‘WINGS ON’ USER STORY

**Technical Assessment**

|  |  |
| --- | --- |
|  | |
| Customer Name | Travix |
| Project/Sub-Project | Technical Assessment documentation |
| File Name | Wings Documentation.docx |
| Version | 0.1 |
| Compatibility Version | 1.0 |
| Author | Mohamed Salah |
| Publish Date | 15-JUN-2019 |
| Reviewer |  |
| Review Date |  |

Disclaimer

© 2019 All rights reserved. This document has been created for use by Travix International. These rights are not transferable to any third party without the prior written consent of the author.

Table of contents

[Record of Changes 5](#_Toc11687399)

[Approvals 6](#_Toc11687400)

[Distribution List 6](#_Toc11687401)

[Documents References 7](#_Toc11687402)

[1 Sum of Multiple 8](#_Toc11687403)

[1.1 Problem statement 8](#_Toc11687404)

[1.2 Assumption 8](#_Toc11687405)

[1.3 Solution Approaches 8](#_Toc11687406)

[1.3.1 Approach A 8](#_Toc11687407)

[1.3.2 Approach B 9](#_Toc11687408)

[1.3.3 Approach C 10](#_Toc11687409)

[1.4 Design Pattern 11](#_Toc11687410)

[1.4.1 Template Method Pattern 11](#_Toc11687411)

[1.4.2 Dependency Injection 12](#_Toc11687412)

[1.5 Statistical Running 12](#_Toc11687413)

[2 Sequence Analysis 13](#_Toc11687414)

[2.1 Problem Statement 13](#_Toc11687415)

[2.2 Assumption 13](#_Toc11687416)

[2.3 Solution Approaches 13](#_Toc11687417)

[2.3.1 Approach A 13](#_Toc11687418)

[2.3.1.1 Pseudocode and Complexity 13](#_Toc11687419)

[2.3.2 Approach B 13](#_Toc11687420)

[2.3.2.1 Pseudocode and Complexity 14](#_Toc11687421)

[2.4 Design patterns 15](#_Toc11687422)

[2.4.1 Factory Method Pattern 15](#_Toc11687423)

[2.4.2 Dependency Injection 16](#_Toc11687424)

[2.5 Statistical Running 16](#_Toc11687425)

[3 Testing Solution 18](#_Toc11687426)

# Record of Changes

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Change Reference |
| 0.1 | 05-Jan-2019 | Mohamed Salah | Initial version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Approvals

This document has been approved by the following people

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Function | Date of Approval | Signature |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Distribution List

This document has been distributed to:

|  |  |
| --- | --- |
| Name | Function |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Documents References

|  |  |  |
| --- | --- | --- |
| Document Name | Document Reference | Version |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 

# Purpose

The purpose of this document is presenting the **design** and **development** approaches to achieve the requirement in the technical assignment.

I did my best to cover most common topic in design Restful API.

# Problem statement

As a Front-End Developer, I need a REST-ful Web API for my ticketing website, so that I can access and manage information related to the passengers.

Website need the following:

