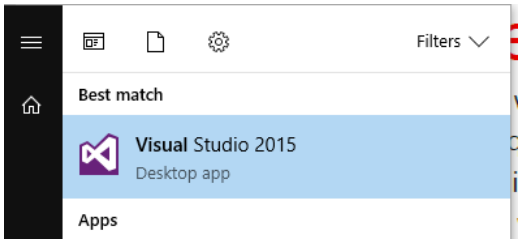
WEB API - RESTFULL Service Call

Enabling Access to External Schema

start the IDE

Open Visual Studio 2015/2013

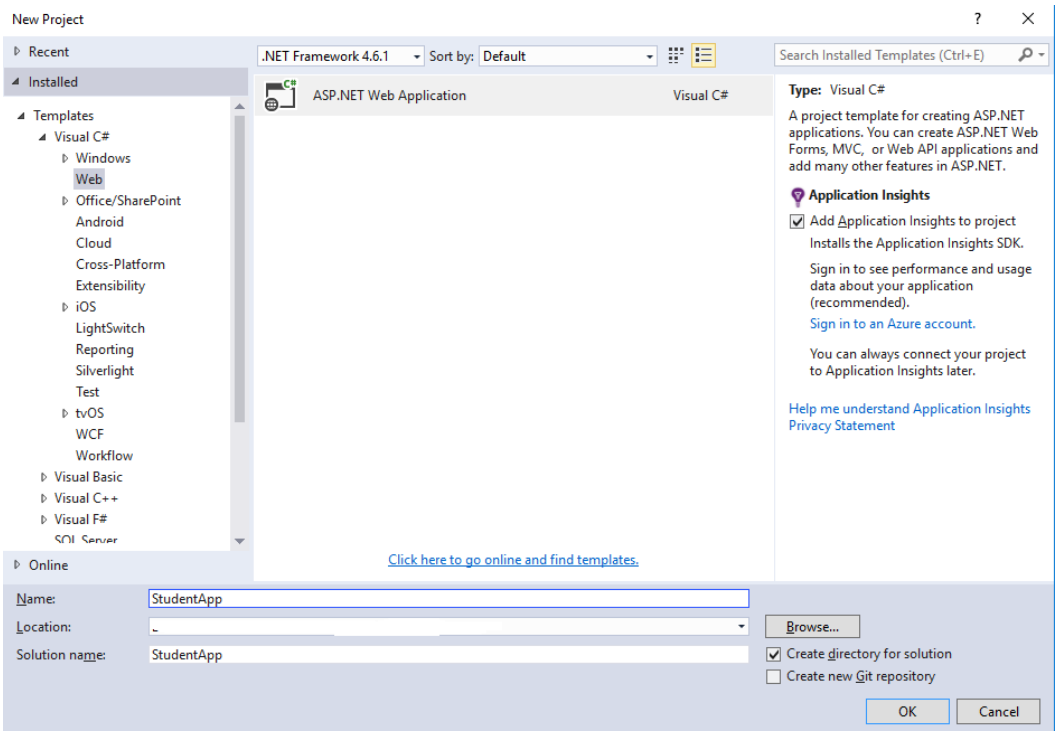
Start -> Visual Studio-> Visual Studio 2015



