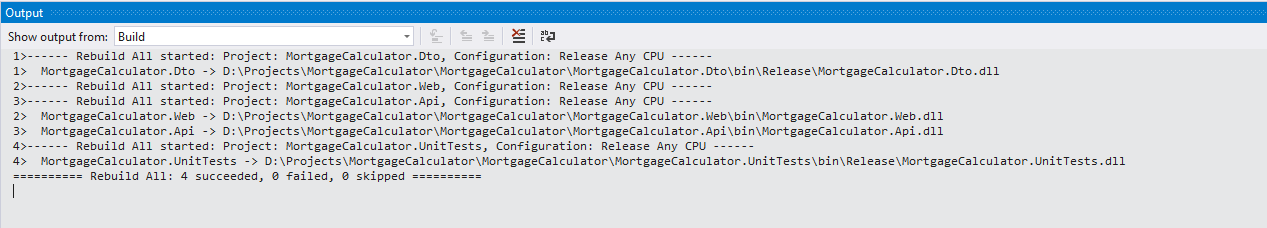
Source code

1. The source code has been compiled in Release Mode and uploaded to Git Repository below,
2. Ensure to clone the repository locally and launch the solution your Visual Studio IDE.

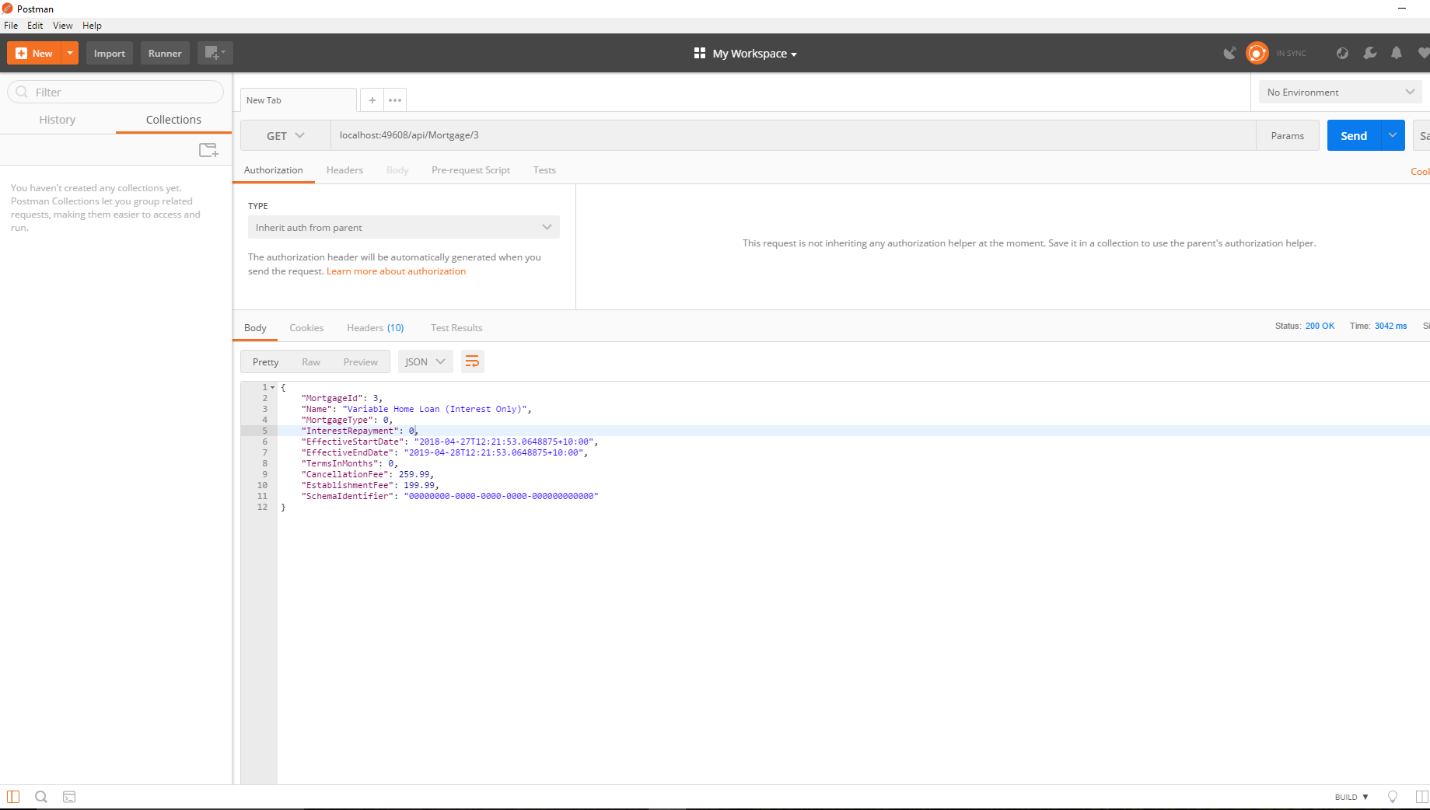
Note: Visual Studio 2017 was used to complete the solution

