**Kloud Coding Test**

User Manual &Technical Reference

Contents

[Introduction 2](#_Toc6482873)

[Proposed Architecture 3](#_Toc6482874)

[Use Case 4](#_Toc6482875)

[Software Requirements 4](#_Toc6482876)

[Hardware Requirements 4](#_Toc6482877)

[Software Requirements 4](#_Toc6482878)

[Software Components 4](#_Toc6482879)

[Kloud\_CarsInfoProcessor.App 5](#_Toc6482880)

[Technology Used 5](#_Toc6482881)

[Structure 5](#_Toc6482882)

[How to run Kloud\_CarsInfoProcessor.App 5](#_Toc6482883)

[Codingtest.kloud.com.au – Replica RESTful API 7](#_Toc6482884)

[Technology Used for Development 7](#_Toc6482885)

[Structure 7](#_Toc6482886)

[Controllers 7](#_Toc6482887)

[Error Log & Exception Handling 8](#_Toc6482888)

[How to run API 9](#_Toc6482889)

[Swagger Testing Tool 9](#_Toc6482890)

[Kloud\_CarsInfoProcessor Unit Test Project 11](#_Toc6482891)

[Technology Used 11](#_Toc6482892)

[Structure 11](#_Toc6482893)

[Available Test 11](#_Toc6482894)

# Introduction

Kloud coding challenge consists of two tasks as follows,

* An App that produces a list of the owners’ names, grouped by car’s brand alphabetically, and sorted by their car’s colour alphabetically.
* An RESTful API service which is a replica that returns the same results as the API provided.

Sampled Output:

**Toyota**

* George
* Petra

**Volkswagen**

* Belinda
* Stephanie
* Nathan

In order to show my programming skills I have introduced more features to Replica API.

# Proposed Architecture

The application uses loosely-coupled, dependency-inverted architecture like clean architecture.

CodingTest.Kloud.com.au - Replica

Kloud\_CarsInfoProcessor.App

Kloud\_CarsInfoProcessor.Test

Cars.json

Models

Controller Layer

Service Layer



CodingTest.Kloud.com.au

# Use Case



# Software Requirements

## Hardware Requirements

* 1.8 GHz or faster processor. Dual-core or better recommended
* 2 GB of RAM; 4 GB of RAM recommended (2.5 GB minimum if running on a virtual machine)
* Hard disk space: up to 130 GB of available space, depending on features installed; typical installations require 20-50 GB of free space.
* Hard disk speed: to improve performance, install Windows and Visual Studio on a solid-state drive (SSD).
* Video card that supports a minimum display resolution of 720p (1280 by 720); Visual Studio will work best at a resolution of WXGA (1366 by 768) or higher.
* Windows 10 version 1507 or higher: Home, Professional, Education, and Enterprise (LTSC and S are not supported)
* Windows Server 2016: Standard and Datacenter

## Software Requirements

* Visual Studio 2017 [Enterprise or Professional Edition]
* .Net Framework 4.7.2
* .Net Core SDK 2.2
* Google Chrome
* Notepad

# Software Components

This code test consists of following 3 projects,

* Kloud\_CarsInfoProcessor.App
* codingtest.kloud.com.au - Replica
* Kloud\_CarsInfoProcessor.Test

# Kloud\_CarsInfoProcessor.App

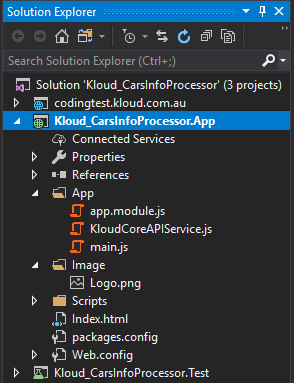
This application is developed using AngularJS and JavaScript which produces a list of the owners’ names, grouped by car’s brand alphabetically, and sorted by their car’s colour alphabetically. This application UI allows the user to connect both <http://codingtest.kloud.com.au/api/cars> or Replica API <http://localhost:9912/api/cars>.

## Technology Used

* Visual Studio 2017
* HTML
* AngularJS
* JavaScript

## Structure

The project consists of HTML, JavaScript files. Below picture shows the file structure of Test project.



