

# Star Wars Travel Planner

By Alexandre Camillo Lins

This application shows all Star Wars ships required stops for resupply. You will only need to inform the distance.

## Summary

Repository Structure .....	4
Installation Notes .....	5
Console Application.....	5
Web Application and API.....	5
Instruction how to use .....	6
Console Application.....	6
Web Application.....	7
API .....	9
Using Browser: .....	9
Using Postman software: .....	10
Technical Details .....	11

## Repository Structure

The source code uploaded on GitHub ( ) has three main folders:

1. StarWarsTravelPlanner\_Console  
Contains the Console version of Star Wars Travel Planner.
2. StarWarsTravelPlanner  
Contains the Web version of Star Wars Travel Planner and the API.
3. Documentation  
Contains the necessary documentation about Star Wars Travel Planner.

## Installation Notes

### Important

The application uses the **.Net Core 2.1** using **.Net SDK v2.1.403**

Please, install .Net Core Runtime before run the application. The file can be downloaded in the follow link: <https://www.microsoft.com/net/download>

Direct Link:

<https://www.microsoft.com/net/download/thank-you/dotnet-runtime-2.1.5-windows-hosting-bundle-installer>

### Console Application

The Console application doesn't need to install. After compilation, it will generate a dll file and you need to execute the command *dotnet run* inside Command Prompt to run.

### Web Application and API

Web Application and API needs to run inside an internet service manager, like IIS. It will be able to run on Windows, Linus or MacOS due the framework used to develop.

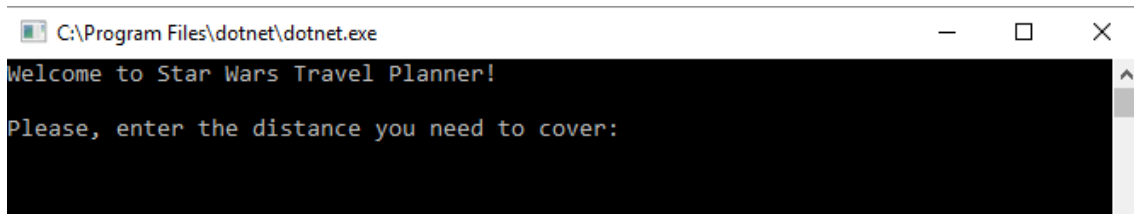
Before use, it is necessary run *npm install* command for web application to restore packages.

## Instruction how to use

You can run application using Visual Studio 2017 v.15.7 or higher.

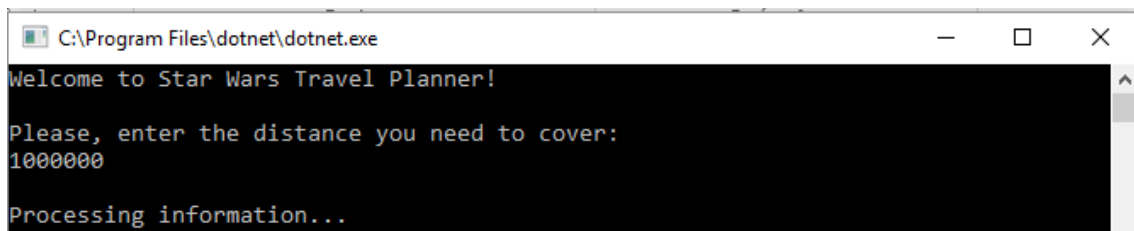
### Console Application

It is a CLI application. The instruction will be on the screen.



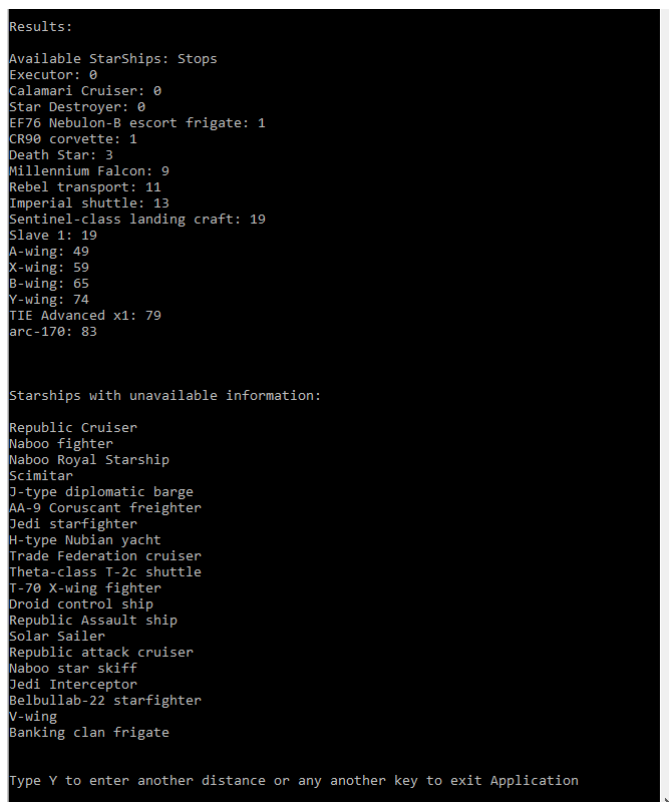
```
C:\Program Files\dotnet\dotnet.exe
Welcome to Star Wars Travel Planner!
Please, enter the distance you need to cover:
```

