Al-Azhar UNIVERSITY

Faculty of Engineering

Computers and Systems Engineering Department

EXPERIMENT 3 – Web service Implementation with REST

OBJECTIVES

- Implement a RESTful Web API
- Call the API from Postman App client

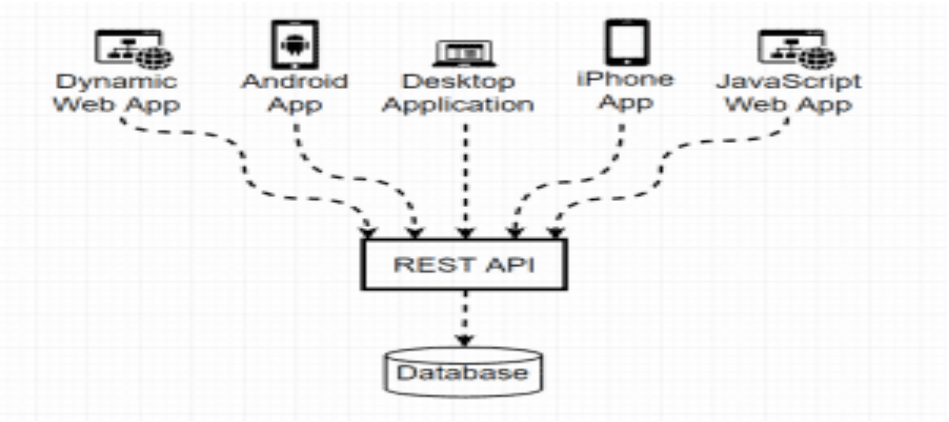
MATERIALS/EQUIPMENT NEEDED

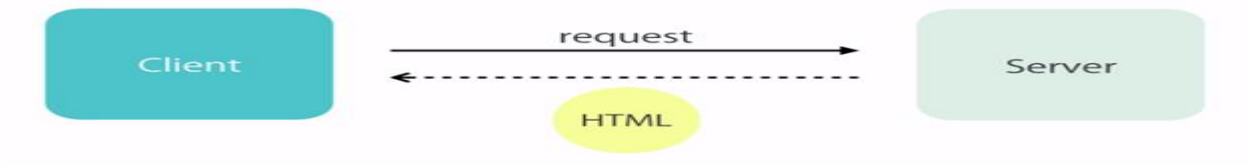
1. Visual Studio Software.
2. Postman Software
3. Web Browser (Optional)

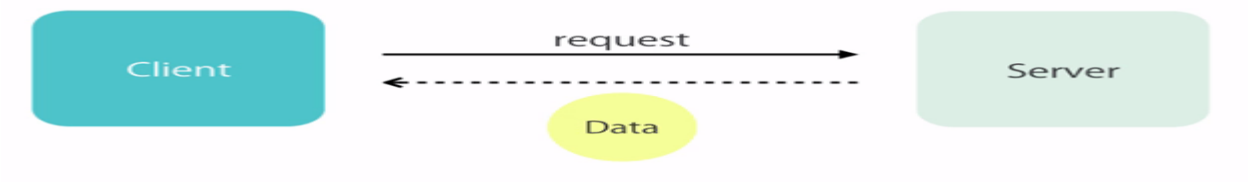
INTRODUCTION

What is Restful Web Service?

REST, or Representational State Transfer, is an architectural style for providing standards between computer systems on the web, making it easier for systems to communicate with each other. REST-compliant systems, often called RESTful systems, are characterized by how they are stateless and separate the concerns of client and server. In the REST architectural style, the implementation of the client and the implementation of the server can be done independently without each knowing about the other. This means that the code on the client side can be changed at any time without affecting the operation of the server, and the code on the server side can be changed without affecting the operation of the client.