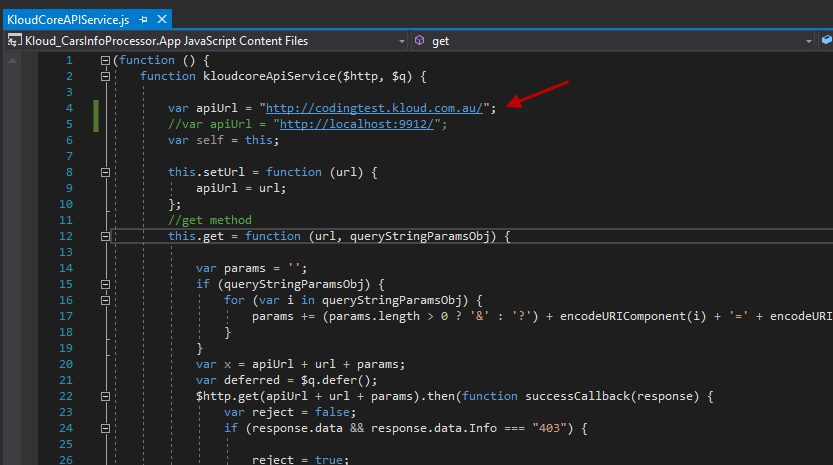
## How to run Kloud\_CarsInfoProcessor.App

Following steps illustrate, how to run the application in IISExpress,

1. Open the Kloud\_CarsInfoProcessor.sln in Visual Studio 2017

2. Make sure Kloud\_CarsInfoProcessor.App.csproj is loaded.

3. Update <http://codingtest.kloud.com.au/> URL in KloudCoreAPIService.js.

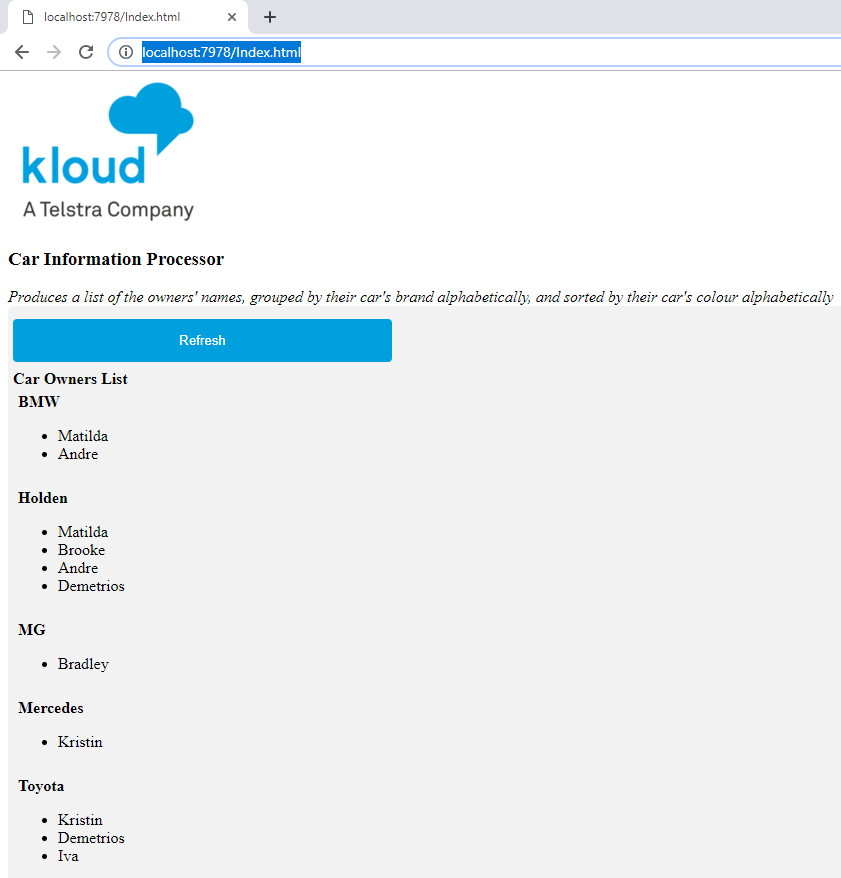


4. The user can switch between actual and replica service by updating the URL as shown above.

5. Build Kloud\_CarsInfoProcessor.App project.

5. Click Run with IISExpress(Google Chrome).

6. Google Chrome opens <http://localhost:7978/Index.html>.



Click “Refresh” button to update the list with latest content.

## Codingtest.kloud.com.au – Replica RESTful API

Codingtest.kloud.com.au Restful API is the replica of actual <http://codingtest.kloud.com.au/api/cars>. This API process the request and provide cars json data as output. The API is developed using .Net Core SDK, hence it can run in both Windows and Linux environment.

