Table of Contents

[Installation guide 3](#_Toc534104969)

[Build the back-end solution 3](#_Toc534104970)

[Built the front-end solution 4](#_Toc534104971)

[Front-end for Task 1 – Hotel establishments 5](#_Toc534104972)

[Home Page 5](#_Toc534104973)

[Search Filter Page 6](#_Toc534104974)

[Mobile View 7](#_Toc534104975)

[Sample for iPhone6 7](#_Toc534104976)

[Angular Artifacts 8](#_Toc534104977)

[UI for Task 2 – Flight segments 9](#_Toc534104978)

[Backend Host 10](#_Toc534104979)

[Business 10](#_Toc534104980)

[Request Engines 10](#_Toc534104981)

[Data - EF Core 11](#_Toc534104982)

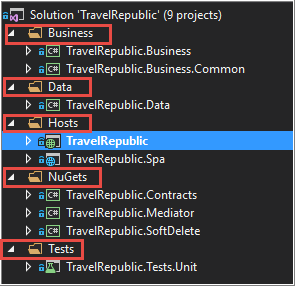
[Unit Tests 11](#_Toc534104983)

[Integration Tests 11](#_Toc534104984)

[Code repository 11](#_Toc534104985)

Solution Architecture Overview

In bird’s eye view, the solution is separated into 5 different areas namely: **Hosts**, **Business**, **Data**, **Tests**



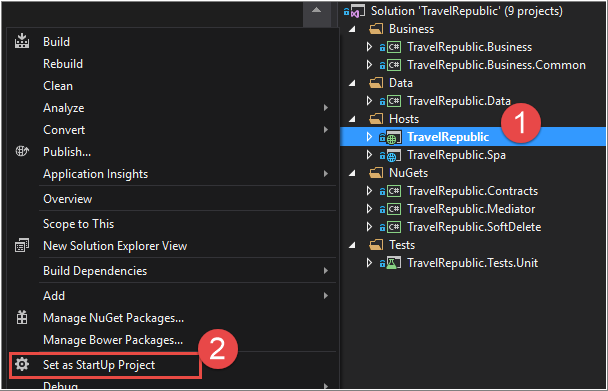
The solution is built using Visual Studio 2017