HTTP Verbs:

There are 4 basic HTTP verbs we use in requests to interact with resources in a REST system:

GET — retrieve a specific resource (by id) or a collection of resources

POST — create a new resource

PUT — update a specific resource (by id)

DELETE — remove a specific resource by id

Data formats the REST API supports include:

- application/json
- application/xml
- application/x-www-form-urlencoded
- multipart/form-data

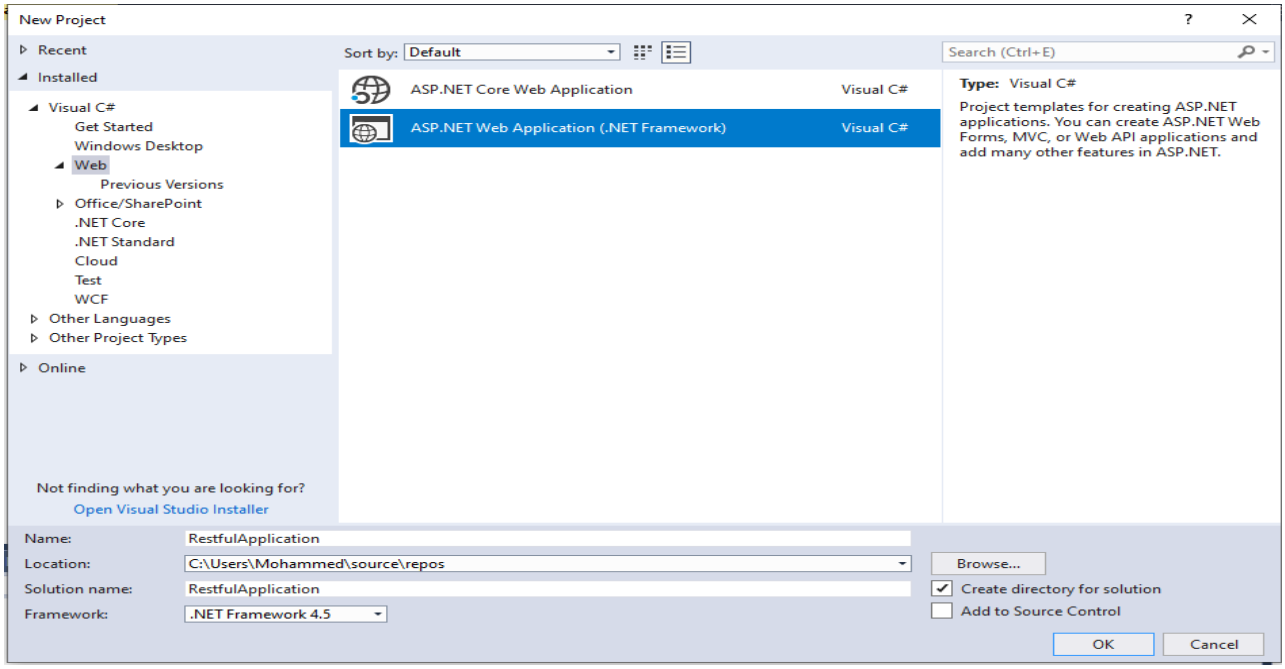
HTTP Status Codes:

Status code	Meaning
200 (OK)	This is the standard response for successful HTTP requests.
201 (CREATED)	This is the standard response for an HTTP request that resulted in an item being successfully created.
204 (NO CONTENT)	This is the standard response for successful HTTP requests, where nothing is being returned in the response body.
400(BAD REQUEST)	The request cannot be processed because of bad request syntax, excessive size, or another client error.
403 (FORBIDDEN)	The client does not have permission to access this resource.
500(INTERNAL SERVER ERROR)	The generic answer for an unexpected failure if there is no more specific information available.