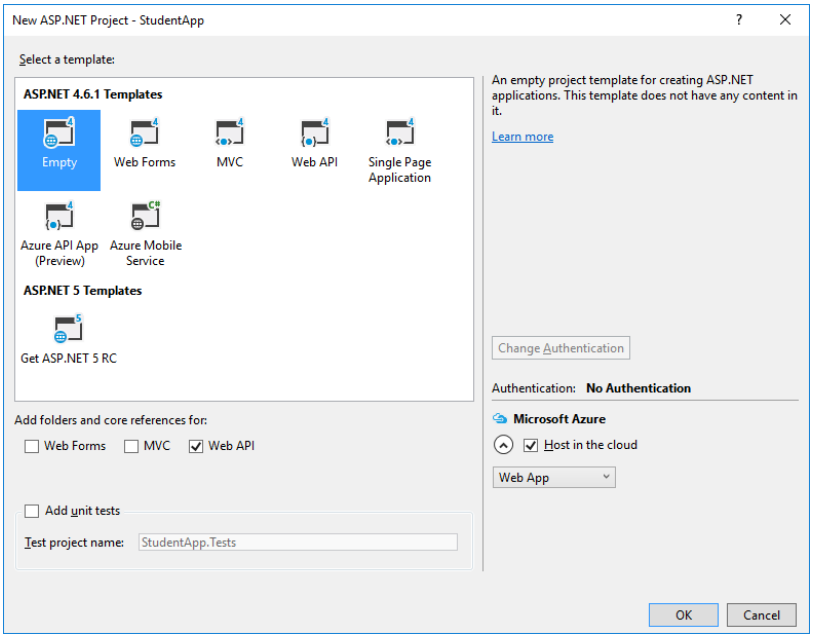
Start Web API

Start Visual Studio and select **New Project** from the Start page. Or, from the **File menu**, select **New** and then **Project**.

In the Templates pane, select **Installed** Templates and expand the Visual C# node. Under Visual C#, select **Web**. In the list of project templates, select **ASP.NET Web Application**. Name the project "StudentApp" and click OK.



In the **New ASP.NET Project** dialog, select the **Empty** template. Under "Add folders and core references for", check **Web API**. Click **OK**.



Then it will open up the Project to work with :D

NOTE:

It opens the project in a way that it works in MVC architecture. View is not available here. Idea behind this is to do Server side task. Clients which is in view may be a different application. But for today, we will create a html client in same project which will call these service in RESTFUL manner.

Start the project

Project Idea:

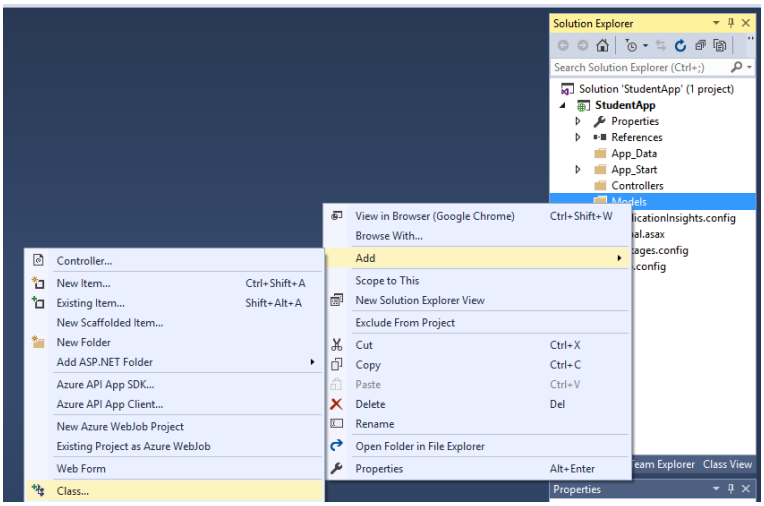
You are going to create a service which has few student details and you are going to access it from thin client (HTML page). At the initial step (page load), you are going to load all the students in a list and display in a page. Then you add a functionality to search a student in existing list and show the details. Both of above are “GET” calls to the API. Then as the final Step you are calling to method which accept only “POST” calls.

Further,

We can check how GET/POST calls are working in NETWORK layer (using Browser functionality)

Creating a MODEL File.

click the **View** menu and select **Solution Explorer**. In Solution Explorer, right-click the Models folder. From the context menu, select **Add** then select **Class**.