– with .NetCore 2.2

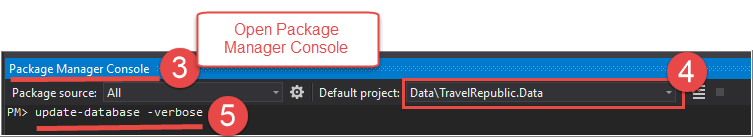
# Installation guide

## Build the back-end solution

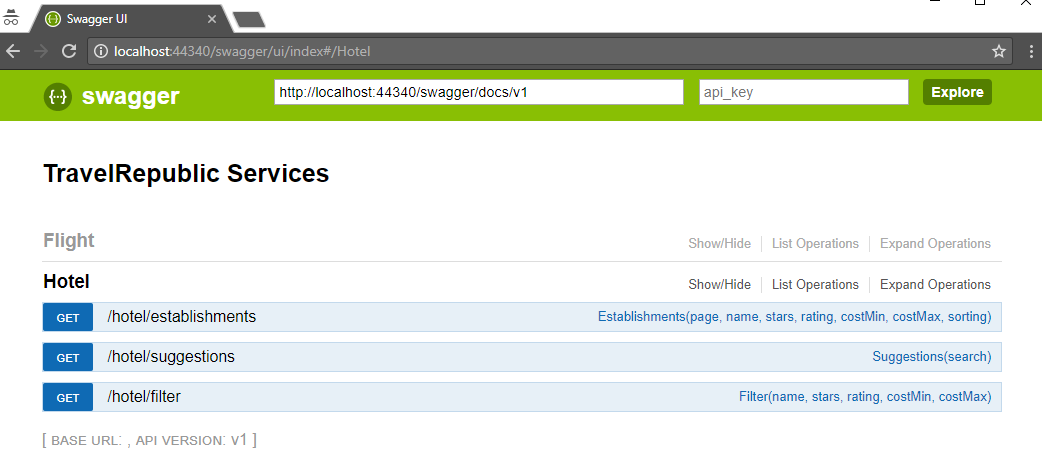
Set start-up Project



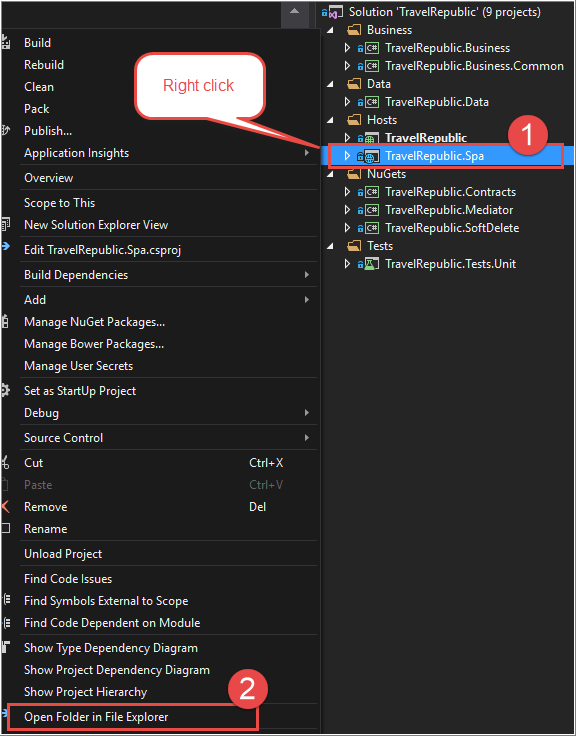
Seed the database via Entity Framework.



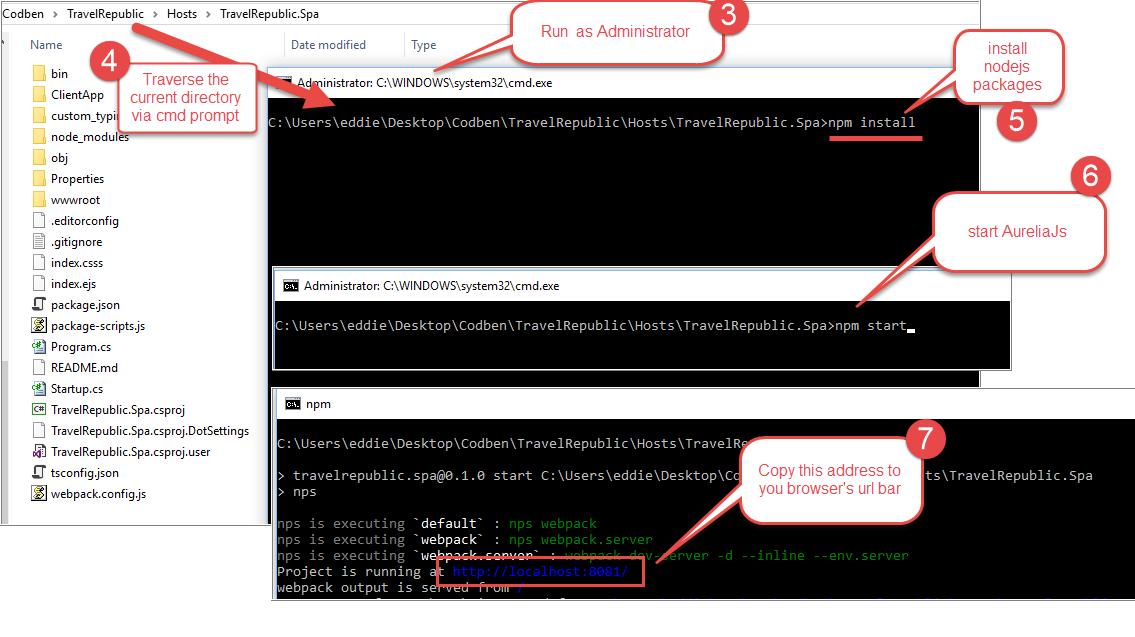
Press F5 to view the WebApi documentation via swagger