1. Ensure the compile is successful.



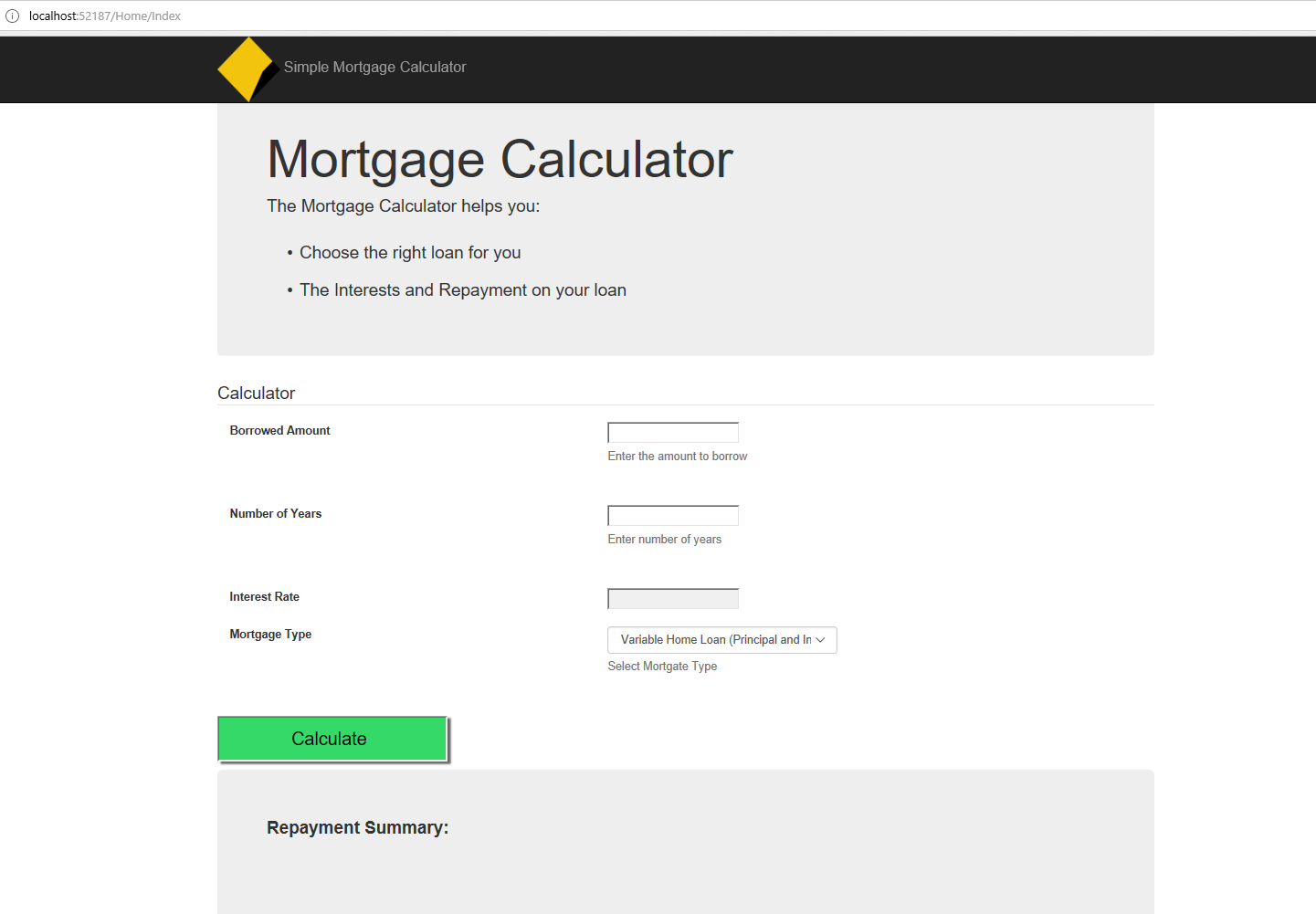
Hosting the Web API:

1. Ensure to host the Web API in IIS or launching the MortgageCalculator.Api project locally.
2. Upon hosting verify if the below URLs return response,
   1. http://localhost:49608/api/Mortgage/
   2. <http://localhost:49608/api/Mortgage/3>
3. The output of 2.b. would look like the below; it is captured using the Postman tool.

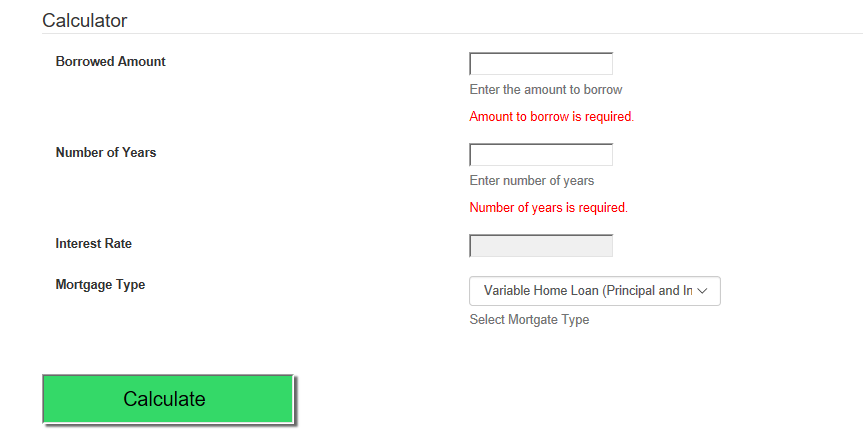


Front-end Application

1. Launch the MortgageCalculator.Web application
2. Home/Index page looks like the below,

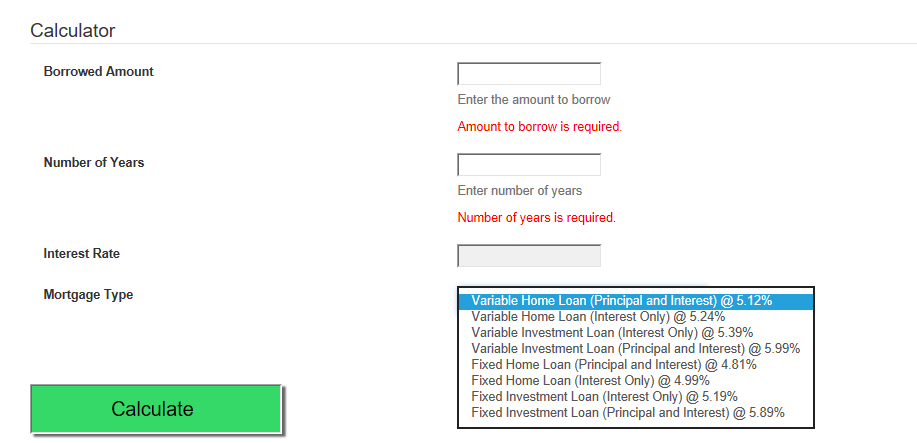


1. Ensure to key-in all the required fields such as Borrowed Amount and Number of Years, otherwise the mandatory field validation will fire and show the below validation messages,