The first step will be enter the distance in MGLT and hit enter button.



```
C:\Program Files\dotnet\dotnet.exe
Welcome to Star Wars Travel Planner!
Please, enter the distance you need to cover:
1000000
Processing information...
```

After a while, the results will be send to screen. If you want to do another search, type “y” (case insensitive) and the distance again. Otherwise, type any key to exit application.



```
Results:
Available StarShips: Stops
Executor: 0
Calamari Cruiser: 0
Star Destroyer: 0
EF76 Nebulon-B escort frigate: 1
CR90 corvette: 1
Death Star: 3
Millennium Falcon: 9
Rebel transport: 11
Imperial shuttle: 13
Sentinel-class landing craft: 19
Slave 1: 19
A-wing: 49
X-wing: 59
B-wing: 65
Y-wing: 74
TIE Advanced x1: 79
arc-170: 83

Starships with unavailable information:
Republic Cruiser
Naboo fighter
Naboo Royal Starship
Scimitar
J-type diplomatic barge
AA-9 Coruscant freighter
Jedi starfighter
H-type Nubian yacht
Trade Federation cruiser
Theta-class T-2c shuttle
T-70 X-wing fighter
Droid control ship
Republic Assault ship
Solar Sailer
Republic attack cruiser
Naboo star skiff
Jedi Interceptor
Belbullab-22 starfighter
V-wing
Banking clan frigate

Type Y to enter another distance or any another key to exit Application
```

If you type wrong distance's value (alphanumeric characters for example), and want to exit application, just type "e" (case insensitive) to exit or inform the correct distance (numbers).

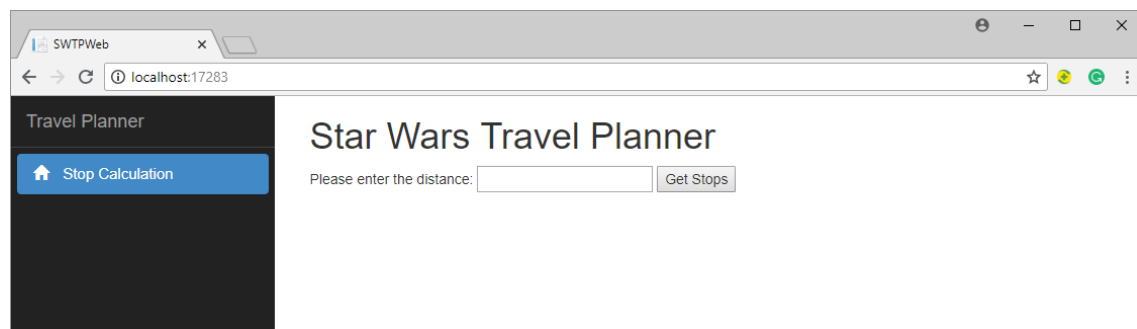
```
Please, enter the distance you need to cover:
asdf

Processing information...
I am sorry. Only numbers are accepted.

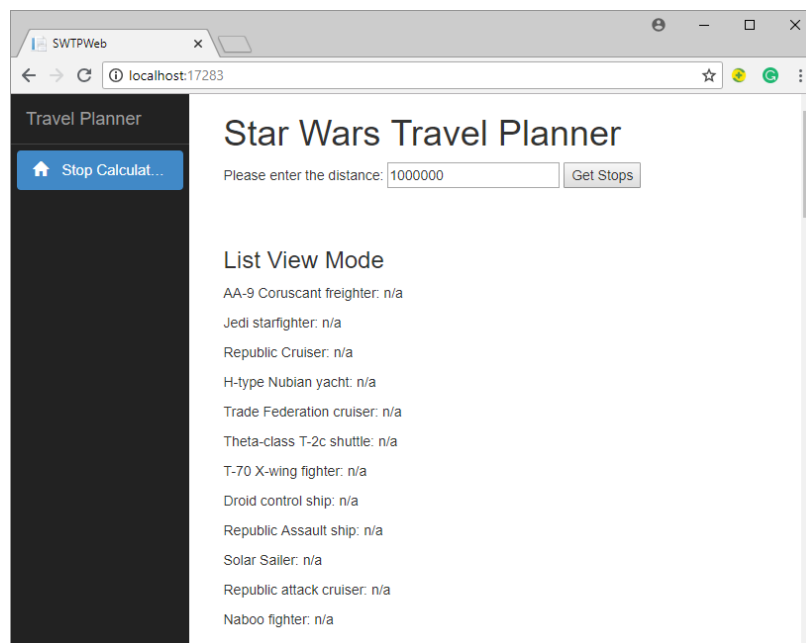
Type E to exit application
Please, enter the distance you need to cover:
```