## Built the front-end solution



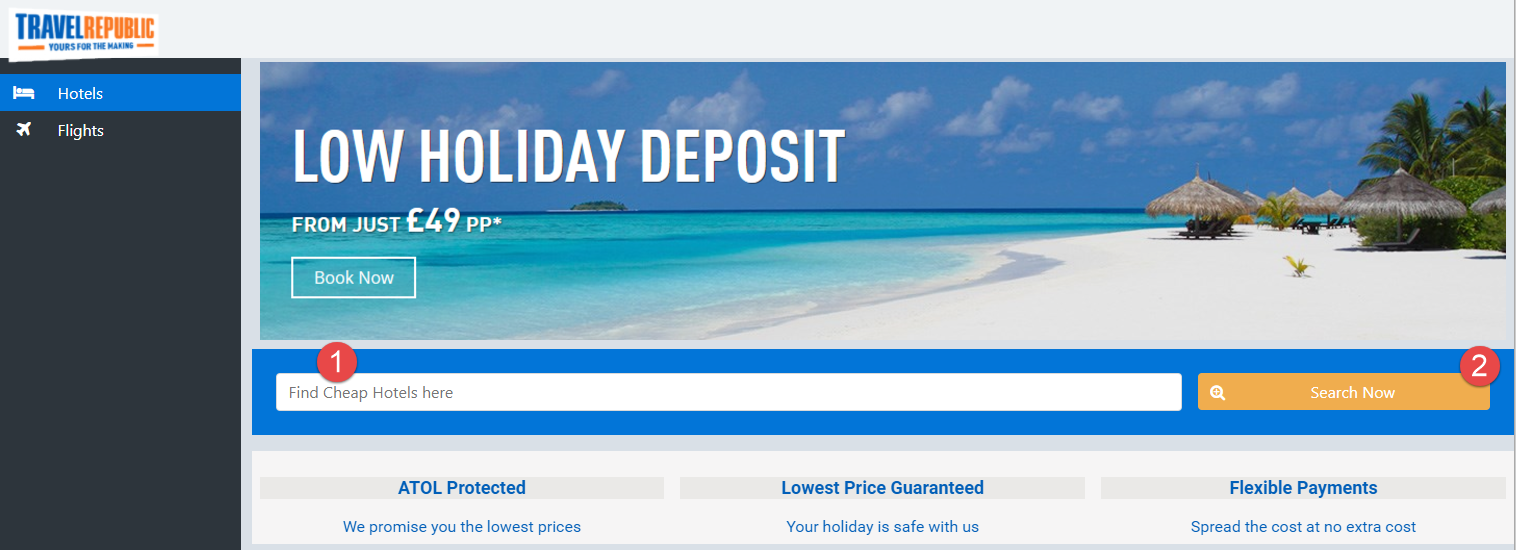
Note: Keep the WebApi solution running while browsing the front-end



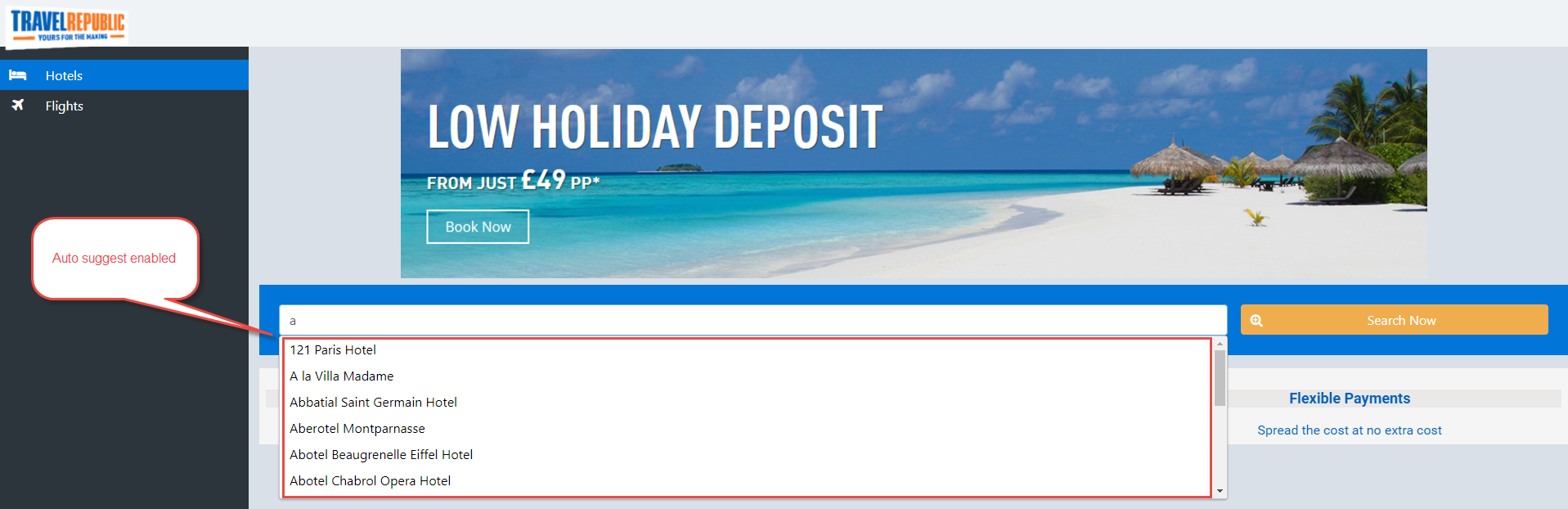
# Front-end for Task 1 – Hotel establishments

