

Bài thực hành 01

Asp Net Core 2.0 Web API

1. Mục tiêu

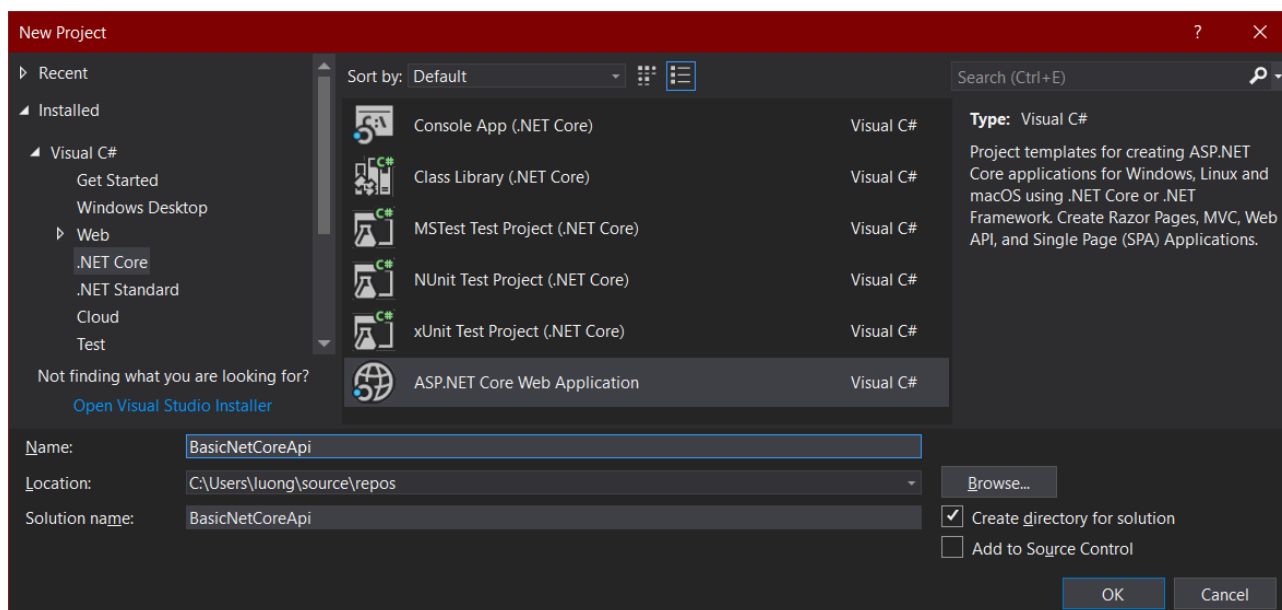
- Tạo project ASP.NETCore Api
- Cài đặt và sử dụng Advanced Client Rest để test Api

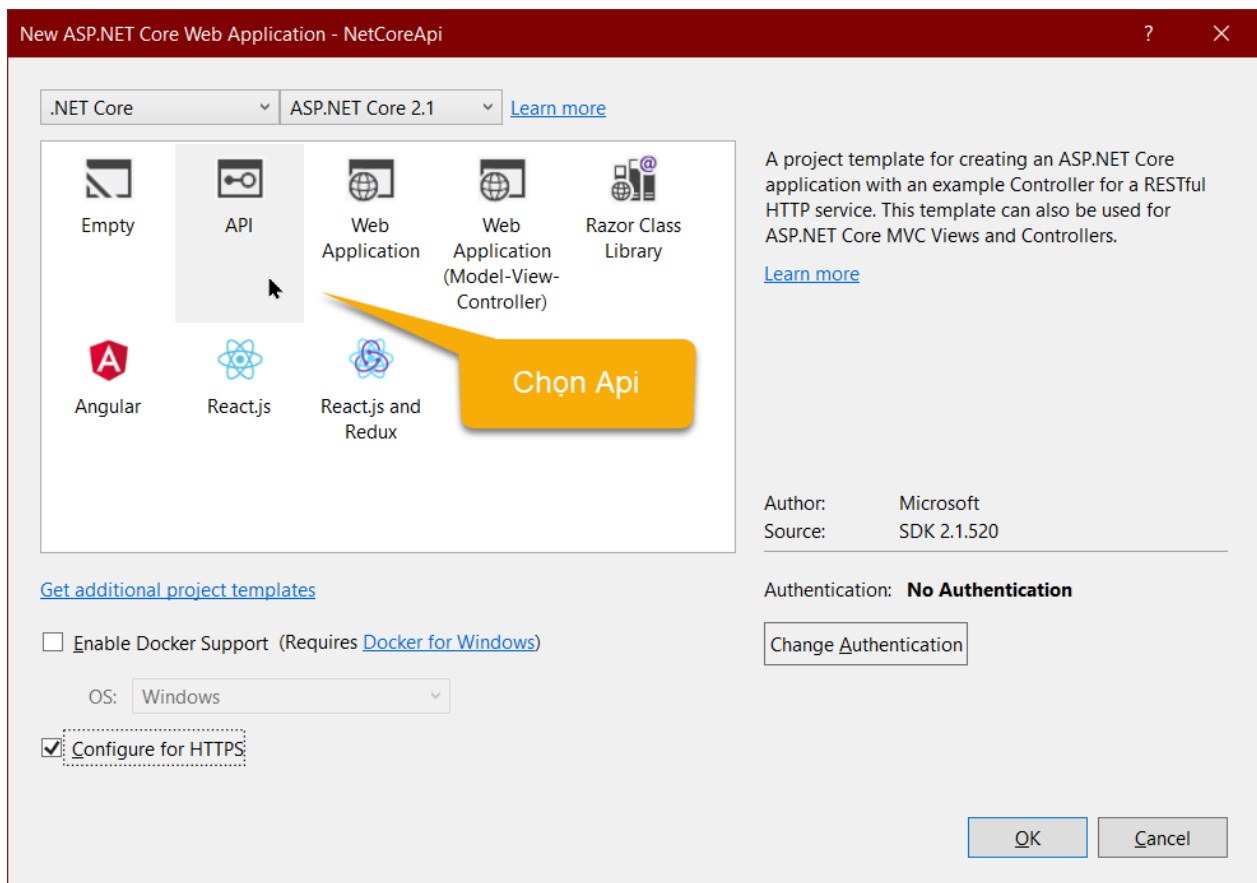
2. Bài thực hành Step by Step (kế thừa từ lab 07)