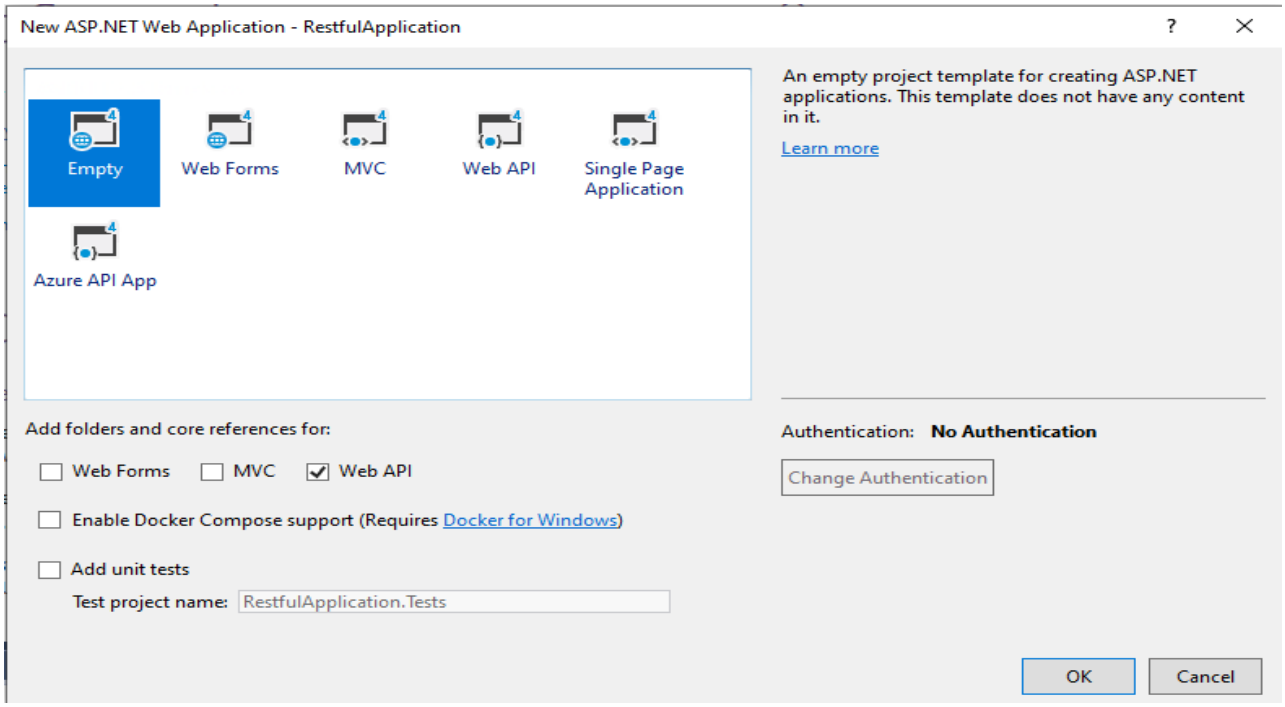
PROCEDURE

Task 1: Create Web API Application

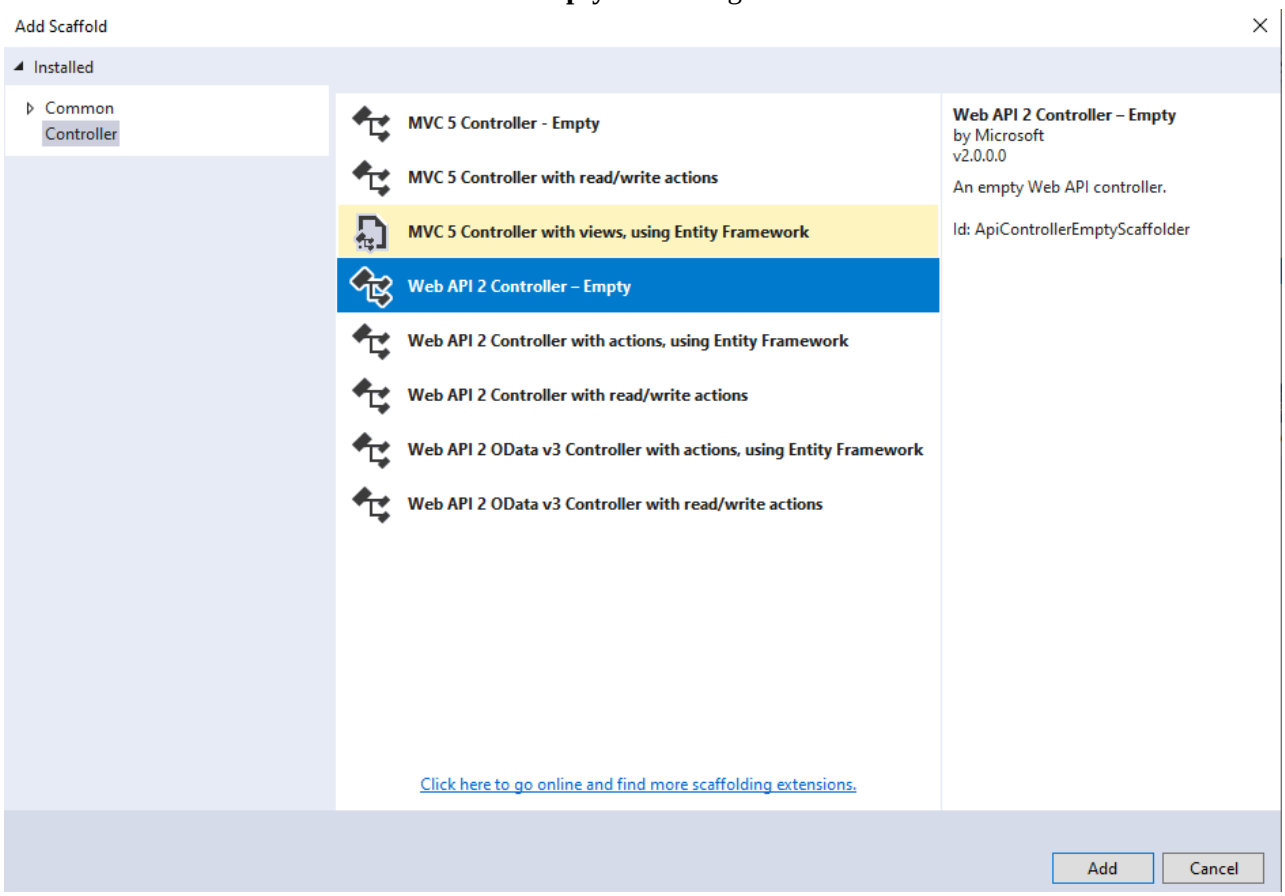
Step 1: From **start menu – All Programs -** run **Microsoft Visual Studio - Microsoft Visual Studio.** From **File** menu select **New Project**. At **New Project** window select **Web** and its **Name** is **RestfulApplication** as shown in the following figure. Then press **OK**