## Home Page

Enter search keywords then hit search.

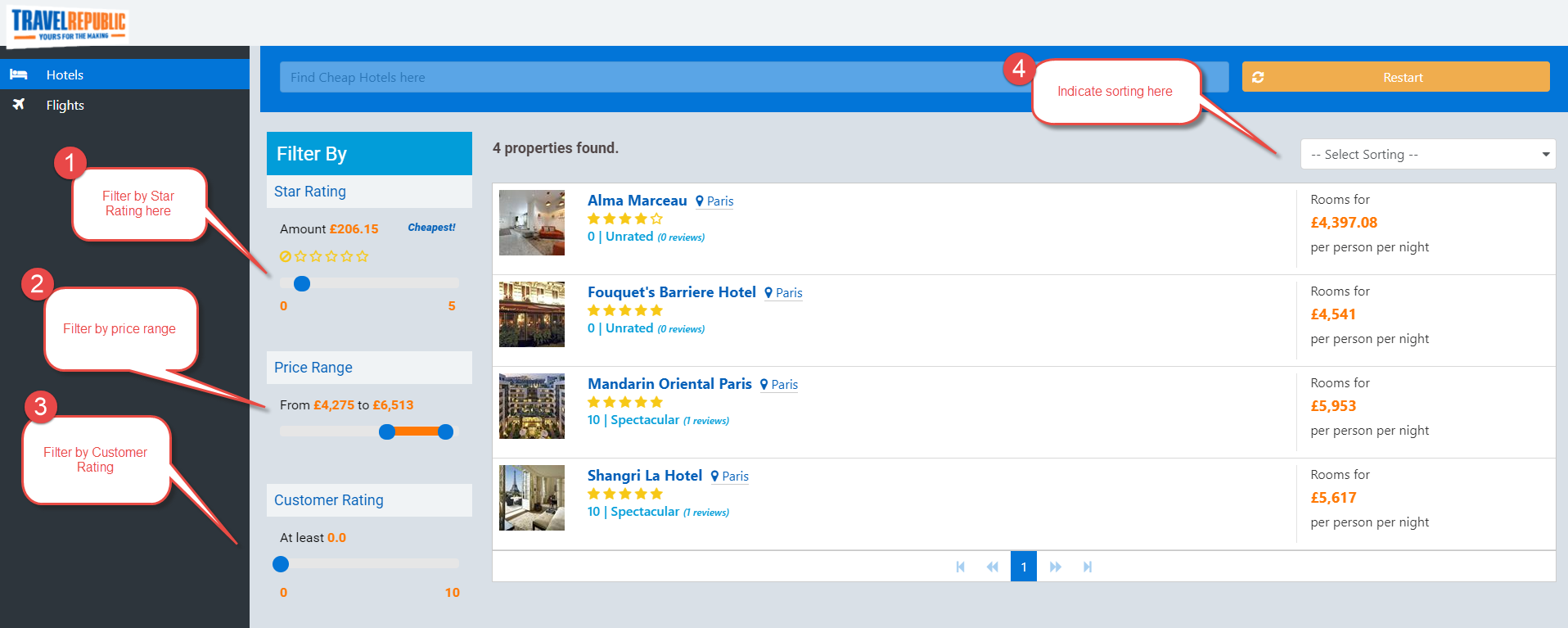


Or select from the suggested Hotels