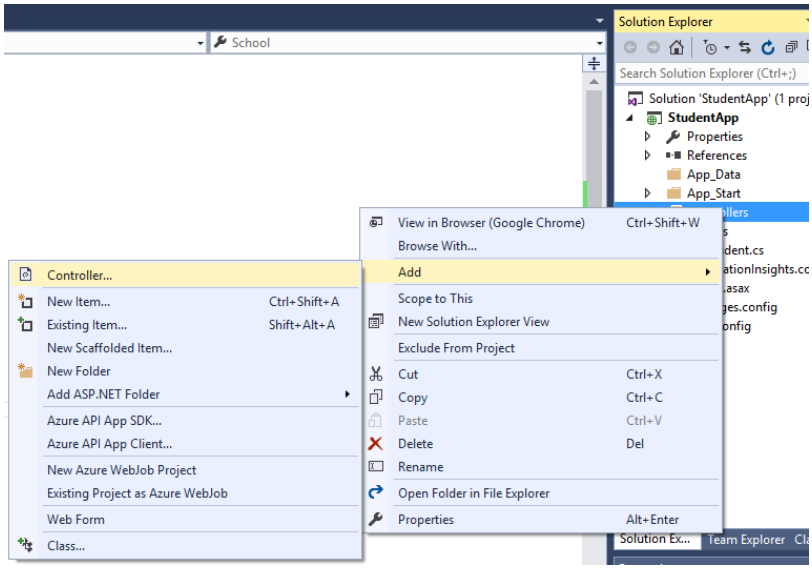


Name the class "Student". Add the following properties to the Student class

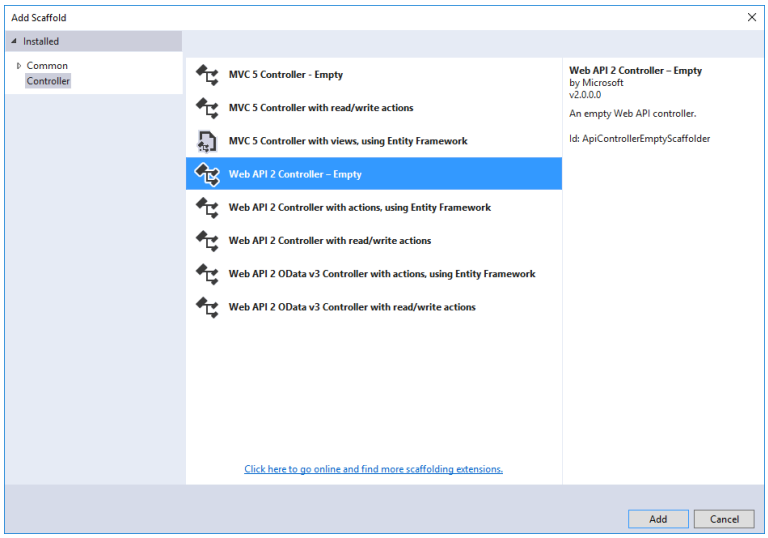
```
Student.cs
StudentApp
StudentApp.

1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Web;
5
6   namespace StudentApp.Models
7   {
        0 references
8       public class Student
9       {
            0 references | 0 exceptions
10          public int Id { get; set; }
            0 references | 0 exceptions
11          public string Name { get; set; }
            0 references | 0 exceptions
12          public string School { get; set; }
            0 references | 0 exceptions
13          public string Grade { get; set; }
            0 references | 0 exceptions
14          public string GuardianName { get; set; }
15      }
16  }
```

Now your Student Model is ok. Then create a controller file.