## Web Application

After starting application on visual Studio, or deploy it into an information manager server, open the website on a web browser.



Insert the distance in MGLT on the input text and click on "Get Stops" button. The Results will be in the screen in two formats: a list view and below a table view:



StarShip	Stops
AA-9 Coruscant freighter	n/a
Jedi starfighter	n/a
Republic Cruiser	n/a
H-type Nubian yacht	n/a
Trade Federation cruiser	n/a
Theta-class T-2c shuttle	n/a
T-70 X-wing fighter	n/a
Droid control ship	n/a
Republic Assault ship	n/a
Solar Sailer	n/a
Republic attack cruiser	n/a
Naboo fighter	n/a
Naboo starship	n/a

When the StarShip doesn't have enough data to calculate how many stops, for example no information about consume or MGLT, the stop column will show "n/a" as answer.

If no information is provided on input text, an empty list will show at screen:

## Star Wars Travel Planner

Please enter the distance:

### List View Mode

### Table View Mode

StarShip	Stops
----------	-------



## API

The Web Application also provides an API to be used without the SWTP front end and provides the same results.

Browser or any application for test API, like Postman, can access it. To use it, use the URL mapping below:

---

*[BaseUrl]/api/starship/getstops/[Distance]/*

---

Where:

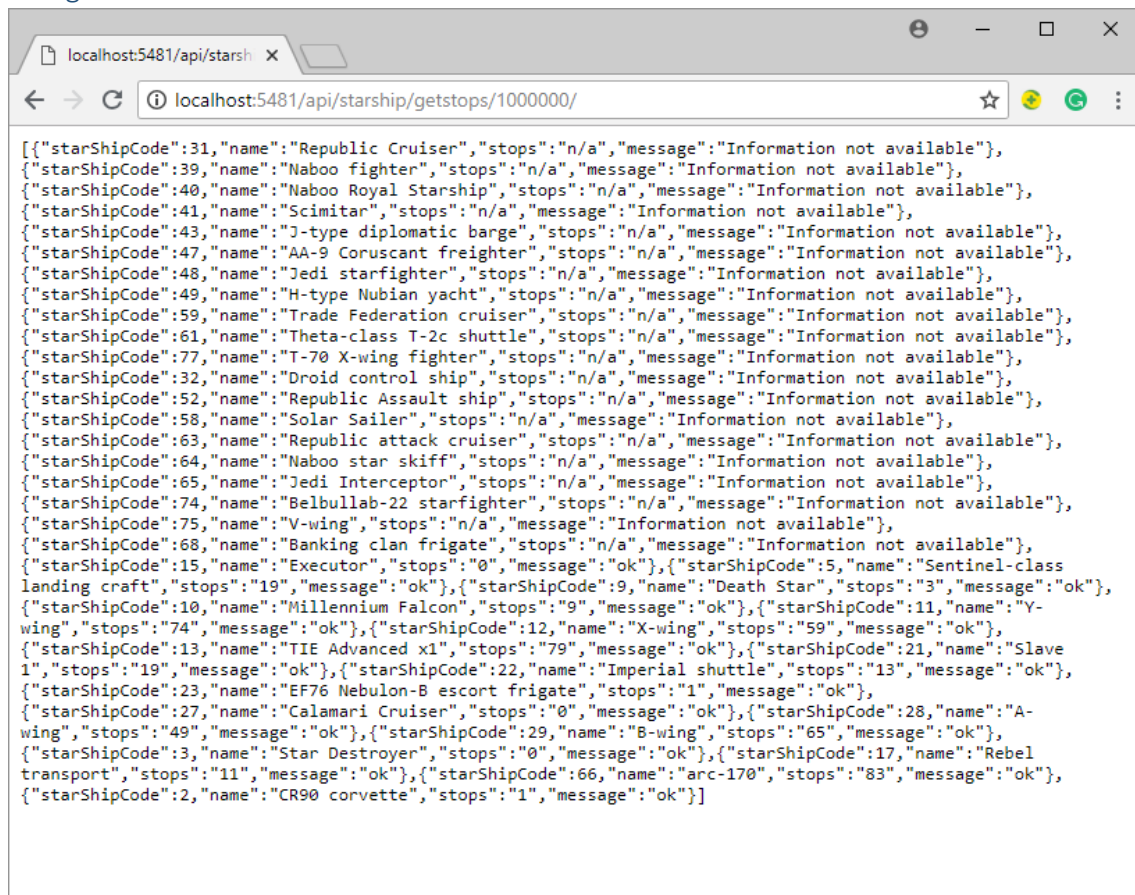
[BaseUrl]: the address the API is running. For example <http://localhost:5481>