Bài 1: Tạo project ASP.NET Core Api

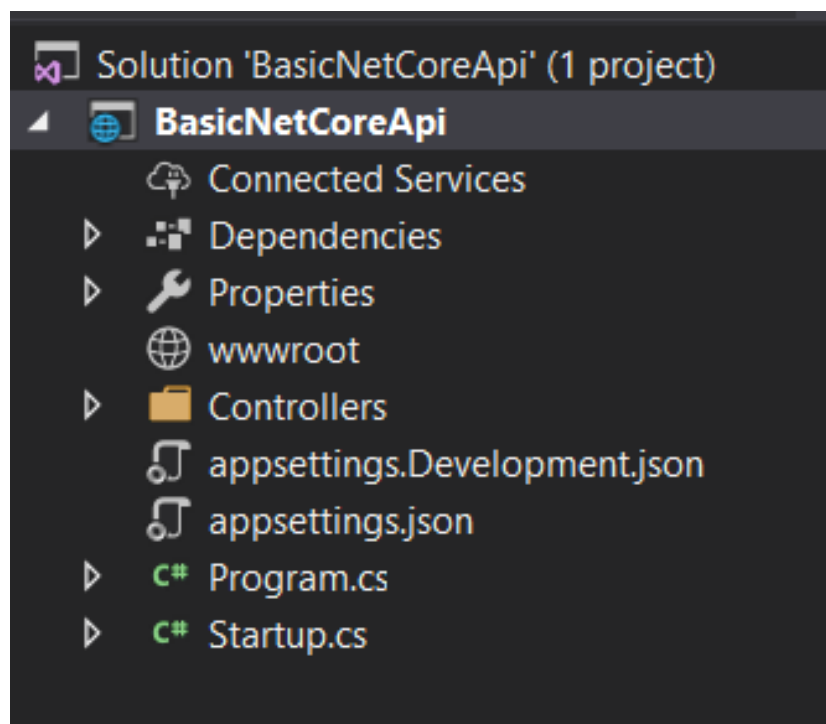
Bước 1: Tạo một project mới với template Api

Mở Visual Studio vào tạo project với tên là **BasicNetCoreApi** chọn template là Api, xem hình