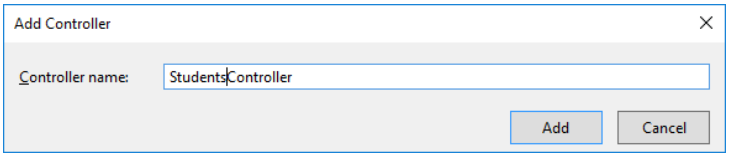
Create Controller



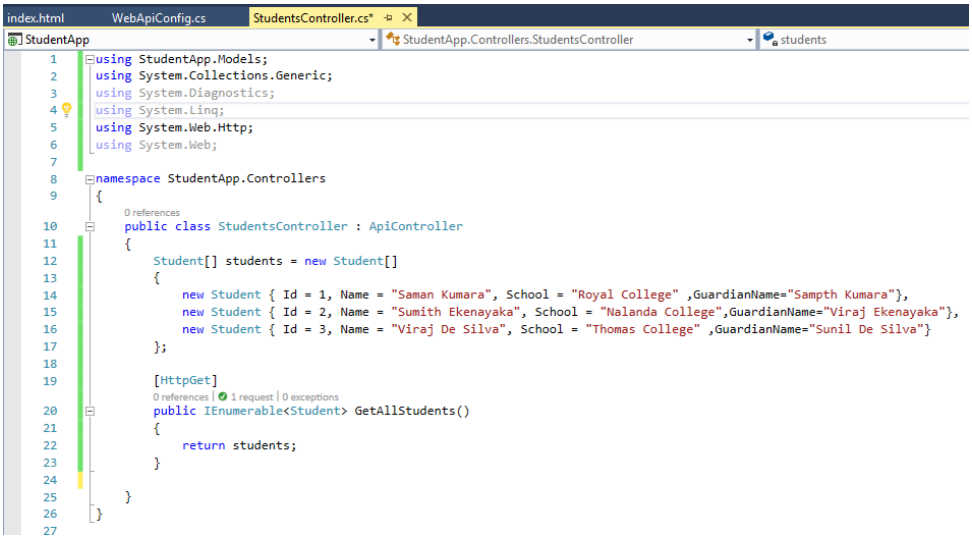
Select Web API 2 Controller -Empty



Name it as StudentsController



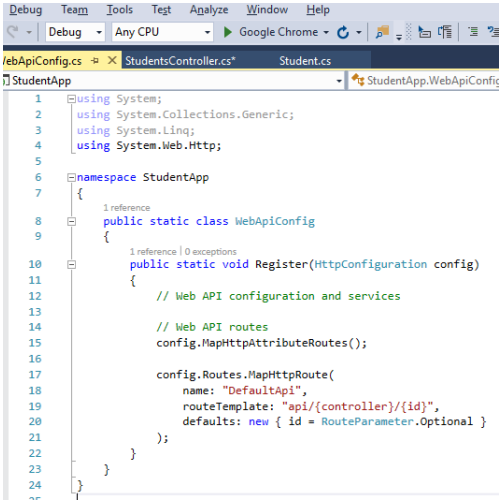
In **Solution Explorer**, right-click the Controllers folder. Select **Add** and then select **Controller**.



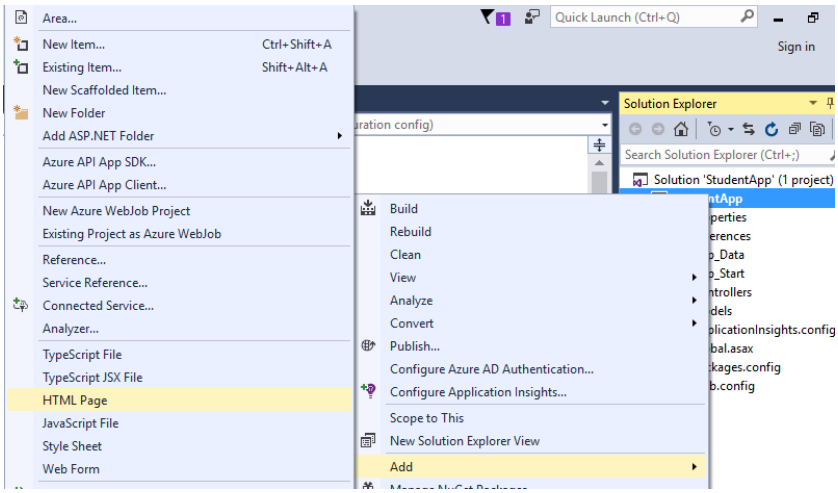
Additional Note:

Go to App_Start -> WebApiConfig.cs

This is the place where it manages the routings



Now Your WEB API is ready. So, create a html page to access this.