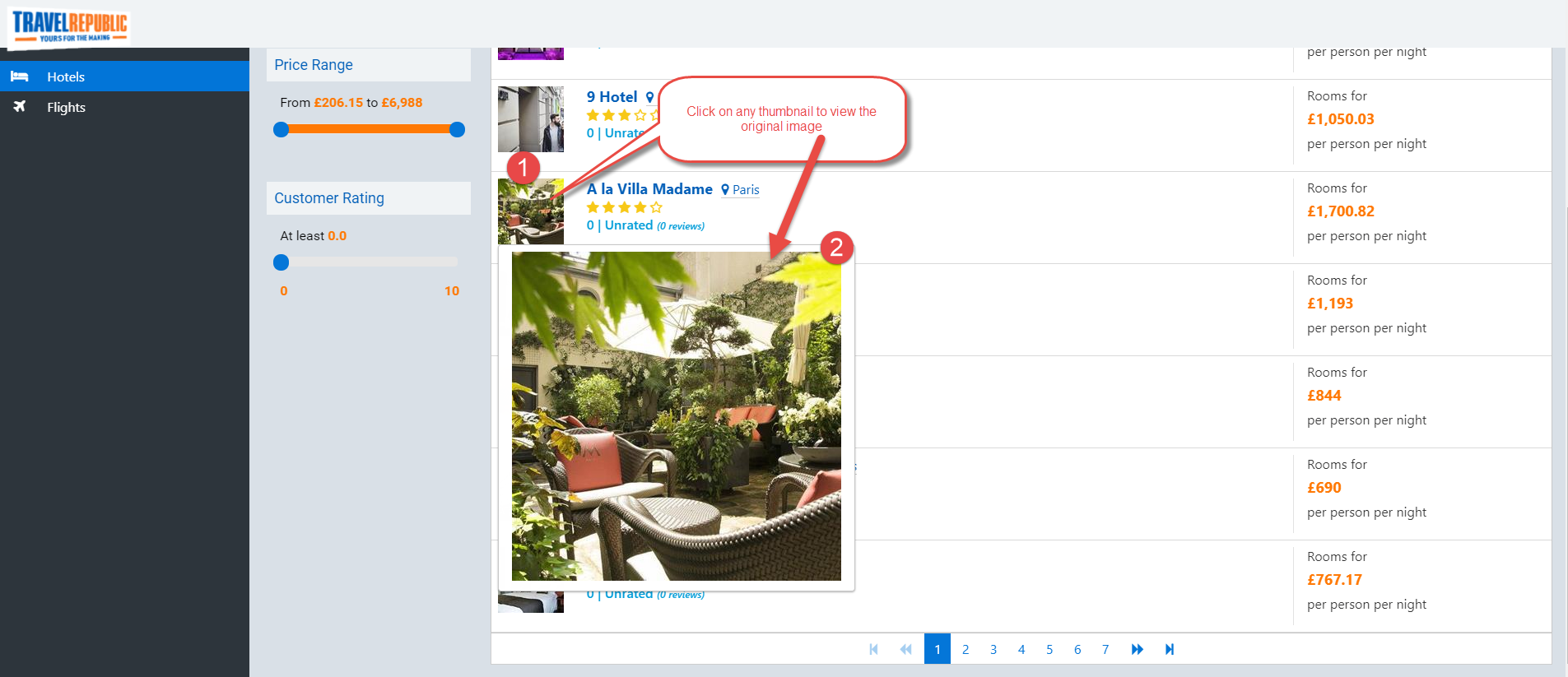


## Search Filter Page

After the search result, it can be sub-filtered in four ways:



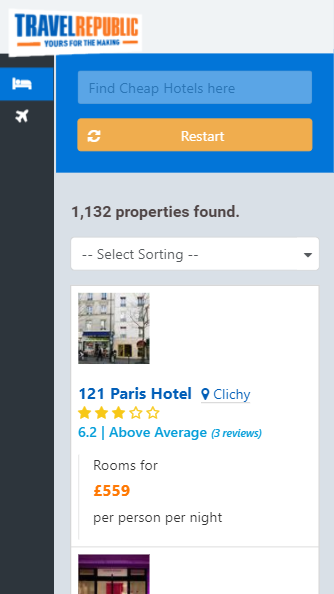
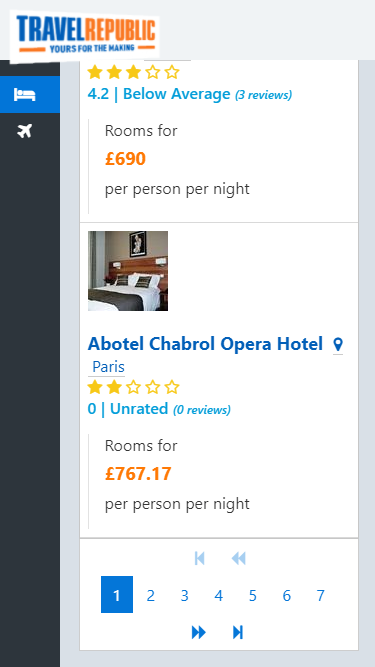
View a bigger picture by clicking on the thumbnail.



## Mobile View

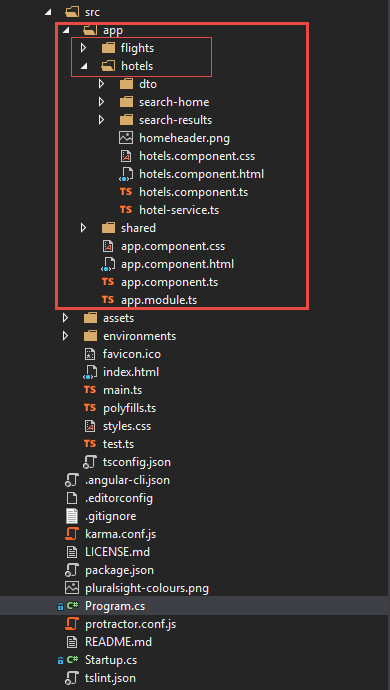
This sample page is designed for mobile browsing as well.