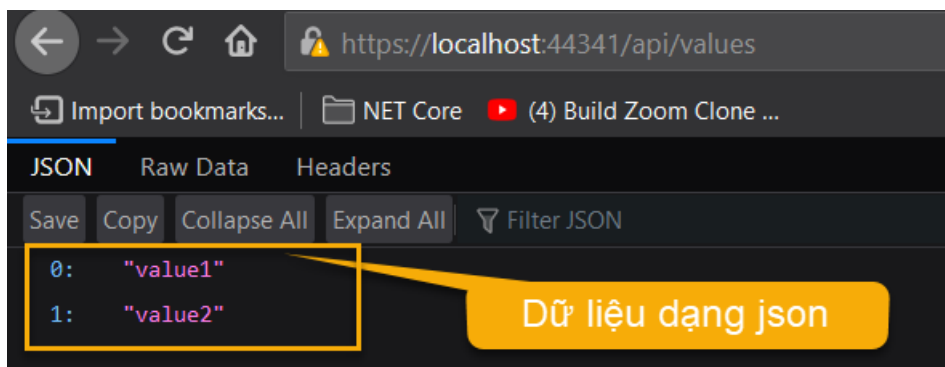
Thư mục dự án có dạng



Bước 2: Build dự án và truy cập thử dữ liệu có dạng

Truy cập link: <https://localhost:44379/api/values>

Kết quả



Dữ liệu được trả về từ ValuesController

Bài 2: Cài đặt Advanced Client Rest để test Api dễ dàng hơn

Lên google tìm với từ khóa Advanced Client Rest để tải và cài đặt

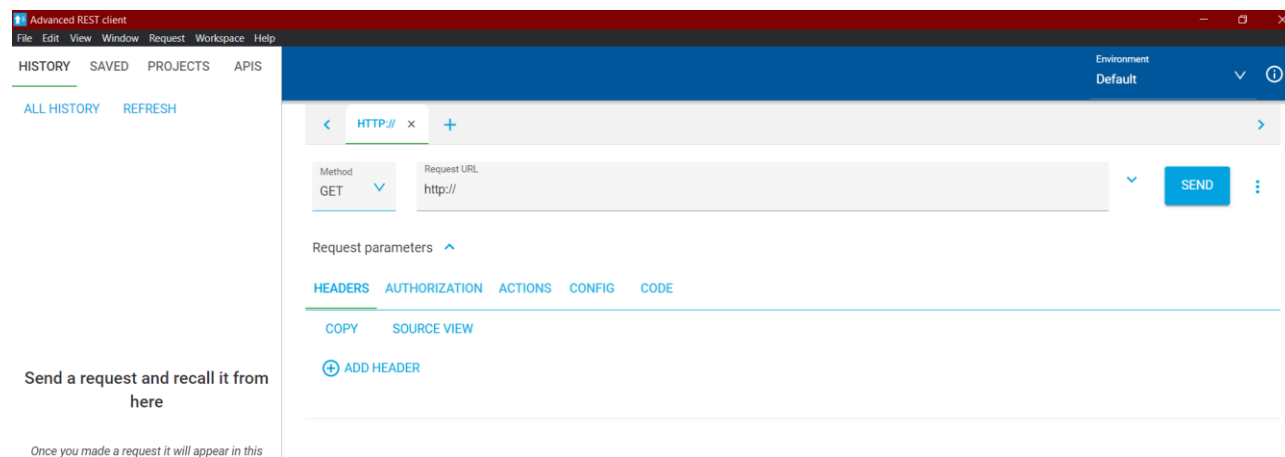
Hoặc truy cập link sau

<https://install.advancedrestclient.com/install>

