# How to Run

For debugging purposes application could be run in Visual Studio 2017. (LocalDb) is used for data storage.

For running application on server it could be imported from publish package into IIS (Deploy package is in {ProjectPath}\WebDeploy\InteresCalculation.zip.

On application import, database connection strings should be specified. Before importing application attaching project databases on sql server is needed.

<https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-attach-db-transact-sql>

Databases are located in the {ProjectPath}\InterestRateCalculation\App\_Data. These 2 databases should be specified in web config.

There is possibility to use migration packages from solution. In Visual Studio Package Manager Console command Add-Migration should be called. If there is no migration created in Migrations folder. After this Update-Database command should be executed. Migration packages should be created for both databases.

# Web Application

Web application developed using asp.net mvc. In Application home page should be implemented.

On the top there are 2 menu items, which were developed for this task – Customers, Agreements.

In the Customers menu item there are creating, updating, deleting, viewing actions.

In Agreements menu item the same actions are developed.

Agreements/Details – Full information about customer and agreement is shown. On the page there is New Base Rate input, where base rate can be selected. After selection and pressing Compare, new fields appears which show new interest rate and difference between current and new interest rates. Button compare posts data to the application and reloads the page. Interest rates values are calculated on the form reload.

Agreements/Edit – There is possibility to edit almost all data except Margin. When new Base Rate is selected AJAX POST command is generated to receive new interest rates. When data is received values are places into relevant fields.

# Web API (REST)

Web service is integrated into the same solution.

Api receives queries through the address: [domain]/api.

Data to the web service method should be passed through GET method.

Api receives Agreement details, customer for whom the agreement is assigned and new base rate code. As a response data about agreement and comparison of current and given interest is given.

List of method parameters:

* CustomerPersonalId
* BaseRateCode – Current base rate code
* NewBaseRateCode – Base rate code to compare with
* Amount
* Margin
* Duration

Request example: localhost/api?CustomerPersonalId=67812203006&BaseRateCode=VILIBOR1m&NewBaseRateCode=VILIBOR3m&Amount=100&Margin=2.2&Duration=20

# Recommendations / Improvements

* Implement better error handling. There’s no decent view’s for error handling and informing user about what happened.
* Refactoring needed for removing duplicate code, there’s some in AgreementsController.
* Implement unit tests.
* Dropdown fields should be changed to autocomplete where data to select from could grow.
* In the Agreement details form, ajax could be used too.