This API consists of two main controllers such as,

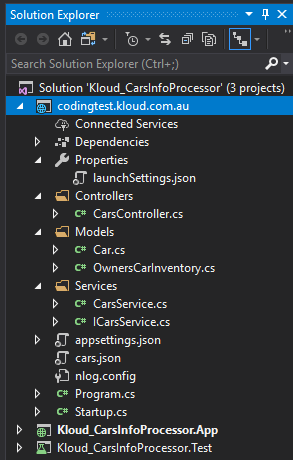
**CarsController** - handle requests to return cars json data.

## Technology Used for Development

* Visual Studio 2017
* ASP.Net Core Web Application
* .Net Core 2.2 SDK
* NLog for .Net Core
* Swashbuckle for Swagger
* Google Chrome
* IIS Express

## Structure

The API project consist of Controllers, Services, Models, Files & Log repositories. Below picture shows the file structure of RestAPI project.



### Controllers

#### Cars Controller

CarsController hosts the Restful methods to return cars json data. The requests are automatically routed to the relevant methods in this class. External application can access the public methods available in this class. This controller class forward the requests to the CarsService class which host the business logic and hidden from external users.

CarsController class consists of following methods,

**Get**

This method returns entire cars json content as ActionResult<string> type the caller which contains json response. This method can be called using CarsController HTTPGet method without any parameters.

**GetJsonByOwner**

This method returns list of specific owners’ cars details in json format. This method can be called using CarsController HTTPGetJsonByOwner method with owner name as parameters.

**GetOwnersByColour**

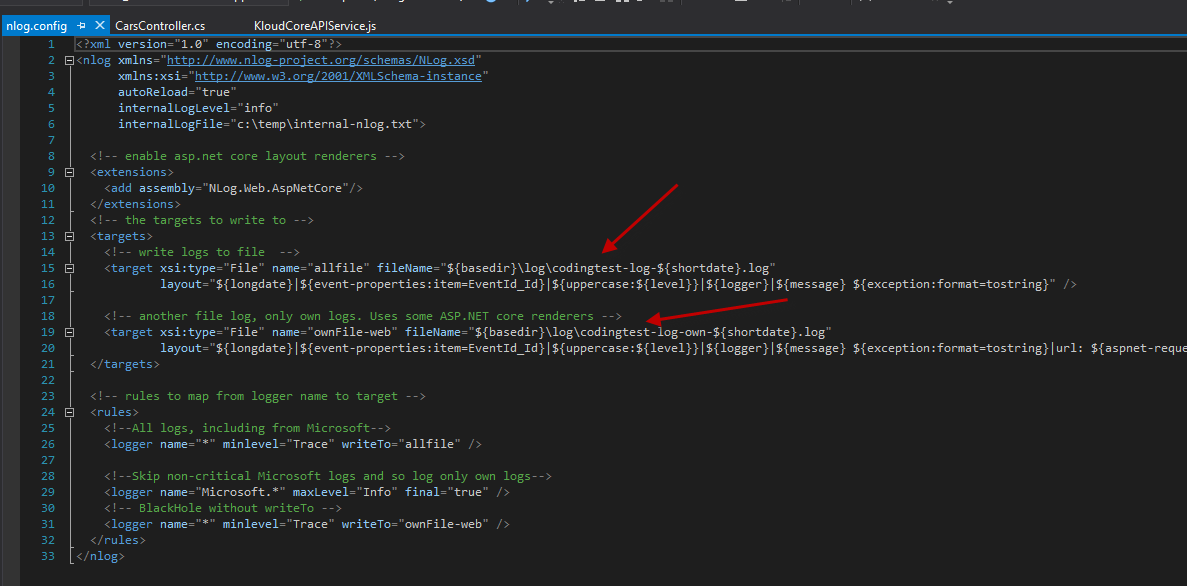
This method returns list of specific owner names based on their car colour. This method can be called using CarsController HTTPGetOwnersByColour method with colour as parameters.

**GetOwnersByBrand**

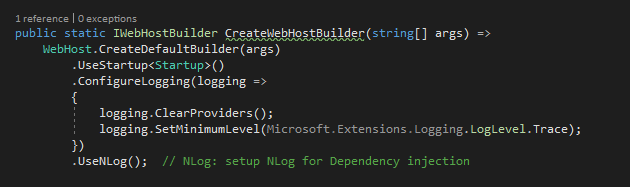
This method returns list of specific owner names based on their car brand. This method can be called using CarsController HTTPGetOwnersByBrand method with brand as parameters.