Tải xong cài đặt bình thường

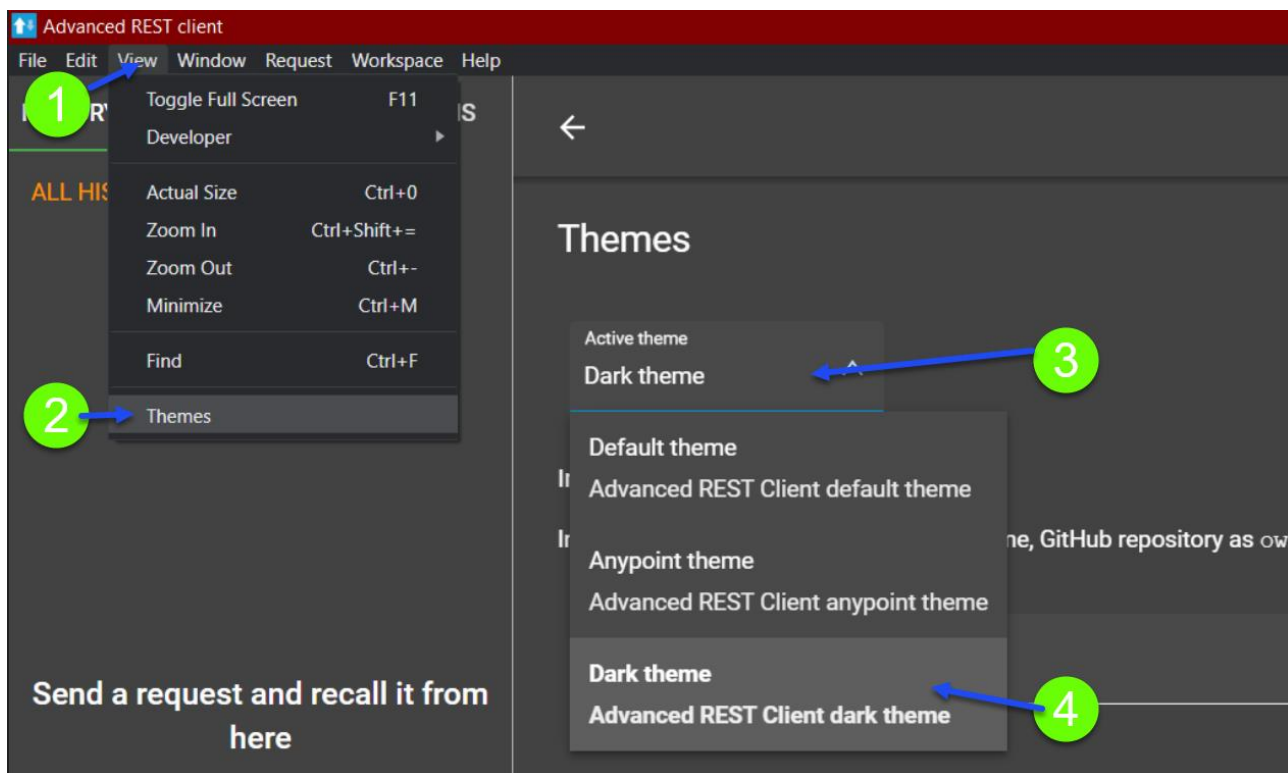
Phần mềm này chỉ sử dụng để test các api trong quá trình xây dựng api

Chú ý: sau khi cài xong giao diện mặc định là



Nếu bạn muốn chuyển sang chế độ màn hình tối (Dark Mode) thì làm các bước sau

Vào menu 1 => View, 2 => Theme, 3 => Select Box, 4 => Dark theme



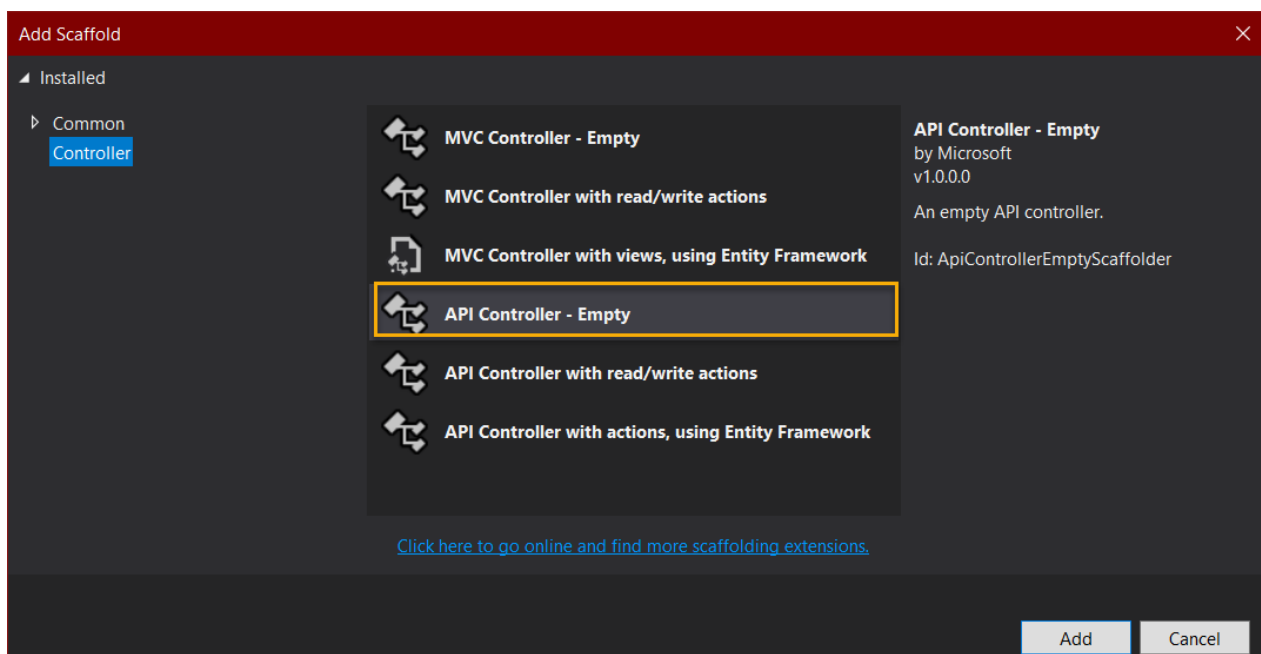
Bài 3: Tạo Api trả về danh sách đối tượng (VD Student)