Add Following code

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Student Manage App</title>
</head>
<body>
    <div>
        <h2>All Student List</h2>
        <ul id="students" />
    </div>
    <div>
        <h2>Search by ID</h2>
        <input type="text" id="studentId" size="5" />
        <input type="button" value="Search" onclick="find();" />
        <p id="student"></p>
    </div>

    <div>
        <input type="button" value="Post" onclick="postMethodcall();" />
    </div>

    <script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-2.0.3.min.js"></script>
    <script>
        var uri = 'http://localhost:56393/api/students';
        $(document).ready(function () {
            jQuery.ajax({
                type: "GET",
                url: uri,
                dataType: "json",
                cache: false,
                crossDomain: true,
                processData: true,
```

```
                success: function (data) {
                    $.each(data, function (key, item) {
                        $('<li>', { text: formatItem(item) }).appendTo($('#students'));
                                });
                },
                error: function (XMLHttpRequest, textStatus, errorThrown) {
                    alert("error");
                }
            });

        });

        function formatItem(student) {
            return 'Student Id : ' + student.Id + ' | Student Name : ' + student.Name + '
|  School : ' + student.School + ' |  Guardian Name : ' + student.GuardianName;
        }
    </script>
</body>
</html>
```

Now Run and See you can see the students.

Please make sure to change the port name accordingly

“'http://localhost:56393/api/students';”

To Enable Search
Add this function to Script

```
function find() {
            var id = $('#studentId').val();
            $.getJSON(uri + '/GetStudent/' + id)
                .done(function (data) {
                    $('#student').text(formatItem(data));
                })
                .fail(function (jqXHR, textStatus, err) {
                    $('#student').text('Error: ' + err);
                });
        }
```

Add this method to Controller

```
[HttpGet]
public IHttpActionResult GetStudent(int id)
{
   var student = students.FirstOrDefault((s) => s.Id == id);
   if (student == null)
   {
       return NotFound();
   }
   return Ok(student);
}
```

Check your Search Function is work or not.
If not, you are having the problem with routing. So Add this to enable routing.

Go to WebApiConfig.cs and add below under Register method.

```
config.Routes.MapHttpRoute(
                name: "ForStudentSearch",
                routeTemplate: "api/{controller}/{action}/{id}",
                defaults: new { id = RouteParameter.Optional }
            );


To Check Post Calls

Add this function to Script in index.html

function postMethodcall() {
            jQuery.ajax({
                type: "POST",
                url: uri + '/myPostmethod',
                dataType: "json",
                cache: false,
                crossDomain: true,
                processData: true,

                success: function (data) {
                    alert("Post method is called...");
                },
                error: function (XMLHttpRequest, textStatus, errorThrown) {
                    alert("Error : Error in Post Call...");
                }
            });
        }
```

Add this method to controller