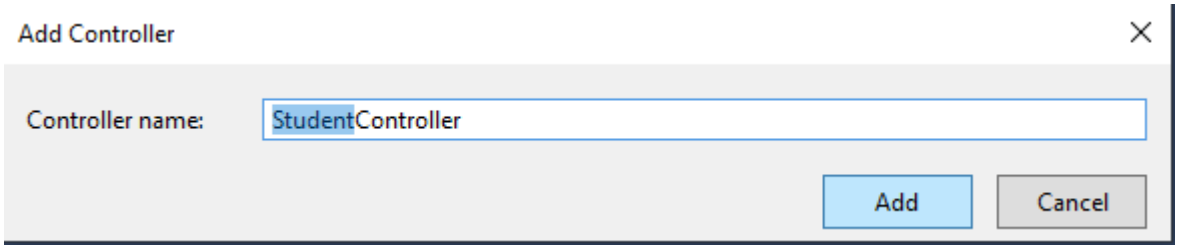


Step 2: In the **ASP.NET Web Application window** project type dialog, select the **Web API project** type. Click **OK**.



Step 3: Creating Student API Controllers: In Solution explorer Right click on Controllers then **add** Controller then select **Web API controller-Empty** then change Controller name to **Student controller**





Step 4: Double click on the **Student controller**. Then change the code to be like this.

