# TAL Premium Calculator

**Project Repo:**

<https://github.com/vondool/TALPremiumCalculator>

|  |  |
| --- | --- |
| **Date Issued** | 14 Nov 2021 |
| **Updated By** | Mehedi Shams Rony |
| **Reviewed By** |  |
| **Security Classification** | Confidential |
| **Version** | 1.0 |

# Table of Contents

[TAL Premium Calculator 1](#_Toc87825597)

[Table of Contents 2](#_Toc87825598)

[Introduction 3](#_Toc87825599)

[Purpose 3](#_Toc87825600)

[Assumptions 3](#_Toc87825601)

[Solution Architecture 3](#_Toc87825602)

[Coding Framework 3](#_Toc87825603)

[Layers 3](#_Toc87825604)

[Operational Flow 4](#_Toc87825605)

[Data Model 4](#_Toc87825606)

[ERD 4](#_Toc87825607)

[Get Occupations 5](#_Toc87825608)

[Requirements 5](#_Toc87825609)

[Sequence flow 5](#_Toc87825610)

[Request JSON Example 5](#_Toc87825611)

[Response JSON Example 5](#_Toc87825612)

[Data Elements 6](#_Toc87825613)

[Get Rating Factor 6](#_Toc87825614)

[Requirements 6](#_Toc87825615)

[Sequence flow 6](#_Toc87825616)

[Request JSON Example 6](#_Toc87825617)

[Response JSON Example 7](#_Toc87825618)

[Data Elements 7](#_Toc87825619)

[Tests 7](#_Toc87825620)

[Repo Test 7](#_Toc87825621)

[Validation Test 7](#_Toc87825622)

[Postman test 8](#_Toc87825623)

[Rating Test 8](#_Toc87825624)

[Occupations Test 9](#_Toc87825625)

[Front-End 10](#_Toc87825626)

[UI 10](#_Toc87825627)

[Conclusion 10](#_Toc87825628)

[Improvements 10](#_Toc87825629)

# Introduction

## Purpose

This document details the process to obtain the monthly premium according to the given age, SI and occupational rating factor.

For any given individual the monthly premium is calculated using the below formula:

Death Premium = (Death Cover amount \* Occupation Rating Factor \* Age) /1000 \* 12

## Assumptions

* Client name can be a maximum of 50 characters long.
* For testing, the database needs to be attached to SQL Server and the connection string needs to be modified at

appsettings.Development.json

appsettings.json

The current connection is:

"TALDB": "Server=(localdb)\\MSSQLLocalDB;initial catalog=TAL;integrated security=sspi"

# Solution Architecture

## Coding Framework

Following frameworks, patterns and practices are used in this solution.

* .NET Core 5.0
* .NET Framework 4.7.2
* Mediator pattern
* Repository pattern
* Unit of work pattern
* CQRS pattern
* Fluent validation
* Moq

## Layers

Three layers are used for interactivity whereas one extra test project is added also.

* TALDataModel
  + Framework .NET Core 5.0
* TALWebAPI
  + Framework .NET Core 5.0
* TALWebSiteDotNet
  + Framework .NET Framework 4.7.2
* TALAPITest
  + Framework .NET Core 5.0

## Operational Flow

TALWebSiteDotNet

TALWebAPI

DataModel

**STPService**

Request

Response

Request

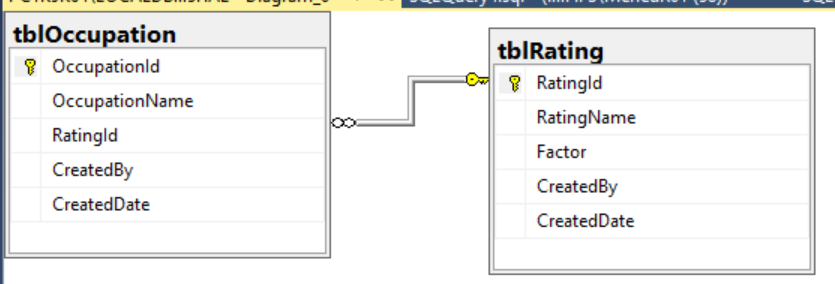
Response

Request

Response

# Data Model

## ERD



# Get Occupations

## Requirements

Get all the available listed occupations.

## Sequence flow

TALWebSiteDotNet

TALWebAPI

DataModel

**STPService**

Occupation Query

Occupation list

Occupation Query

Occupation list

Occupation Query

Occupation list

## Request JSON Example

This API doesn’t need a request JSON.

## Response JSON Example

{

OccupationId = 1,

OccupationName = "Cleaner",

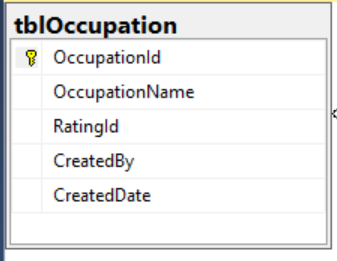
RatingId = 3

},

|  |  |  |
| --- | --- | --- |
| Field name | Description | Database field |
| OccupationId | Unique id of the occupation | TAL->tblOccupation->OccupationId |
| OccupationName | Occupation title | TAL->tblOccupation->OccupationName |
| RatingId | Corresponding rating in the rating table. | TAL->tblOccupation->RatingId |

Below response will be sent to the consumer on success:

## Data Elements



# Get Rating Factor

## Requirements

Get all the rating factor associated with a given occupation.