## Error Log & Exception Handling

Nlog is used to record exception log occurred in API. Nlog configurations are available in nlog.config file and please update the log file target path to server environment as shown below.



Following image shows NLog configuration in Program.cs.



## How to run API

Following steps illustrate, how to run the API in IISExpress,

1. Open the Kloud\_CarsInfoProcessor.sln in Visual Studio 2017

2. Make sure codingtest.kloud.com.au.csproj is loaded.

3. Build codingtest.kloud.com.au project and make sure it is the startup project.

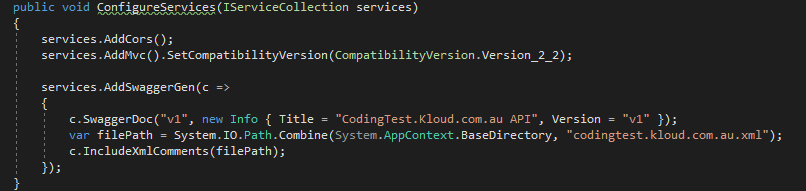
5. Click Run with IISExpress(Google Chrome).

6. Google Chrome opens <http://localhost:9912/swagger/index.html>. This page shows all methods with help notes and comments.

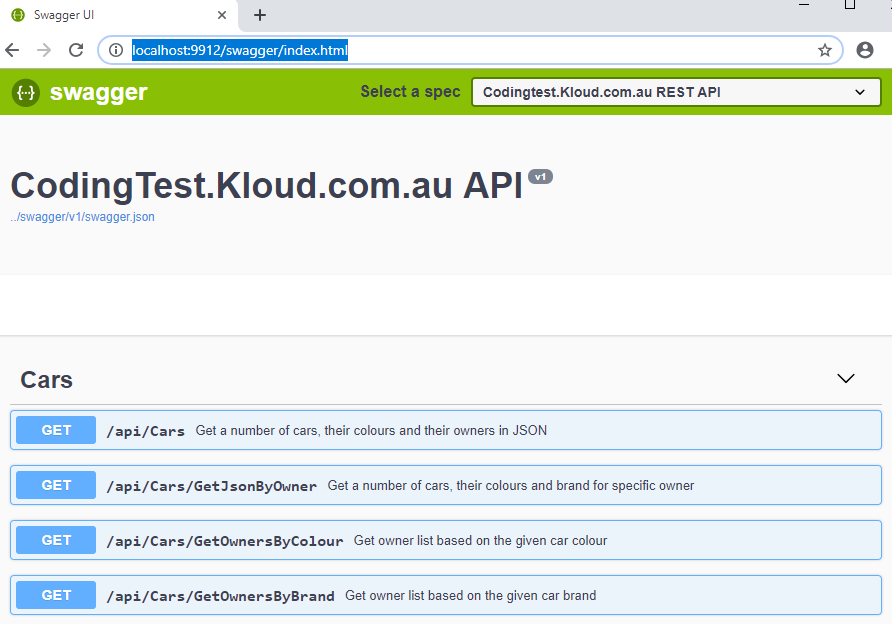
7. User can test the API by executing from swagger UI.

## Swagger Testing Tool

Following image shows the Swagger configuration in Startup.cs

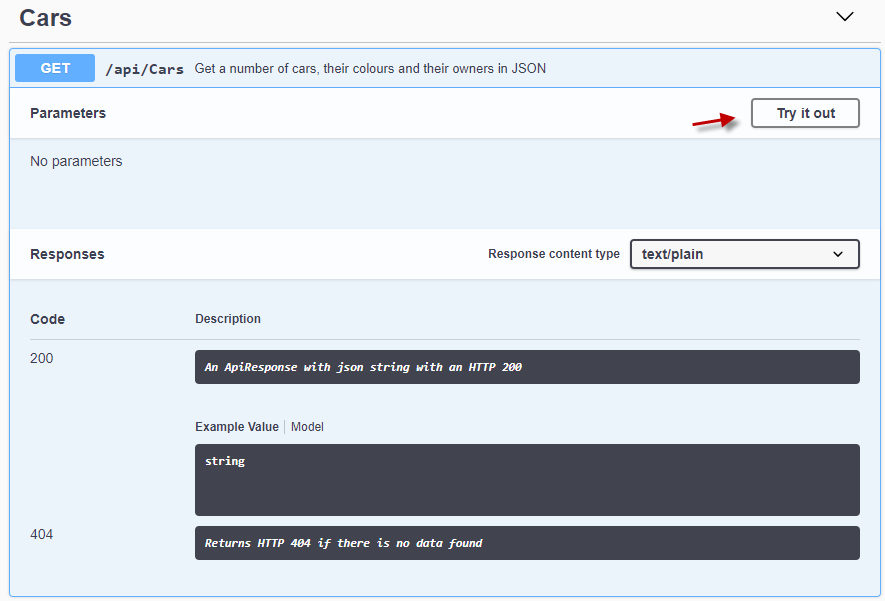


Codingtest.kloud.com.au Restful API Swagger page.