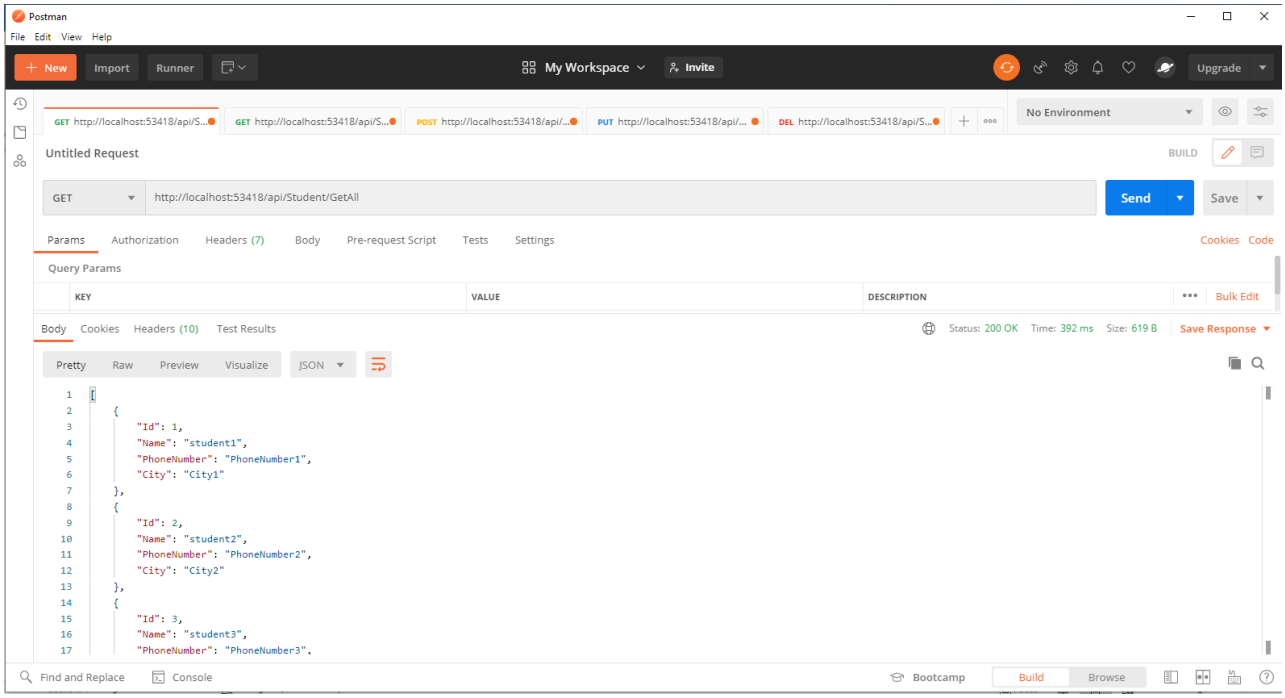
```
using System.Collections.Generic;
using System.Web.Http;

namespace RestfulApplication.Controllers
{
    public class StudentController : ApiController
    {
        public  static List<Student> listStudents = new List<Student>
        {
            new Student{ Id=1,Name="student1",PhoneNumber="PhoneNumber1",City="City1" },
            new Student{ Id=2,Name="student2",PhoneNumber="PhoneNumber2",City="City2" },
            new Student{ Id=3,Name="student3",PhoneNumber="PhoneNumber3",City="City3" },
        };

        [HttpGet]
        public List<Student> GetAll()
        {
            return listStudents;
        }
        [HttpGet]
        public Student GetStudent(int id)
        {
            Student student = null;
            foreach(Student std in listStudents)
            {
                if (std.Id==id)
                {
                    student = std;
                }
            }
            return student;
        }
        [HttpPost]
        public bool AddStudent(Student student)
        {
            listStudents.Add(student);
            return true;
        }
        [HttpPut]
        public bool UpdateStudent(Student student)
        {
            foreach (Student std in listStudents)
            {
                if (std.Id == student.Id)
                {
                    std.Name = student.Name;
                    std.PhoneNumber = student.PhoneNumber;
                    std.City = student.City;
                    return true;
                }
            }
            return false;
        }
```

```
        [HttpDelete]
        public bool DeleteStudent(int id)
        {
            Student student = null;
            foreach (Student std in listStudents)
            {
                if (std.Id == id)
                {
                    student = std;
                    break;
                }
            }
            if(student!=null)
            {
                listStudents.Remove(student);
                return true;
            }
            return false;
        }

    }

    public class Student
    {

        public int Id { set; get; }
        public string Name { set; get; }
        public string PhoneNumber { set; get; }
        public string City { set; get; }

    }
}
```