1. Endpoint that returns a person by Id.

2. Endpoint that returns all passengers on the flight by number for example ‘PZ696’.

3. Endpoint that updates a person’s email address.

4. Endpoint that creates a booking of an existing flight for a new passenger.

5. Endpoint that lists all the male passengers.

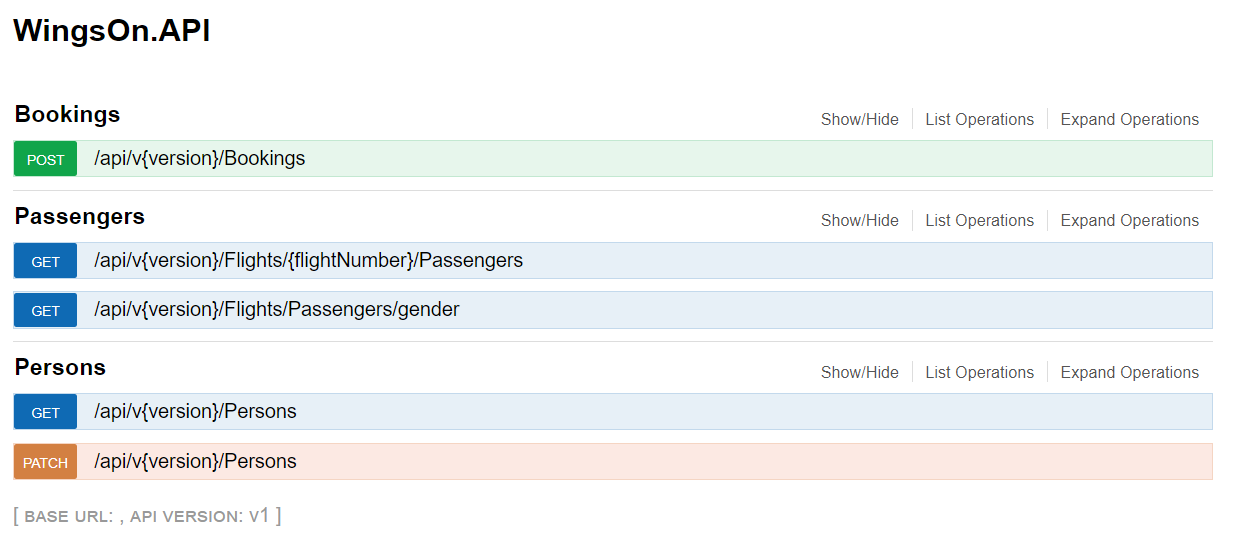
# Endpoint Table

The following is the Endpoint Table

|  |  |  |
| --- | --- | --- |
|  | Endpoint | Usage |
| POST | /api/v1/Bookings |  |
| GET | /api/v1/Flights/{flightNumber}/Passengers |  |
| GET | [/api/v1/Flights/Passengers/gender](http://localhost:63312/swagger/ui/index#!/Passengers/Passengers_Get) |  |
| GET | [/api/v1/Persons](http://localhost:63312/swagger/ui/index#!/Persons/Persons_Get) |  |
| PATCH | [/api/v1/Persons](http://localhost:63312/swagger/ui/index#!/Persons/Persons_Get) |  |

Also, you can find this using Swagger url

http://<hostname>:<port>/swagger/ui/index



# Assumption

The following assumption will apply for the different approaches

* This approach is focus of build restful API but not deeply focus of business validation for example available capacity of the flight and the departure date of the flight
* No authentication required for that test

# Best Practice followed for Building REST API

### Use nouns but no verbs

For an easy understanding use this structure for every resource:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resource | GET read | POST create | PUT update | PATCH  Update Partially |
| /Bookings |  | Create a new Booking |  |  |
| /Flights/{flightNumber}/Passengers | Returns Passengers in specific flight number |  |  |  |
| [/Flights/Passengers/{gender](http://localhost:63312/swagger/ui/index#!/Passengers/Passengers_Get)} | Returns Passengers in specific gender (male or female) |  |  |  |
| [/Persons](http://localhost:63312/swagger/ui/index#!/Persons/Persons_Get)/{Id} | Returns a specific person |  |  |  |
| [/Persons](http://localhost:63312/swagger/ui/index#!/Persons/Persons_Get)/{Id} |  |  |  | Update person email Email |

Package Used

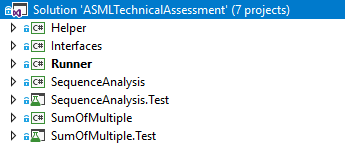
# Testing Solution

To prove the solution for both problem and according to what is minimum required.

New Solution “ASMLTechnicalAssessment” contains the following projects

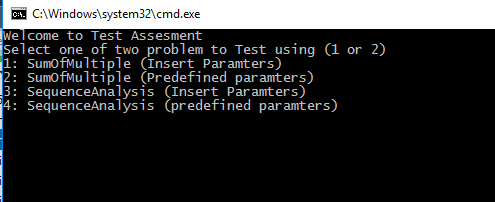
Using Visual studio 17 and .Net Framework 4.7

|  |  |
| --- | --- |
| Project Name | Usage |
| Helper | Contains any common helper methods like  Time tracker , string generator, … |
| Interface | Interfaces for Sequence Analysis & Sum Of Multiple |
| Runner | It will ask the user which of the problems below to solve |
| SequenceAnalysis | Solve the issue of Sequence Analysis with two approaches |
| SequenceAnalysis.Test | Test Solution of the issue of Sequence Analysis with two approaches |
| SumOfMultiple | Solve the issue of Sum Of Multiple with two approaches |
| SumOfMultiple.Test | Test Solution of the issue of Sum Of Multiple with two approaches |



For the Runner project

When user run it, the next screen will appear



|  |  |
| --- | --- |
| Selection Number | Description |
| 1 | this method is used to test Sum Of Multiple  it allow user to test with inserting limit parameters  and also defined the divisors |
| 2 | this method is used to test Sum Of Multiple  it uses predefined limit parameters start from 1000 ,10000,...., 100000000  Also predefined the divisors [2, 3, 5, 7]. |
| 3 | this method is used to test Sequence Analysis  it allow user to test with inserting string parameter |
| 4 | this method is used to test Sequence Analysis it uses  a predefined string parameter that is randomly generated  by method Random String that takes a length of string  In addition, returns a random string. |
| Esc | To exist |