# .Net Interview Project v2

Insurance Calculation

Version 1.0

Ömer Karamanlı

08/11/2022

|  |  |
| --- | --- |
| Document Versioning | 1 |
| 1. Introduction | 2 |
| 1.1. Purpose of the Project | 2 |
| * 1. Technologies | 2 |
| 2. How Does It Work | 2 |
| 2.1. What to Expect as a Response | 2 |
| 2.2. Methods | 3 |
| 2.2.1 Product | 3 |
| 2.2.2 Product Order | 4 |
| 2.2.3 Surcharge | 5 |

## Document Versioning

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Remarks |
| 04/11/2022 | 1.0 | Ömer Karamanlı | First version documentation |

## Introduction

### Purpose of the Project

The purpose of the project is to develop an API to calculate the insurance value of the products and insure the products with the calculated value so that CoolBlue gets money back in case the product gets lost or damaged before reaching the customers.

### Technologies

The technologies used in this project are as follows:

.Net 6

Packages that were used

* Serilog
* CsvHelper
* NewtonSoft

## How Does It Work?

### What to Expect as a Response

The response from the request will be in JSON format. An example of response model is as follows:

|  |
| --- |
| {      "stateCode": 10,  // 10 Success, 20 Error      "errorMessage": **null**,  // When state code is 20, the explanatory error message will be filled and printed here      "errorCode": 0,  // When state code is 20, this area will be bigger than 0.  // Error codes: 2000 Has Exception, 2001 Model is not Valid, 2003 Product not found      "result": 1650  //This area is the result, final insurance value. When there is an error, this area will be 0  } |

### Methods

|  |  |  |
| --- | --- | --- |
| Address | Method | Explanation |
| {base\_url}/api/insurance/product | POST | Returns the insurance value of a product |
| {base\_url}/api/insurance/productOrder | POST | Returns the insurance value of an order |
| {base\_url}/api/insurance/surcharge | POST | Add surcharge rates to the insurance values |

#### Product

|  |  |
| --- | --- |
| HTTP-METHOD | POST |
| REST-URI | {base\_url}/api/insurance/product |
| BODY | JSON type |

This method is used to calculate insurance value of a single product.

Example – Request Body

|  |
| --- |
| {"ProductId" : 861866} |

Example – Response Body

|  |
| --- |
| {      "stateCode": 10,      "errorMessage": **null**,      "errorCode": 0,      "result": 1650  } |

Assumptions:

I assumed that only product id will be given but prepared for other InsuranceDto properties to be given as a parameter too

I assumed that whether a digital camera can be insured or not, the insurance value will be increased by 500 euros because of feature 2

#### Product Order

|  |  |
| --- | --- |
| HTTP-METHOD | POST |
| REST-URI | {base\_url}/api/insurance/productOrder |
| BODY | JSON type |

This method is used to calculate insurance value of an order.

Example – Request Body

|  |
| --- |
| [{"ProductId" : 735246},  {"ProductId" : 735246}] |

Example – Response Body

|  |
| --- |
| {      "stateCode": 10,      "errorMessage": **null**,      "errorCode": 0,      "result": 2000  } |

Assumptions:

I assumed that only a list of product ids will be given but prepared for other InsuranceDto properties to be given as a parameter too

I assumed that whether a digital camera can be insured or not, the insurance value will be increased by 500 euros because of feature 2

#### Surcharge

|  |  |
| --- | --- |
| HTTP-METHOD | POST |
| REST-URI | {base\_url}/api/insurance/surcharge |
| BODY | JSON type |

This method is used to add surcharge rates to the insurance values of a single product.

Example – Request Body

|  |
| --- |
| {"ProductId": 8618665,  "SurchargeRate":10} |

Example – Response Body

|  |
| --- |
| {      "stateCode": 10,      "errorMessage": **null**,      "errorCode": 0,      "result": 0  } |

Assumptions:

Because it said surcharge rate, I assumed that it will be a percentage instead of a value.

I assumed that there won’t be a database (even though it would be better) so I’m keeping the surcharge rates for every product in a csv file.

I assumed that there could be negative surcharge rates because there might be a discount.

I assumed that the limit for a surcharge rate is going to be 100%