[Distance]: Distance in Mega lights (MGLT)

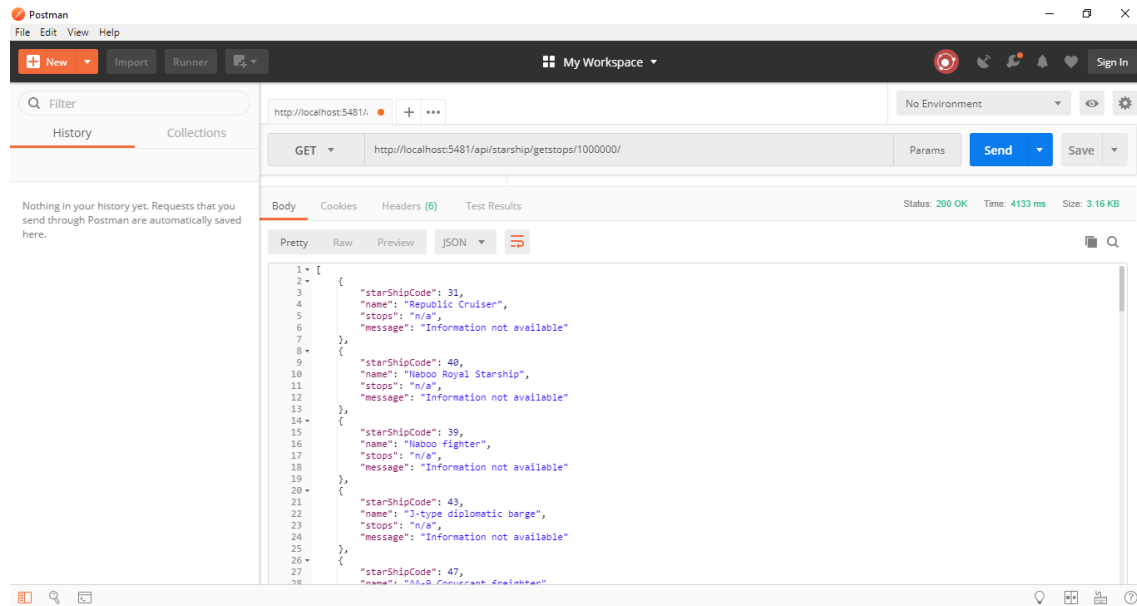
Method: GET

<http://localhost:5481/api/starship/getstops/1000000/>

Using Browser:



## Using Postman software:



## Technical Details

This application was build using .Net Core 2.1 Framework for backend and Console Application and Angular 5 for frontend.

It is using a DDD (Domain Driven Design) with all layers (View, Application, Domain, Infra and Crosscutting), split by folders. It is possible to configure this APIs to using docker.

The View Folder has API and Web application due uses the same implementation for other layers.

To integrate with DDD and get a better interface-implementation structure, it was used NativeInjector, to manage dependencies.

For Infra layer it was used a library available on SWAPI website (<https://swapi.co/>), called SharpTrooper that was created by Olcay Bayram (<https://github.com/olcay>). It was check all code to guarantee no risk for information and compatibility with required. The StarShip entity was complemented with missing properties.

It is already prepared to implement the EntityFramework, using Code First perspective, to use data stored in the database if http request is not available. (Not implemented in this version)

For Console application was used an n-tier design, without Dependency Injection due to monolithic perspective of architecture. The request uses SharpTrooper library (mentioned above) and it is not prepared for database perspective.