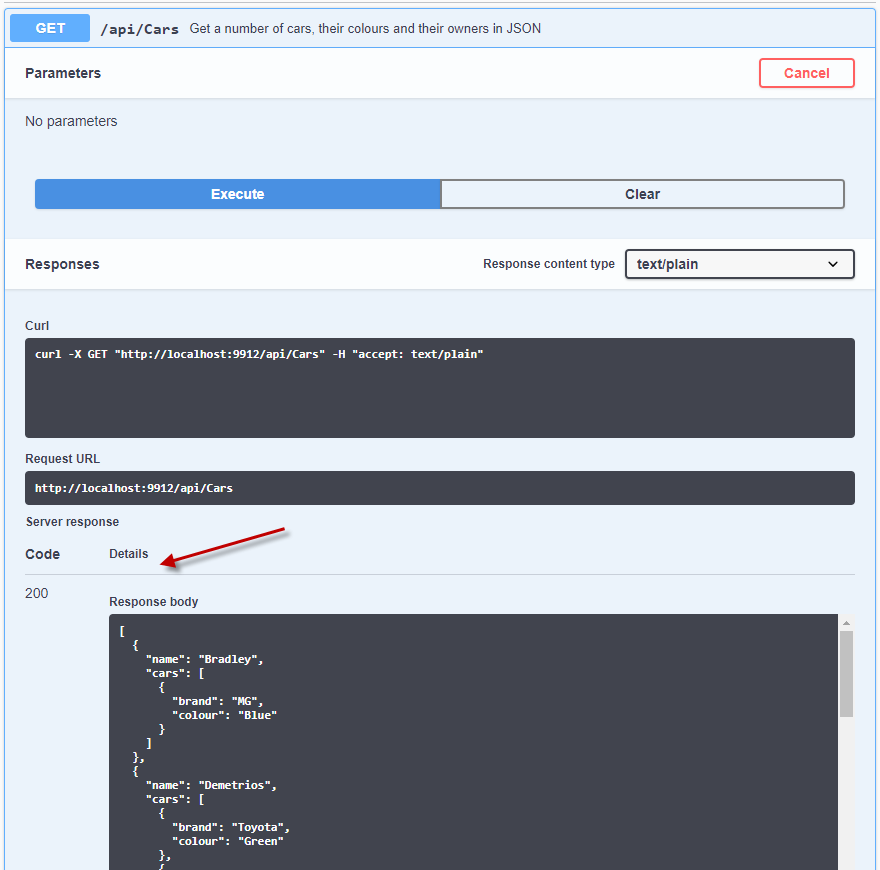


Testing codingtest.kloud.com.au Rest API using Swagger

1. Run codingtest.kloud.com.au Rest API which opens <http://localhost:9912/swagger/index.html>. This page shows all methods with help notes and comments.
2. Expand Get method in Cars Controller
3. Click Try it out button



1. Click Execute button.
2. Response will be displayed in Server Response section as shown below.



## Kloud\_CarsInfoProcessor Unit Test Project

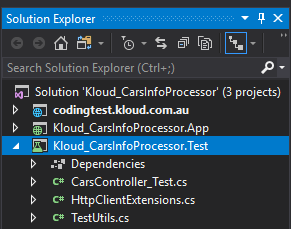
License Calculator MS Test project consists of unit testcase to codingtest.kloud.com.au Restful API.

### Technology Used

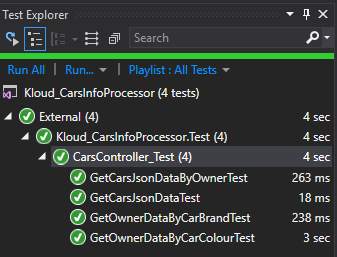
* Visual Studio 2017
* ASP.Net Core Web Application
* .Net Core 2.2 SDK

### Structure

The project consists of Tests, Utilities and HTTPClient extensions. Below picture shows the file structure of Test project.



### Available Test



Each test contains of several test case using DataTest methods. Example,

