# Web API Overview

ASP.NET Web API from the beginning was designed to be a service-based framework for building **RESTful** (**Representational State Transfer**) services. It is essentially the MVC framework minus the **“V”** (view).

These services can be called by any technology, not just those under the Microsoft umbrella, so we might use it with a front-end framework such as JQuery or React, or Angular.

Calls to a Web API service are based on the core HTTP verbs (**Get, Put, Post, Delete**) through a **URI** (Uniform Resource Identifier) such as the following:

<http://www.mysite.com:33826/api/category>

If this looks like a URL (Uniform Resource Locator), it’s because it is! A URL is simply a URI that points to **physical** resource on a network.

Calls to Web API use the **HTTP** (Hypertext Transfer Protocol) scheme on a particular host (in this example **www.mysite.com**), on a specific port (**33816**), followed by the path (**api/category**) and optional query and fragment (not shown in this example).

Note that the port for HTTP calls defaults to **80** if one isn’t specified. Web API calls can also include text in the body of the message

*API’s are used to push data around the net and to allow developers to connect in and consume that information. Think of an API like an electrical socket you can plug anything that needs power into.*

**An API that delivers Electricity A front end that consumes the electricity**



This analogy falls down when you realise that you can send data up the API, back into the plug.

[ProgrammableWeb.com](https://www.programmableweb.com/) is a website that provides API’s for you to use. Here are all the New Zealand ones. <https://www.programmableweb.com/category/all/apis?keyword=New%20Zealand>

Here is an API that allows you to put in a vehicles registration Number and see all the following details.

Make, Model, Engine Size, Fuel Type, Body Style, VIN number, Engine Code, Number of seats, Colour, Country of assembly, Stolen indicator, Odometer reading (last and current), Warrant of Fitness, and Import status. Available in JSON, SOAP, and XML formats with API Key. <https://www.programmableweb.com/api/new-zealand-car-registration-number-lookup>

Here is their documentation for it <http://nz.carregistrationapi.com/>

This website may be using this API <https://www.carjam.co.nz/>Try it out and see what you get.

# How to Create a Web API in ASP.NET Core

1. Select the ASP.net Core Web API

Graphical user interface, text, application, website

Description automatically generated

1. Name the API

Graphical user interface, application

Description automatically generated

1. Don’t Change any option here

Graphical user interface, application

Description automatically generated

1. Delete files related to weather forecast.

Graphical user interface, application

Description automatically generated

1. Create folder with name Models in the project

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Add a Tutor class under Models folder

A screenshot of a computer

Description automatically generated

1. Right click on Controllers Folder, then click on Add option and choose Controller from that. It will open below mentioned window. From there select API option from Common (Left side) and then choose API Controller with actions, using Entity Framework. Click Add option.

Graphical user interface, text, application, Word

Description automatically generated

1. Select Tutor under Model class option and name of the context under Data context class. And click Add option.

Graphical user interface, text, application

Description automatically generated

1. It will add Class under Controllers folder as well as create a new folder with name i.e. Data

Graphical user interface, text, application

Description automatically generated

1. Write ‘add-migration’ command under package manager console to create the Migration files under migration folder.

Graphical user interface, text

Description automatically generated

A picture containing text, screenshot, computer, computer

Description automatically generated

1. Write ‘update-database’ command in package manager console to migrate those files into local database.

A picture containing text, screenshot, computer, computer

Description automatically generated

A picture containing text, screenshot, computer, indoor

Description automatically generated

We have got the 4 CRUD methods. However, they are given different names

|  |  |  |
| --- | --- | --- |
| **GET** | READ information. - get **all** the data, | // GET: api/Tutors |
| **GET** | or get a **single** entry | // GET: api/Tutors/5. |
| **POST** | CREATE a **new** entry | // POST: api/Tutors/5 |
| **PUT** | UPDATE an **existing** entity | // PUT: api/Tutors/5 |
| **DELETE** | DELETE | // DELETE: api/Tutors/5 |

So I can take the API in green from the Get Method, **//GET:** api/Tutors run the program and paste it into the address bar after the URL <https://localhost:44382/api/Tutors>it will send me down all the data. In JSON format.

Graphical user interface, text, application

Description automatically generated

Our JSON from the Get returns this data

So we can see that our fields are

* Tutor Name --- "Tutor\_Name":"Jatinder Singh",
* Email --- "Email":"Jatinder@nzse.ac.nz",
* Mobile --- "Mobile":"2233223025",
* Address --- "Address":"Takanini"