**Project Title**

Mortgage Calculator

**Getting Started**

**Prerequisites**

Visual Studio 2017

Nunit (Nuget Package)

AxoCover(for code coverage) – integrated with Visual studio

jQuery

Moq

Resharper (optional)

**Separation of layers**

#### MortgageCalculator.api

Controller – MortgageController exposes Mortgage data to the client

Repo – Data Repository, fetches data from MortgageDataContext and returns Mortgage List

Service – Acts as an interface between Controller and Repo

Helpers – WebApiOutcache to cache data for 24hrs

**Code changes:**

1. 24hrs duration custom Caching implemented.(As stated Calling the third party data source is costly and data only gets updated every 24 hours)
2. CORS are enabled by adding custom header Access-Control-Allow-Origin in web.config.
3. Removed unused namespaces in the entire project using Resharper

#### MortgageCalculator.Dto

This is a data layer with data transfer objects. This project contains a Mortgage class with properties and fields which defines the Mortgage Type members

**Code changes:**

1. Added property field InterestRate to the class Mortgage
2. EstablishmentFee = mortgage.EstablishmentFee ( code correction)
3. Added new property field public string MortgageTypeValue { get; set; } (To display in the Motgages combo as text)
4. TermsInMonths = (mortgage.EffectiveEndDate.Month - mortgage.EffectiveStartDate.Month) + 12 \* (mortgage.EffectiveEndDate.Year - mortgage.EffectiveStartDate.Year)

*MortgageCalculator.Repo*

**Code changes:**

1. Filtered only the active mortgages where startdate is less than today and enddate is greater than today.
2. Mortgage data is sorted by MortgageTypeText and then by InterestRate.

#### MortgageCalculator.UnitTests

API – Test classes to test the API

HttpClientTests- Test the http responses

#### MortgageCalculator.Web

**Code changes:**

1. Added the jQuery ajax calls and calculations in jQueryHttp.js file and placed in Scripts/Custom folder and this script bundle is added to bundles and registered in BundleConfig.cs
2. Front end form designed using Html5 and text box number validations are handled by Html5 textbox type control.

**Assumptions made:**

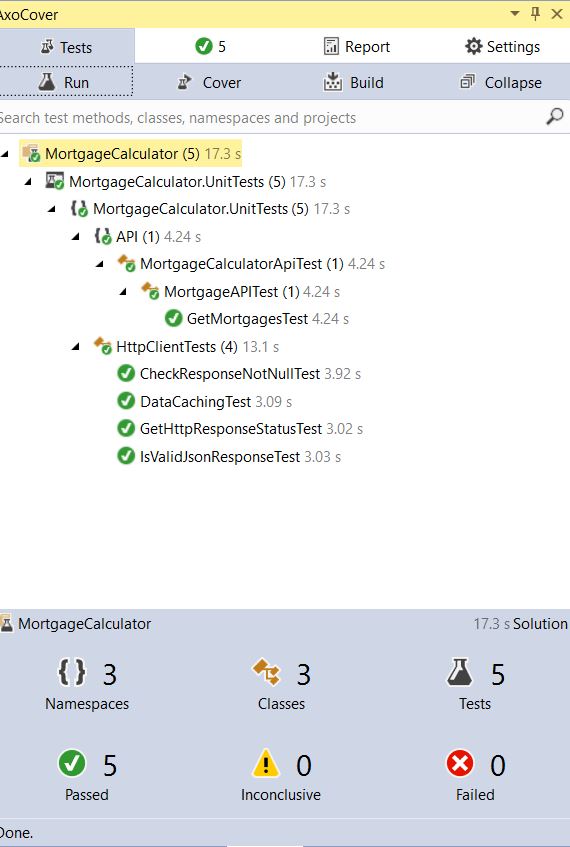
1. Load duration(In years) and Loan Amount fields only accepts numbers greater than 0 and does not accept decimals.
2. Assuming the formula for calculating the repayments is the same for all mortgage types.

If the formula varies depending on mortgage types , banks and country, then the code has to be refactored to make it loosely coupled.

