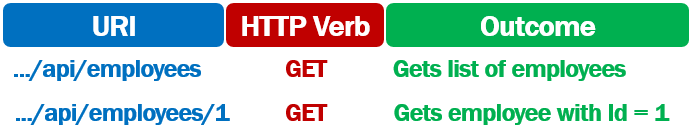
ASP.NET Core REST API get by id

In this video we will learn how to retrieve a resource by id. For example, retrieve Employee by Id. In our previous video we discussed how to retrieve the list of all employees. In this video we want to retrieve a specific Employee i.e Employee by ID.

REST API URLs



* A GET request to the URI /api/employees returns the list of Employees
* A GET request to the URI /api/employees/1 should return the Employee whose ID is 1

ASP.NET Core REST API Get by Id Example

using EmployeeManagement.Api.Models;

using EmployeeManagement.Models;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

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<IEnumerable<Employee>>> GetEmployees()

{

try

{

return (await employeeRepository.GetEmployees()).ToList();

}

catch (Exception)

{

return StatusCode(StatusCodes.Status500InternalServerError,

"Error retrieving data from the database");

}

}

[HttpGet("{id:int}")]

public async Task<ActionResult<Employee>> GetEmployee(int id)

{

try

{

var result = await employeeRepository.GetEmployee(id);

if (result == null) return NotFound();

return result;

}

catch (Exception)

{

return StatusCode(StatusCodes.Status500InternalServerError,

"Error retrieving data from the database");

}

}

}

}

Copy

A method decorated with HttpGet attribute specifies, that amethod should handle HTTP GET request. Both GetEmployees() and GetEmployee() are decorated with the HttpGet attribute. The difference is, HttpGet attribute on GetEmployee() method also has a route template. The id of the employee will be appended to the URL /api/employees.

[HttpGet("{id:int}")]

public async Task<ActionResult<Employee>> GetEmployee(int id)

Copy

* So if the URL is /api/employees then it is handled by GetEmployees() method. If the URL is /api/employees/1 then it is handled by GetEmployee(int id) method.
* The employee id value in the URL is automatically mapped to the id parameter on the GetEmployee(int id) method.
* As we are using the int route constraint on the id route parameter, the id value in the URL is mapped to the method parameter, only if the value is an integer.
* The employee id is then passed to employeeRepository.GetEmployee(id) method which retrieves the respective employee from the database.
* If the employee is found, ASP.NET Core automatically serializes the employee object to JSON and writes it into the response body. This response body along with the http status code 200 OK is returned to the client. If the employee is not found, then http status code 404 NotFound is returned.