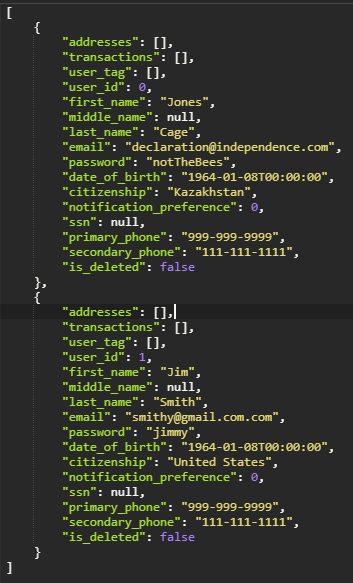


Here is an example Entity model:



**APIs that currently exist:**

* Under the User Controller
  + *GetAllUsers*
    - **Description**: Retrieves an array of all the users that are currently stored in the database.
    - **Purpose**: We probably won’t need to use this in our application for displaying any data. It is useful however for seeing which users exist without having to open SQL Server to interact directly with our database.
    - **Route**: http://moolah-financial-api.azurewebsites.net/api/Users
    - **JSON Body in Request**: None
    - **Example Response**:



* + *GetUserById*
    - **Description:** Retrieves a single user based on the passed in user\_id (integer).
    - **Purpose:** The application needs to know which user is currently signed in to the application.
    - **Route:** http://moolah-financial-api.azurewebsites.net/api/Users/{id}
    - **JSON Body in Request**: None
    - **Example Response:** (if user\_id is 1 and the user exists)