### Sample for iPhone6

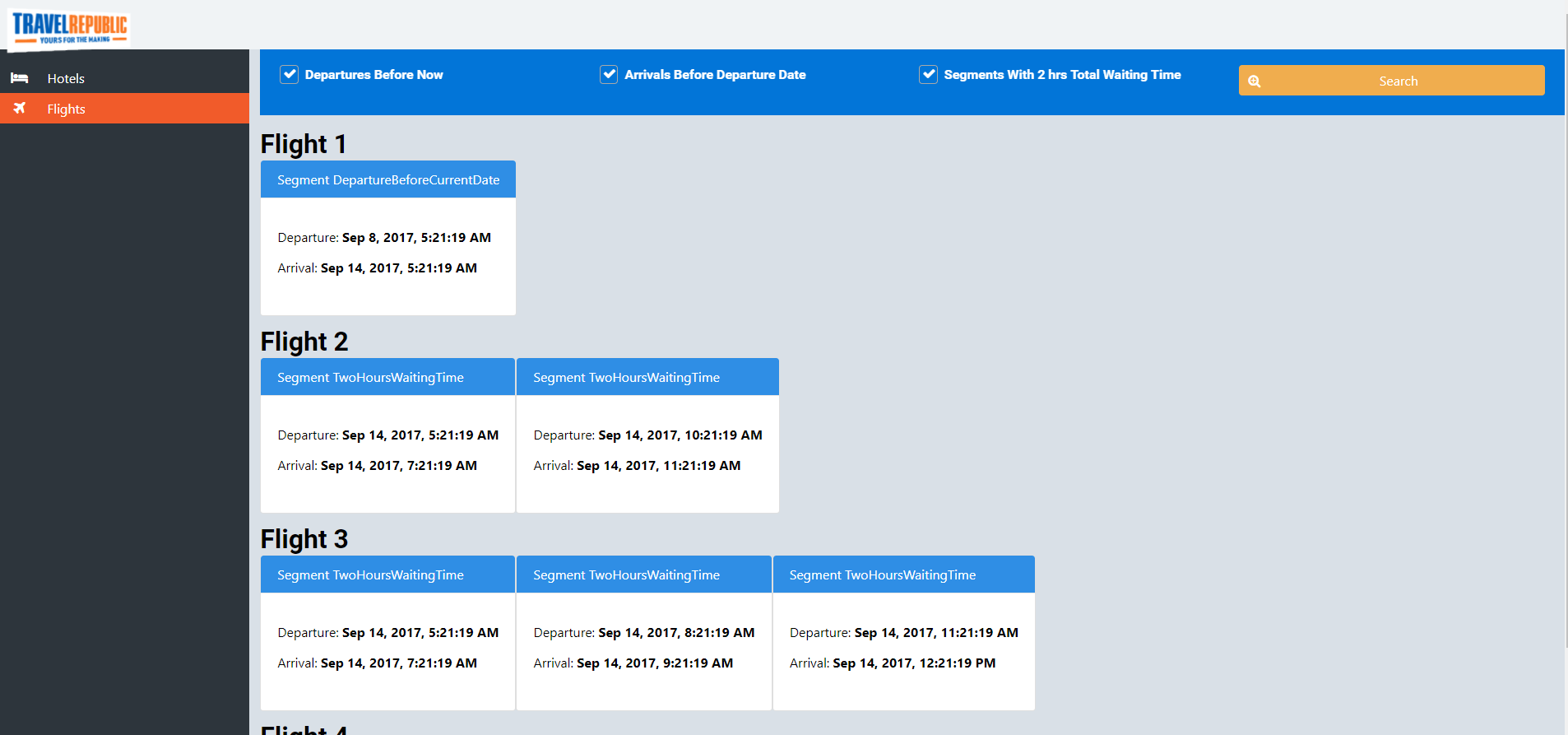


## Angular Artifacts

Hotels and Flight artifacts are located under the app folder as shown below:

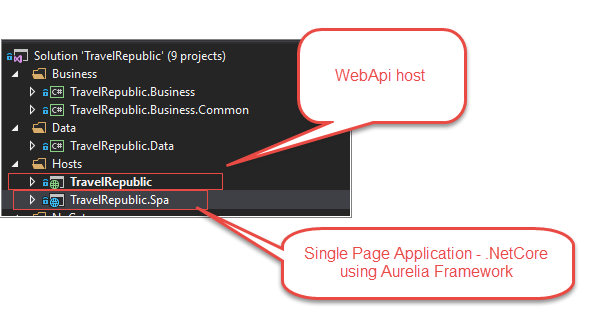


# UI for Task 2 – Flight segments

For easier checking if my truth table is correct. 

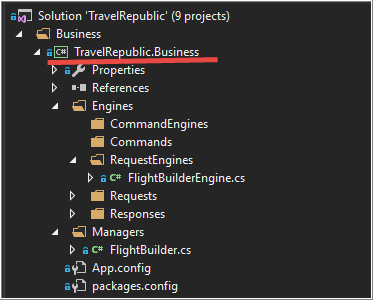
## Backend Host

Contains the WebApi projects



## Business

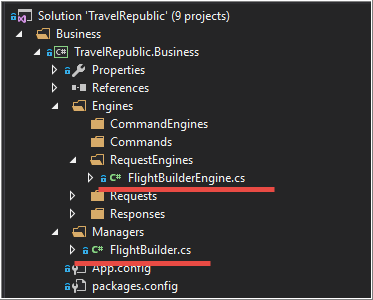
The purpose of this separate project is to capture the ‘business use cases’ into one location for reusability and maintainability. This is where most of the business logic is found.



## Request Engines

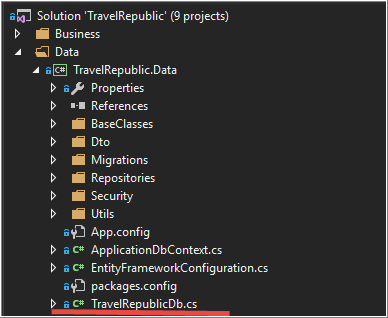
Use case that performs a functionality then returns a response.

Engines are the code representation of one specific business use case or functionality. It provides a clearer way of implementing a functionality through its ‘single responsibility’ characteristic.



## Data - EF Core

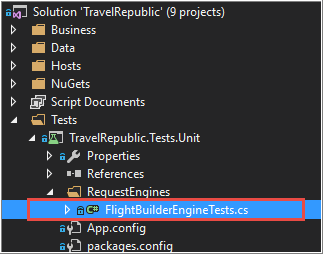
TravelRepublicDb is located in:



## 

## Unit Tests

Sample test:



## Integration Tests

Some tests to ensure that IoC/DI are working fine.

# Code repository

TFS Git/GitHub