## Sequence flow

TALWebSiteDotNet

TALWebAPI

DataModel

**STPService**

Rating Query

Occupation id and rating id

Rating Query

Join tables and get the factor

Rating Query

Factor

## Request JSON Example

{

OccupationId = 1

}

|  |  |  |
| --- | --- | --- |
| Field name | Description | Database field |
| OccupationId | Unique id of the occupation | TAL->tblOccupation->OccupationId |

## Response JSON Example

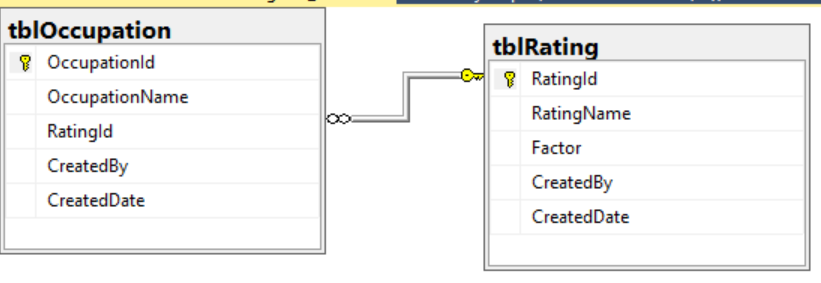
{

Factor = 1.75

}

## Data Elements

Inner join the two tables to obtain the corresponding factor for a given occupation.



# Tests

## Repo Test

The solution provides repo test. This test ensures the following.

* All the available occupations are returned on the GetOccupations() call.
* The corresponding factor for a given occupation is correctly returned.

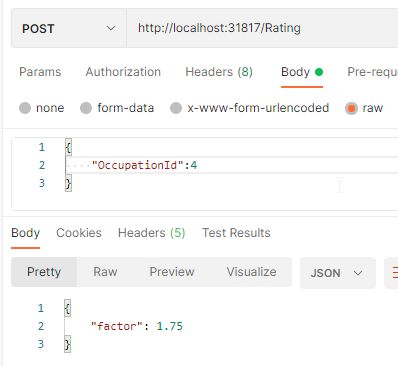
## Validation Test

The solution provides validation test for the Rating query. This test ensures the following.

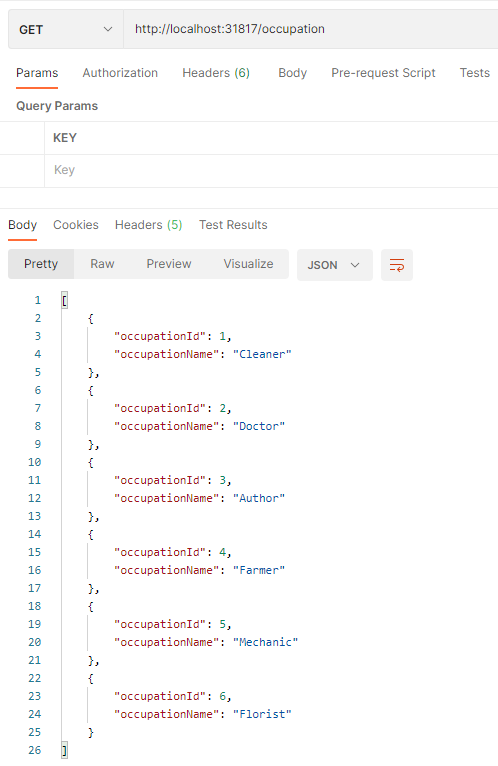
* The occupation id is an integer in a specified range.

## Postman test

### Rating Test

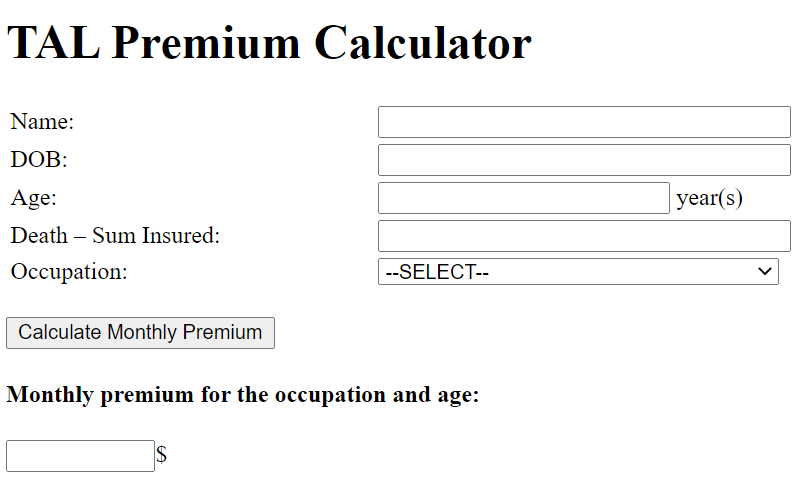


### Occupations Test



# Front-End

## UI



# Conclusion

## Improvements

* Code commenting is not done yet due to time constraint
* Swagger is not added yet; that might be added