```
[HttpPost]
public void myPostmethod()
{
     Debug.WriteLine("My Post Method is called");
}
```

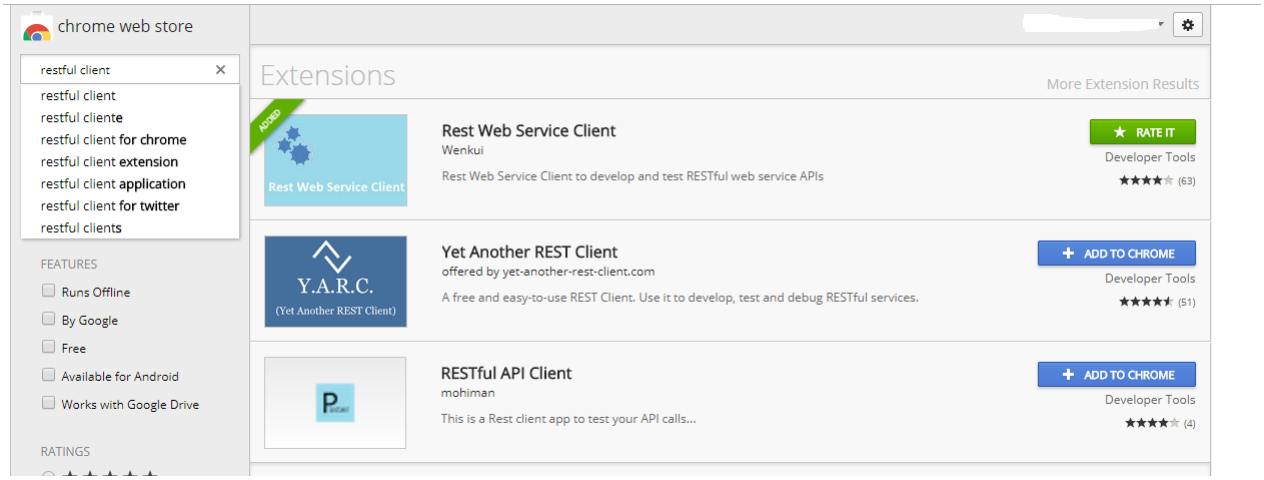
Check the output window after run the code

Use a restful client to check web service instead of javascript call.

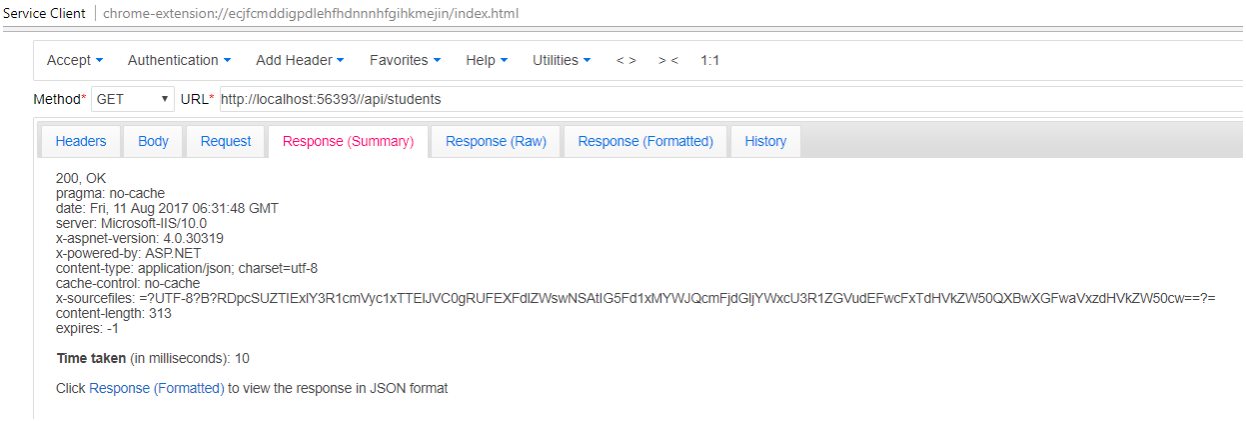
You can verify that your code is working as expected by using these applications.

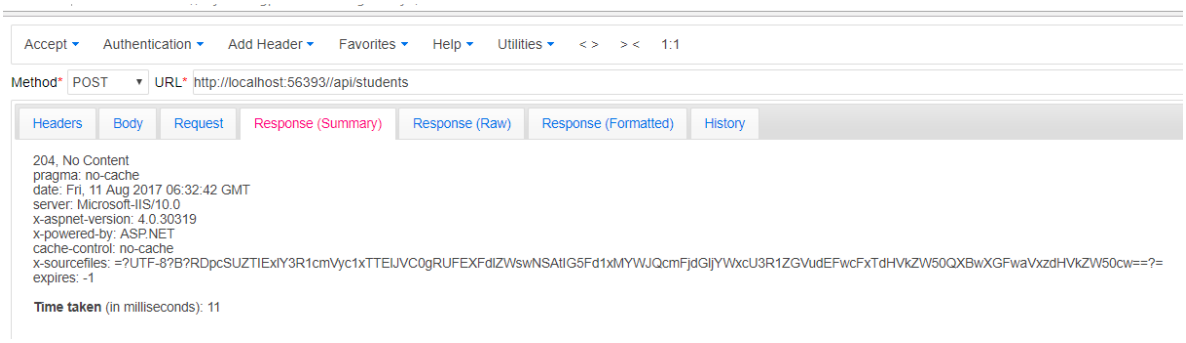
Go to google chrome
Go to this web address : https://chrome.google.com/webstore/search/restful%20client

Download and install one of the client



Check the URL like this.





Or Press F12 Key in the key board. You can analyze the network calls