Get list of resources using ASP.NET Core REST API

In this video we will learn

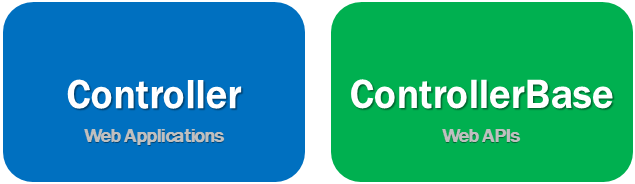
* Creating a REST API using ASP.NET Core
* Return a list of resources i.e a list of employees from the REST API

Create REST API using ASP.NET Core

To create a REST API using ASP.NET core, create a controller class that derives from the built-in ControllerBase class. ControllerBase is in Microsoft.AspNetCore.Mvc namespace.

Controller vs ControllerBase in ASP.NET Core

If you are creating a REST API, make your Controller class derive from ControllerBase and not Controller class. Controller derives from ControllerBase and adds support for MVC views.



So create a controller that derives from Controller class if you are building an MVC web application. On the other hand, if you are creating a Web API, create a controller class that derives from ControllerBase class. So in short, Controller is for MVC web applications and ControllerBase is for MVC Web APIs.

If you are planning to use the controller both for a web application and for a web api, then derive it from the Controller class.

ASP.NET Core API Controller Example

* EmployeesController is an API Controller so decorate it with [ApiController] attribute
* As the name implies, [Route] attribute specifies the route. The word [controller] in square brackets specifies we just use the name of the Controller in the URL to get the EmployeesController.
* IEmployeeRepository is injected into the Controller class using the standrad constructor injection. It is this IEmployeeRepository that retrieves employee data from the underlying SQL Server database. We implemented this repository class in our previous videos in this series.
* [HttpGet] attribute specifies that this method should respond to http GET request.
* The return type of Task<ActionResult> allows us to return status code 200 OK along with the list of employees or status code 500 if there is a server error retrieving data from the database.

using EmployeeManagement.Api.Models;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using System;

using System.Threading.Tasks;

namespace EmployeeManagement.Api.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class EmployeesController : ControllerBase

{

private readonly IEmployeeRepository employeeRepository;

public EmployeesController(IEmployeeRepository employeeRepository)

{

this.employeeRepository = employeeRepository;

}

[HttpGet]

public async Task<ActionResult> GetEmployees()

{

try

{

return Ok(await employeeRepository.GetEmployees());

}

catch (Exception)

{

return StatusCode(StatusCodes.Status500InternalServerError,

"Error retrieving data from the database");

}

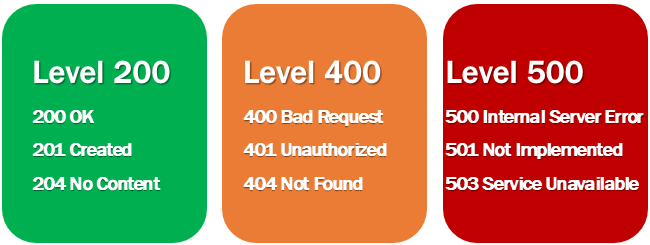
}

}

}

Copy

Http Status codes



Provides the caller i.e the client of the API know the status of the request. Some of the common status codes are 200/OK, 404/Not Found, 204/No Content. For the complete list of HTTP status codes and what they mean, please visit <https://en.wikipedia.org/wiki/List_of_HTTP_status_codes>.

ASP.NET core provides the following helper methods to return HTTP Status Codes.