Bước 1: Tạo thêm thư mục Models và tạo model **Student** gồm các thuộc tính sau

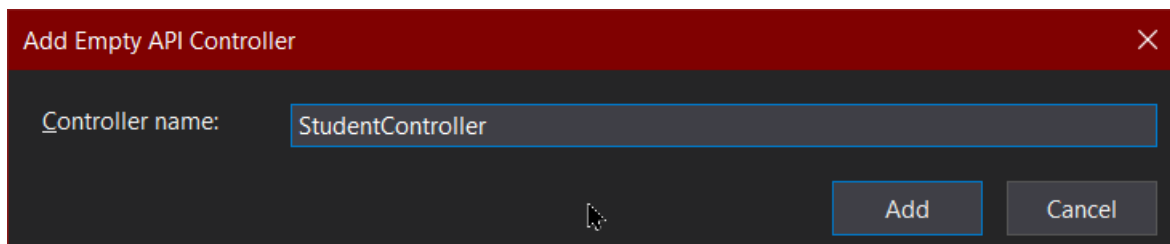
```
namespace BasicNetCoreApi.Models
{
    public class Student
    {
        public string Code { get; set; }
        public string FistName { get; set; }
        public string LastName { get; set; }
        public string Email { get; set; }
        public string Phone { get; set; }
        public string Address { get; set; }
    }
}
```

Bước 2: Tạo controller **StudentController**

Click phải chuột vào thư mục Controllers của dự án, chọn Add -> Controller
Sau đó chọn template là: Api Controller – empty xem hình sau



Đặt tên là **StudentController**



Bước 3: Mở StudentController lên và thêm action Index trả về danh sách các sinh viên với code như sau

```
using BasicNetCoreApi.Models;
using Microsoft.AspNetCore.Mvc;
using System.Collections.Generic;

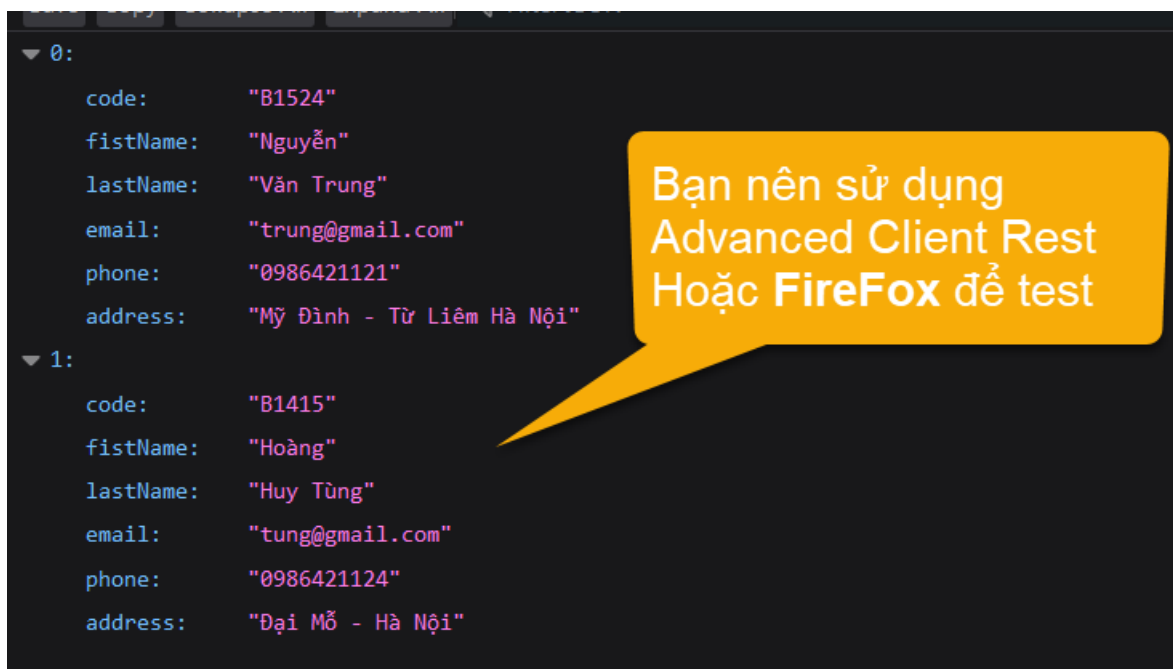
namespace BasicNetCoreApi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class StudentController : ControllerBase
    {
        public IActionResult Index()
        {
            // Khởi tạo list các sinh viên
            List<Student> students = new List<Student>
            {
                // thêm sinh viên vào danh sách
                new Student () {
                    Code = "B1524",
                    FirstName = "Nguyễn",
                    LastName = "Văn Trung",
                    Email = "trung@gmail.com",
                    Phone = "0986421121",
                    Address = "Mỹ Đình - Từ Liêm Hà Nội"
                },
                new Student ()...
                // ... vv các sinh viên tiếp theo
            };

            return Ok(students);
        }
    }
}
```

