Christopher Evans (5829035234)

User guide

Contents

[Introduction 1](#_Toc83982513)

[Setup 1](#_Toc83982514)

[Default Super User 2](#_Toc83982515)

[Building with Visual Studio 2019 2](#_Toc83982516)

[Authentication 3](#_Toc83982517)

[Registering 4](#_Toc83982518)

[Registering an Admin 4](#_Toc83982519)

[Logging out 4](#_Toc83982520)

[Endpoints 4](#_Toc83982521)

[JSON Objects 4](#_Toc83982522)

[User Credentials 4](#_Toc83982523)

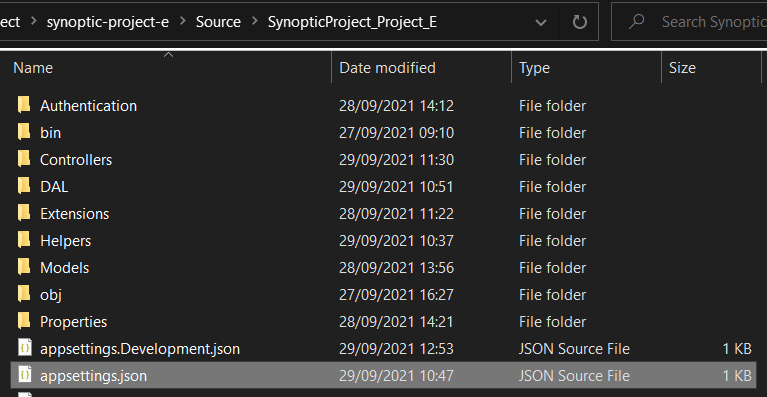
[User Register 5](#_Toc83982524)

# Introduction

The purpose of this document is to detail how to set up the project, how to build, unit test and run the project and the usage guide for the RESTful API web service.

# Setup

Before running for the first time, take due note of the appsettings.json file, located in the SynopticProject\_Project\_E folder.



Inside it is a field named “AppSettings”. It is an object containing additional fields which can be configured to suit the environment the application will be deployed to. The fields are defined as follows:

|  |  |  |
| --- | --- | --- |
| Field Name | Field Type | Description |
| ConnectionString | String | A MongoDB connection string to connect to. Default value is localhost. |
| DatabaseName | String | The name of the database to store collections in. |
| DefaultSuperUser | Object | An object detailing the default Super User. On app start up, if no administrator account exists, one will be created with these details. |

## Default Super User

The default super user object is defined as follows (All fields are required):

|  |  |  |
| --- | --- | --- |
| Field Name | Field Type | Description |
| CardId | String (16 characters) | The card ID (alphanumeric, 16 characters) |
| PIN | String (4 characters) | 4 digit PIN |
| EmployeeId | 16-bit Integer | Unique employee ID |
| FirstName | String (50 characters) | User’s first name |
| Surname | String (50 characters) | User’s surname |
| EmailAddress | String(50 characters) | User’s email address |
| MobileNumber | String (20 characters) | User’s mobile number |

## Building with Visual Studio 2019

Once you’ve configured the appsettings.json file, open the solution file in Visual Studio 2019. The project is a standard ASP.NET Core 3.1 Web API written in C#.

Shape

Description automatically generated with medium confidence

Before running, in the solution explorer, right click the solution and select “Restore NuGet Packages” to install the project dependencies.

A screenshot of a computer

Description automatically generated with medium confidence

Optionally, you can run the Unit Tests by opening the Test Explorer and clicking “Run All Tests In View”.

Graphical user interface, application

Description automatically generated

Then, to run, click the “Play” button at the top of the program.

Graphical user interface, application

Description automatically generated

# Authentication

Authentication to the API uses Basic Authorization, thus a Basic Authorization Header is required to make requests. The Authentication string is UTF-8 Base64 encoded, with a colon separating the Card ID and the PIN. For example, a card ID of “0000111122223333” with PIN “2468” would be “0000111122223333:2468” encoded to UTF-8 Base64.

Example Header:

*Authorization: Basic MDAwMDExMTEyMjIyMzMzMzoyNDY4*

Next, a login request should be sent to /*Authenticate/Login/* to perform a handshake with the API. This will start the user’s session. If a user doesn’t send any requests within 5 minutes of the last, their session will end and they will need to log in again.

## Registering

If you cannot login, the server will likely respond saying the credentials are invalid or the user needs to register. You can submit a request to the /*Authenticate/Register/* endpoint to register. You will need to submit a [User Register JSON Object](#_User_Register_JSON) to register the details – note that this is the only method that doesn’t require a Basic Authorization header. Once completed, you will be able to log in.

## Registering an Admin

To create an admin, you will need to login as an admin user as only admins can create other admins. To create an admin, submit a [User Register JSON Object](#_User_Register_JSON) to */User/CreateAdmin/*. If an admin doesn’t exist, you can use the default admin user.

## Logging out

To log out, submit an empty POST request to */Authenticate/Logout/* and the server will respond with “Goodbye”.

# Endpoints

|  |  |  |  |
| --- | --- | --- | --- |
| Endpoint | HTTP Verb | Body | Description |
| /Authenticate/Login/ | POST | [User Credentials](#_User_Credentials) | Used to login to the API. This endpoint must be called to allow other actions. |
| /Authenticate/Register/ | POST | [User Register](#_User_Register_JSON) | Used to register users. Does not require Basic Authorization header. |
| /Authenticate/Logout/ | POST | None | Used to logout a user. |
| /User/?cardId=[CARD\_ID] | GET | None | Get a user by card ID. cardId is a query parameter. Admin users can retrieve the information of any user, however a standard user can only retrieve their own. |
| /User/CreateAdmin/ | POST | [User Register](#_User_Register_JSON) | Create an admin user. This action can only be performed by other admin users. |

# JSON Objects

## User Credentials

|  |  |  |
| --- | --- | --- |
| Field Name | Field Type | Description |
| CardId | String (16 characters) | The card ID (alphanumeric, 16 characters) |
| PIN | String (4 characters) | 4 digit PIN |

## 

## User Register

Used for registering users and creating admin users.

|  |  |  |
| --- | --- | --- |
| Field Name | Field Type | Description |
| CardId | String (16 characters) | The card ID (alphanumeric, 16 characters) |
| PIN | String (4 characters) | 4 digit PIN |
| EmployeeId | 16-bit Integer | Unique employee ID |
| FirstName | String (50 characters) | User’s first name |
| Surname | String (50 characters) | User’s surname |
| EmailAddress | String(50 characters) | User’s email address |
| MobileNumber | String (20 characters) | User’s mobile number |