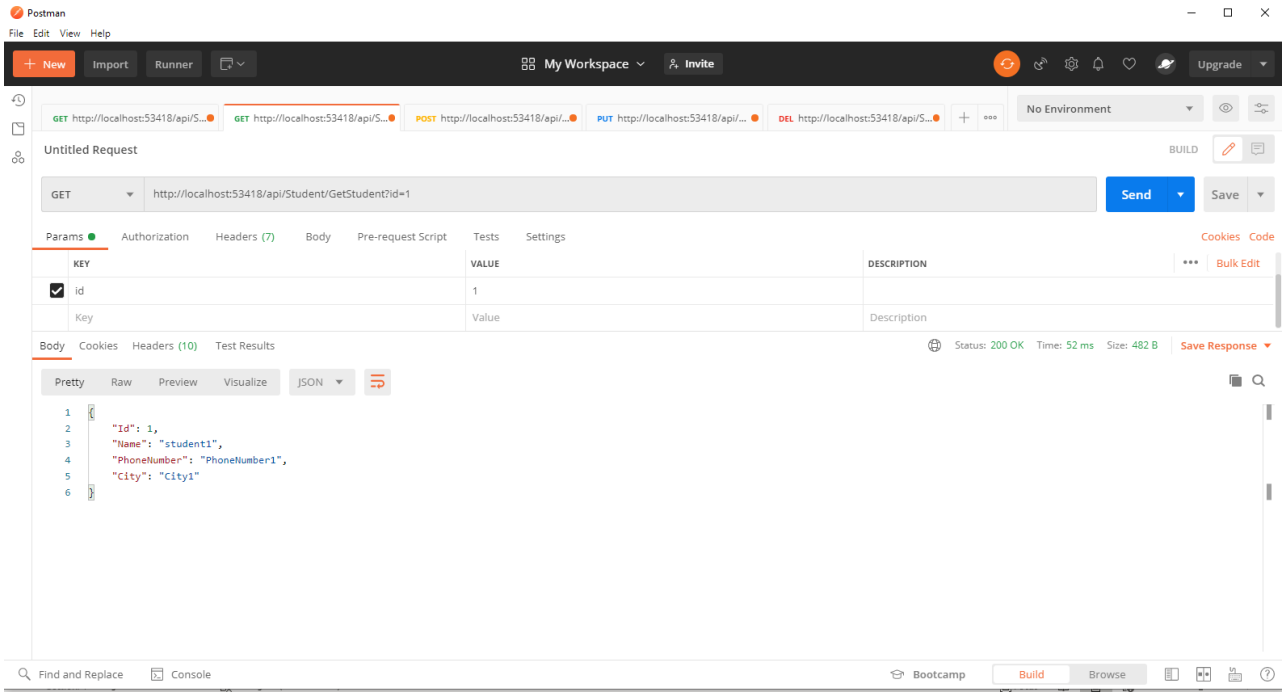
Task 2: Calling Web API s

Step1: Run Web API Application: Right Click on Restful Application from Solution Explorer then Select Debug then Select start new instance