Bước 4: Build lại dự án và truy cập api có dạng

<https://localhost:44379/api/student>

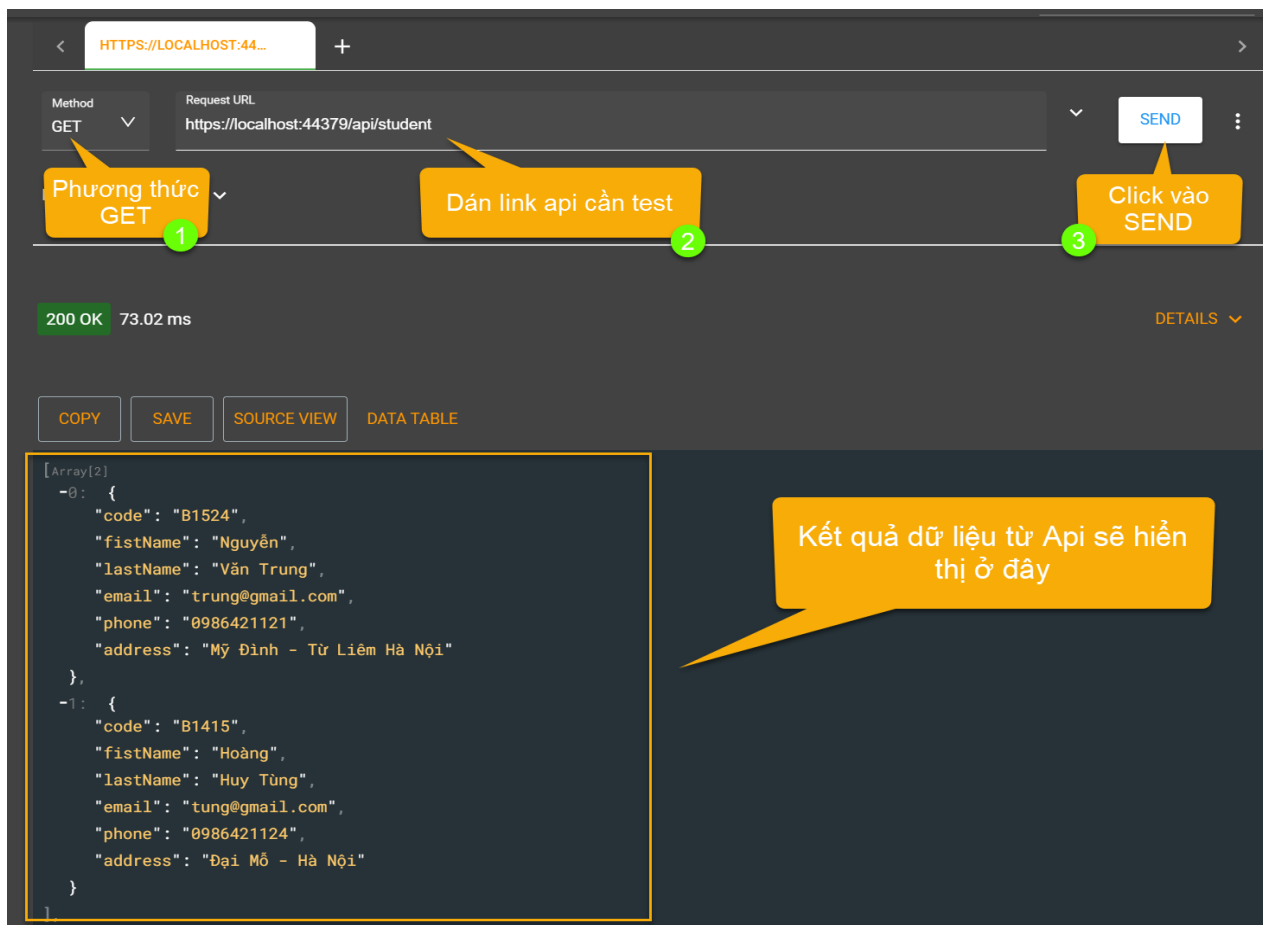
Kết quả sẽ nhìn thấy json data có dạng sau



Bước 5: Test Api trên Advanced Rest Client (ARC)

Copy link Api cần test và dán vào ARC để test như hình sau'