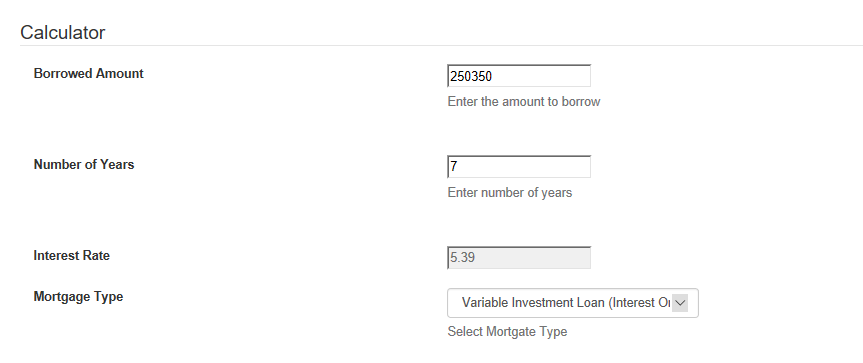


1. The available Mortgage types are loaded into the drop-down list through a REST call from jQuery ($.ajax and GET method) to the Web API controller MortgageController.Get()

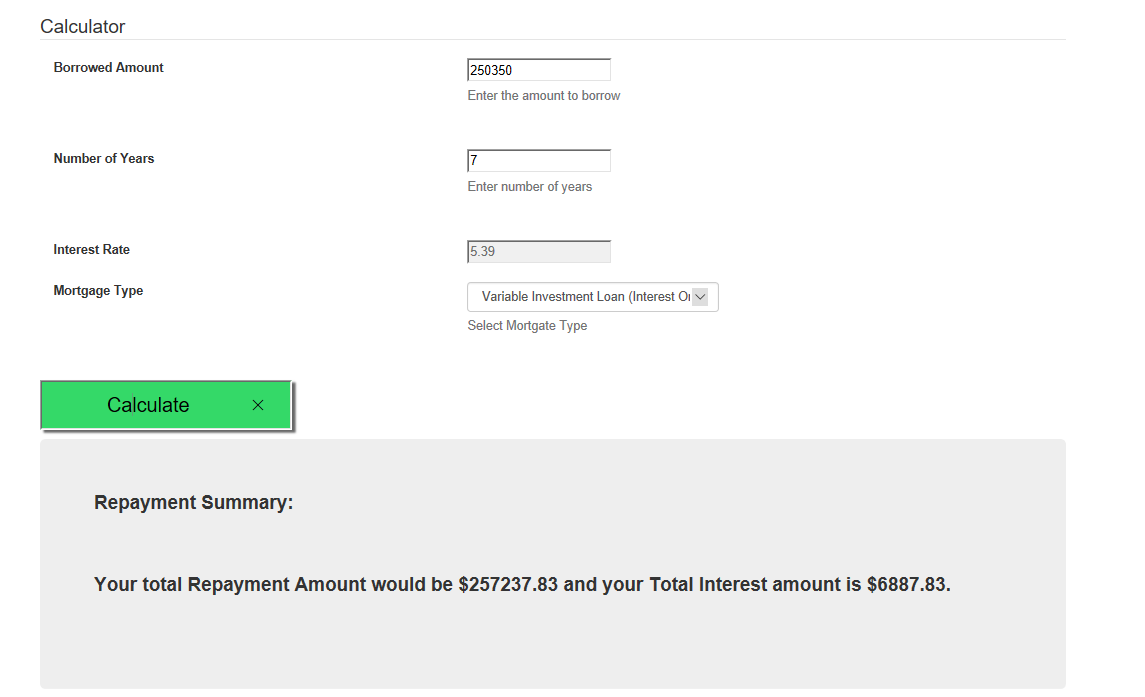
Web -> API -> MortgageData.dll



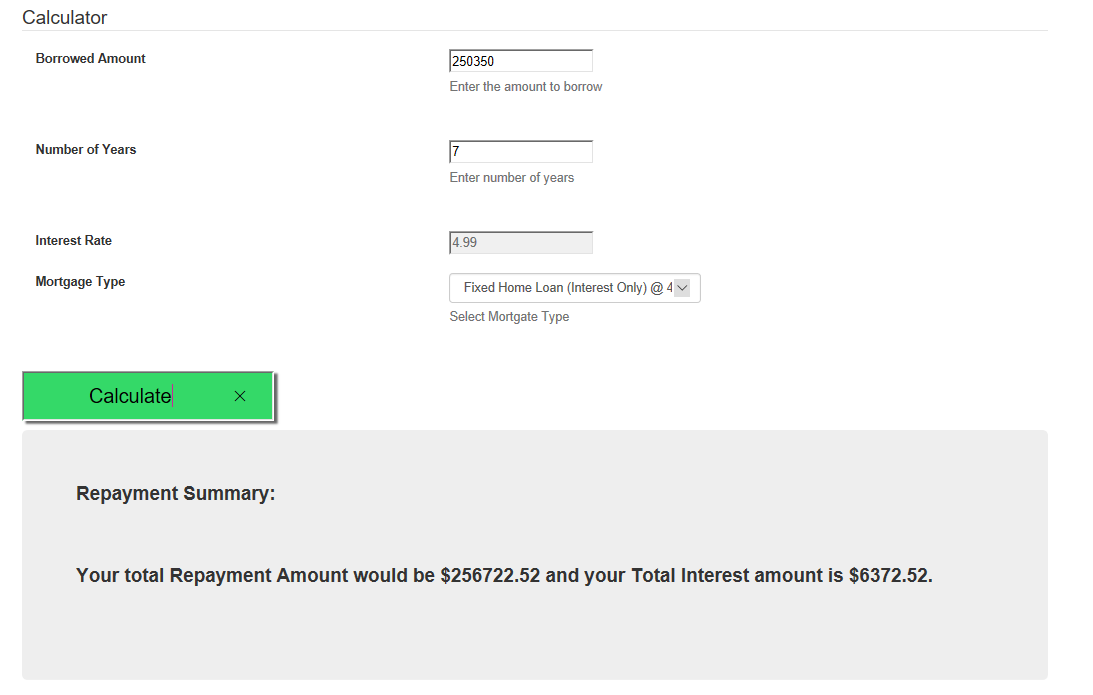
1. The Interest Rate gets auto-populated once Mortgage type is selected. The Interest rate cannot be modified by User as the rate is decided by the Mortgage type selected.



1. Complete the Borrowed Amount, Number of Years after step 5 and click on calculate,



1. Try to change the Mortgage type to Fixed (4.99%), click on Calculate. The Repayment amount and Interest are calculated again based on the new values supplied.



1. The Business Logic for the Total Repayment and Total Interest calculation is self-assumed. Used the classical Loan Amortization Formula. Refer GetRepaymentSummary() in the HomeController.

Unit Testing

1. There are 7 unit tests included in the Unit test project that covers the Web Controllers, Web API Controllers, DTOs, Route Configurations, Repositories.
2. The Unit test coverage is captured using the dotCover tool
3. The test coverage report is included as a HTML report in the project folder,

\\MortgageCalculator\TestCoverageResults\_dotCover\MortgageCalculator.UnitTests.CodeCoverage\_20180430.html