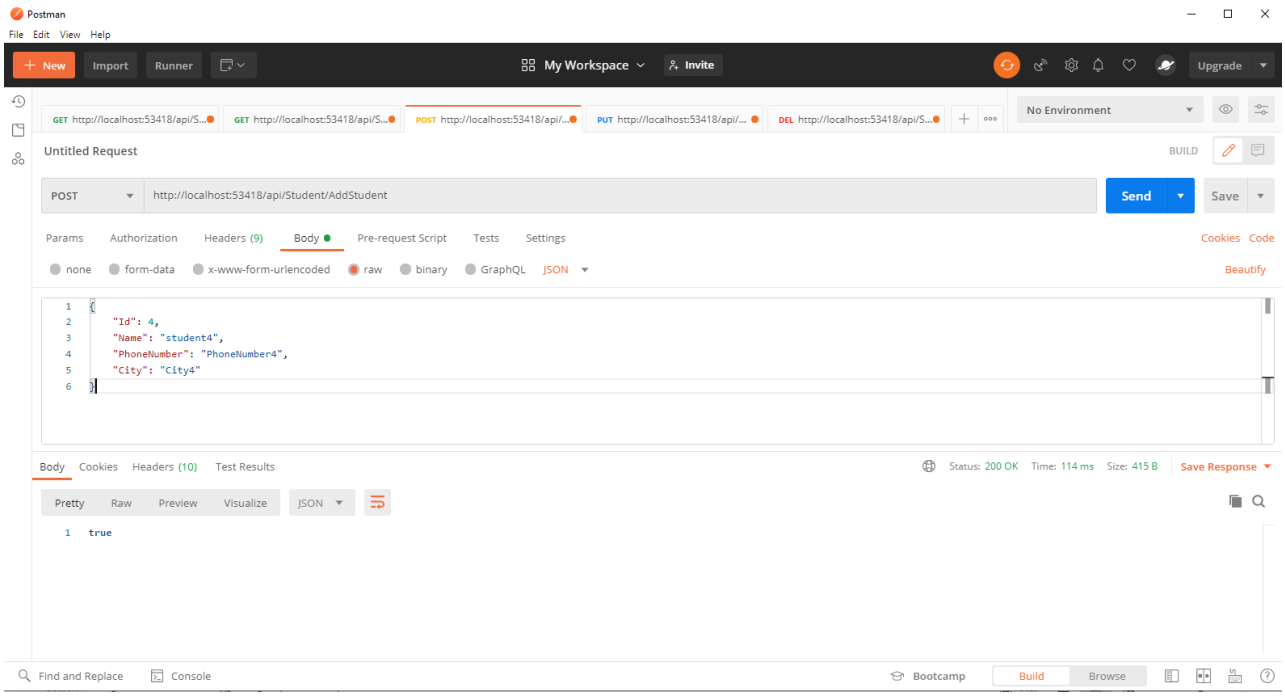
Step2: Call **GetAll** API - Open **Postman** application then enter **GetAll** API URL (http://localhost:53418/api/Student/GetAll) then choose **GET** Action then Press **Send**



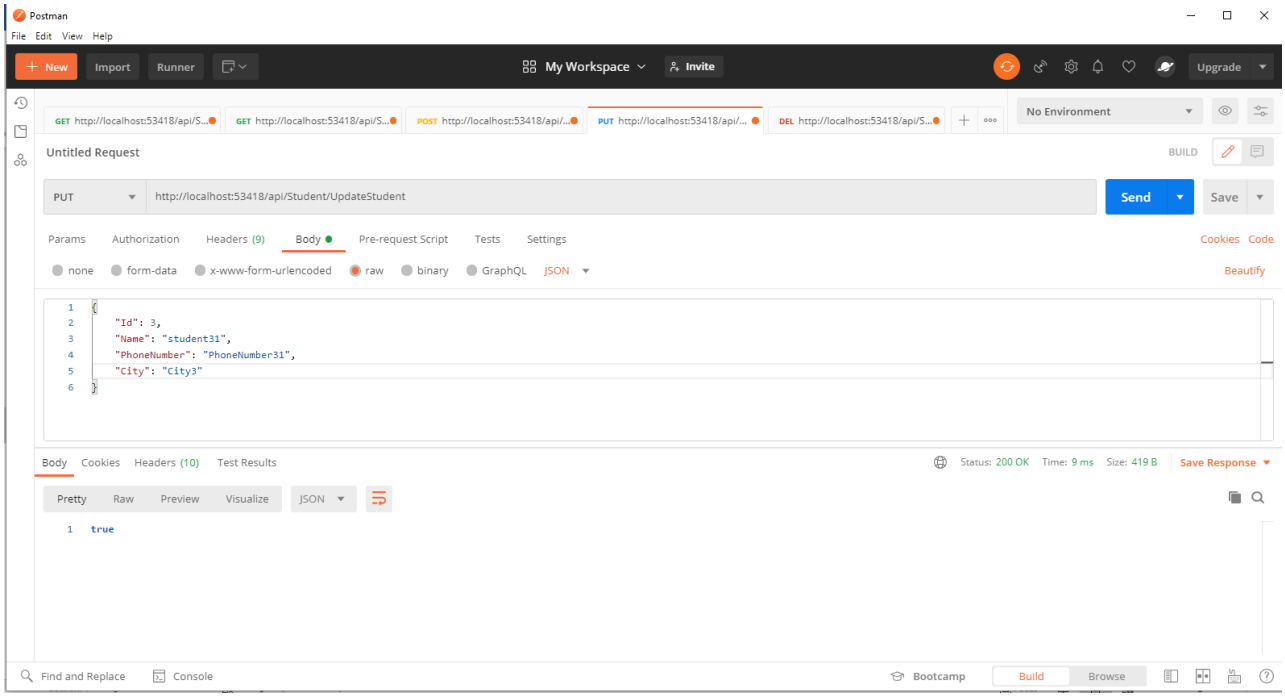
Step3: Call **GetStudent** API - Open **Postman** application then enter **GetStudent** API URL (http://localhost:53418/api/Student/GetStudent?id=1) then choose **GET** Action then Press **Send**