Chọn phương thức: tùy vào api yêu cầu phương thức gì thì chọn phương thức cho phù hợp, thường thì: GET, POST, PUT, DELETE



Bài 3: Api hiển thị chi tiết thông tin một đối tượng

Bước 1: Mở lại **StudentController** lên và thêm Action Detail với code có dạng sau

```
///api/student/detail
[Route("detail")] // đặt tên cho route
public IActionResult Detail()
{
    // Khởi tạo và gán giá trị cho các thuộc tính cho đối tượng Student
    Student student = new Student
    {
        Code = "B1524",
        FirstName = "Nguyễn",
        LastName = "Văn Trung",
        Email = "trung@gmail.com",
        Phone = "0986421121",
        Address = "Mỹ Đình - Từ Liêm Hà Nội"
    };

    return Ok(student);
}
```



Bước 2: Mở lại phần mềm ARC

Truy cập link có dạng: <https://localhost:44379/api/student/:code>

Trong đó code là mã tùy chọn



Bài 4: Tạo Api phương thức POST nhận dữ liệu từ client gửi lên

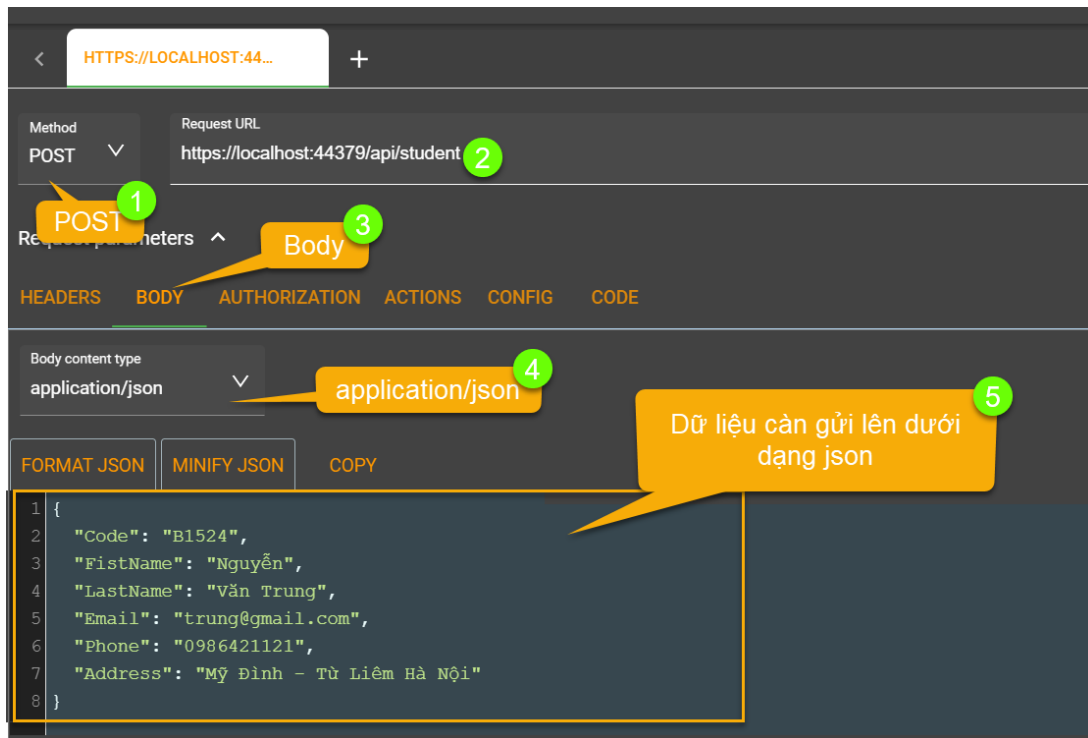
Mở lại **StudentController** thêm phương thức đặt tên Create

```
///api/student  
[HttpPost] // quy định phương thức là POST  
public IActionResult Create(Student student)  
{  
    // trả về dữ liệu y nguyên như dưới client gửi lên  
    return Ok(student);  
}
```

Build lại dự án

Bước 2: Mở lại ARC và test link <https://localhost:44379/api/student>

Chọn phương thức POST



Chú ý dữ liệu dưới client gửi lên server phương thức POST hoặc PUT thường là json data, trong trường hợp model **Student** này nó có dạng

```
{
  "Code": "B1584",
  "FstName": "Đỗ",
  "LastName": "Việt Trung",
  "Email": "trungdv@gmail.com",
  "Phone": "0986421121",
  "Address": "Mỹ Đình - Từ Liêm Hà Nội"
}
```