1. Loan duration is not limited to max number of years.

**Running the tests**

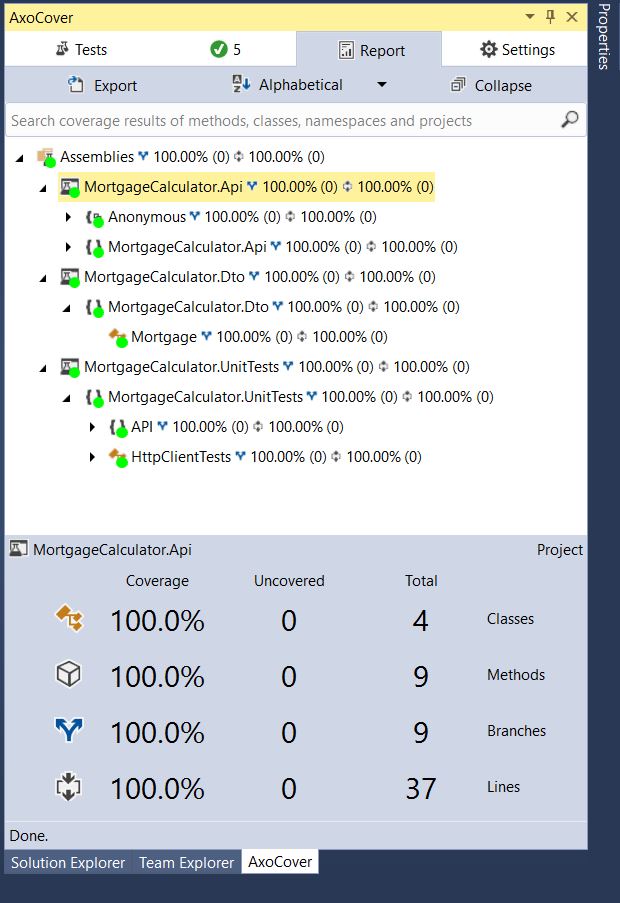
Test Report



**Code Metrics**

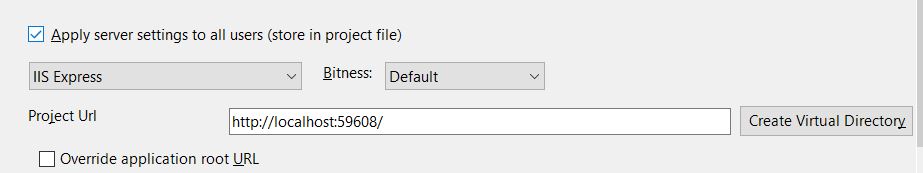
Used AxoCover for code coverage. Below is the codecoverage report

Attained 100% code coverage for all the 3 projects in the solution.

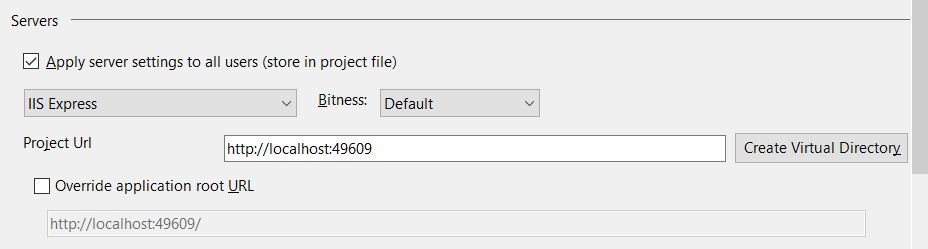


**Deployment**

API is configured to run on the port 59608



Web application is configured to run on the port 49609



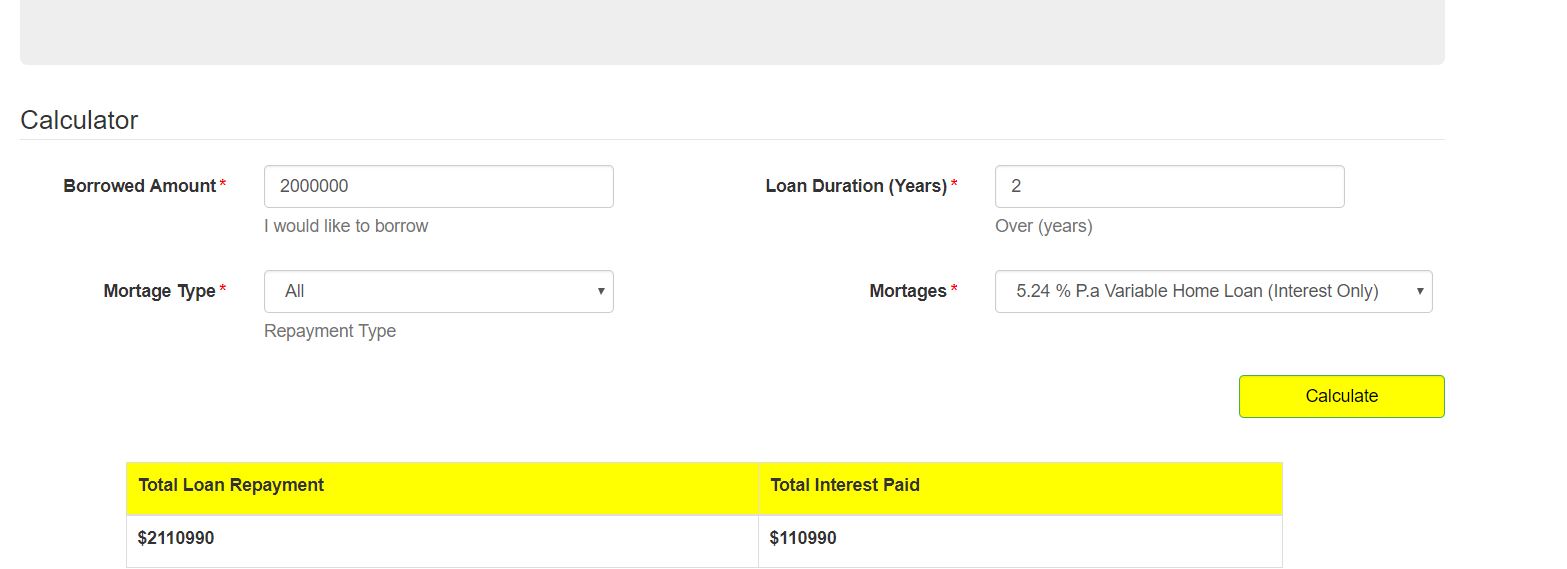
**Versioning**

.Net framework – 4.5.2

Moq – v4.0 (Runtime version)

NUnit – V4.0 (Runtime version)

**Screen UI**



Assembly Meta data information gathered for the MortgageData.dll using ildasm.