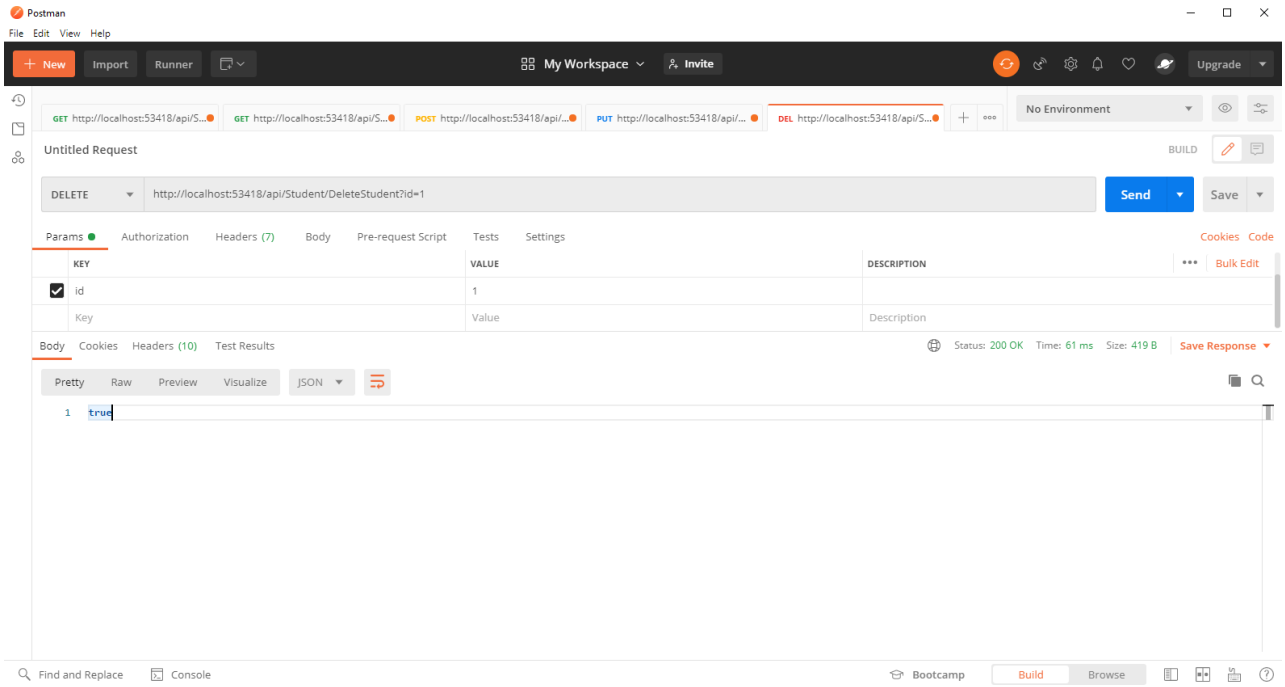
Step4: Call **AddStudent** API - Open Postman application then enter **AddStudent** API URL (http://localhost:53418/api/Student/AddStudent) then choose **Post** then add student Json object Action then Press Send



Step4: Call **UpdateStudent** API - Open Postman application then enter **UpdateStudent** API URL (http://localhost:53418/api/Student/UpdateStudent) then choose **Post** then add student Json object Action then Press Send



Step3: Call **DeleteStudent** API - Open **Postman** application then enter **DeleteStudent** API URL (http://localhost:53418/api/Student/DeleteStudent?id=1) then choose **DELETE** Action then Press **Send**



POST-LAB:

Post-Lab questions must be answered in each experiment's laboratory report.

Exercise 1:

Create and Test Search For students by name API

Exercise 1:

Create and Test Add multiple student API