Tương ứng với các thuộc tính của model đang sử dụng

Bài 4: Tạo api với phương thức PUT, nhận dữ liệu từ client lên

Mở **StudentController** thêm action Update phương thức PUT

```
// PUT => api/student
[HttpPut] // quy định phương thức là PUT
public IActionResult Update(Student student)
{
    // trả về dữ liệu y nguyên như dưới client gửi lên
    return Ok(student);
}
```

Build lại dự án và test như sau



Bài 5: Tạo Api phương thức DELETE

Mở **StudentController** thêm action Delete phương thức Delete

```
// DELETE => api/student/B1524
[HttpDelete] // quy định phương thức là DELETE
[Route("{code}")] // đặt tham số cho route
public IActionResult Delete(string code)
{
    // Khởi tạo và gán giá trị cho các thuộc tính cho đối tượng Student
    Student student = new Student
    {
        Code = code,
        FistName = "Nguyễn",
        LastName = "Văn Trung",
        Email = "trung@gmail.com",
        Phone = "0986421121",
        Address = "Mỹ Đình - Từ Liêm Hà Nội"
    };

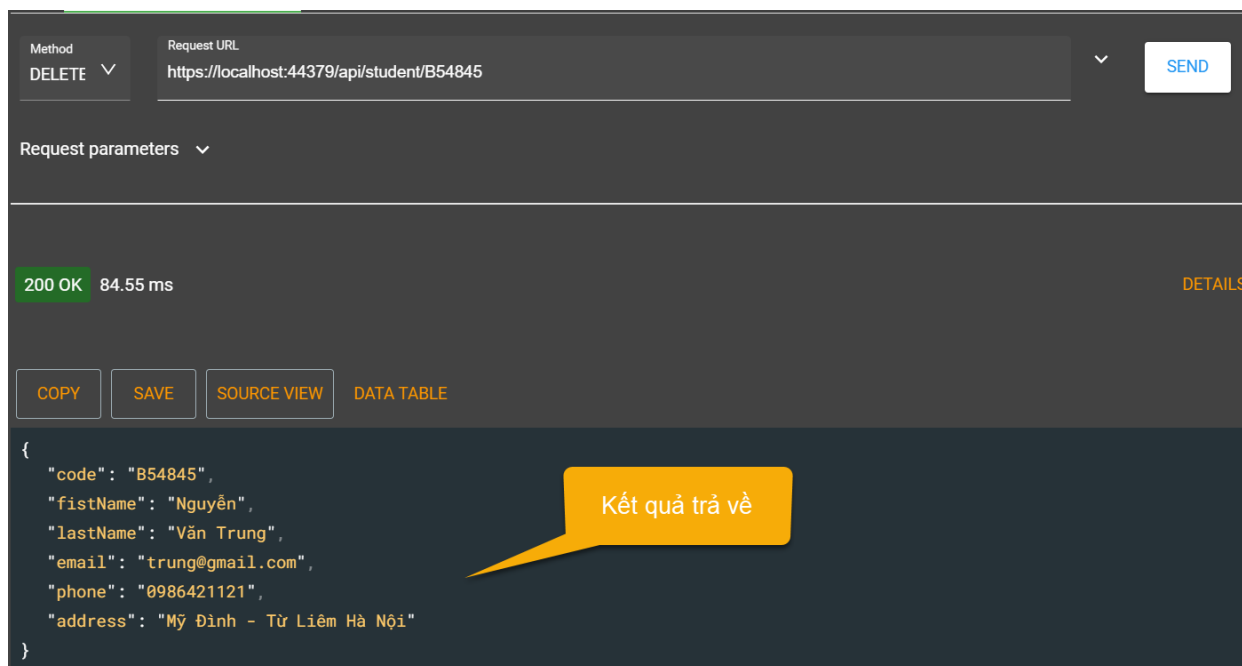
    return Ok(student);
}
```



Build lại dự án và test

<https://localhost:44379/api/student/:code>

Trong đó code là mã tùy chọn



Bài tập tự làm

Sinh viên tạo model Employee có các thuộc tính như sau

```
namespace BasicNetCoreApi.Models
{
    public class Employee
    {
        public int EmployeeId { get; set; }
        public string Name { get; set; }
        public string Email { get; set; }
        public string Phone { get; set; }
        public string Birthday { get; set; }
        public string Gender { get; set; }
        public string Address { get; set; }
    }
}
```

Sau đó tạo các api sau và test trên ARC

Link Api	Phương thức	Mô tả
api/employee	GET	Danh sách Employee
api/employee/:id	GET	Chi tiết 1 Employee
api/employee	POST	Thêm Employee
api/employee	PUT	Sửa Employee
api/employee/:id	DELETE	Xóa Employee

