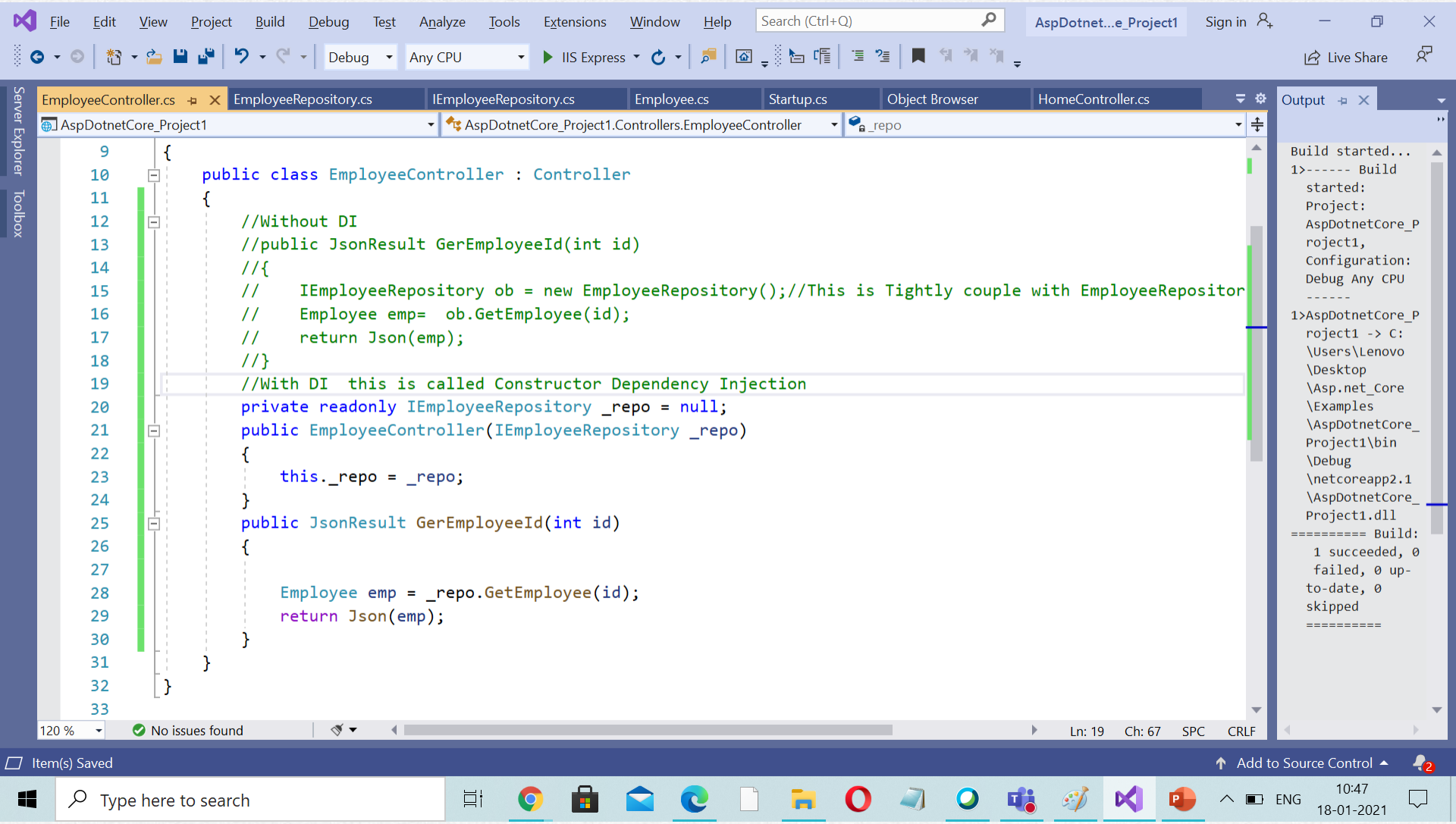
Dependency Injection in ASP.NET Core | AddSingleton, AddScoped and Add Transient

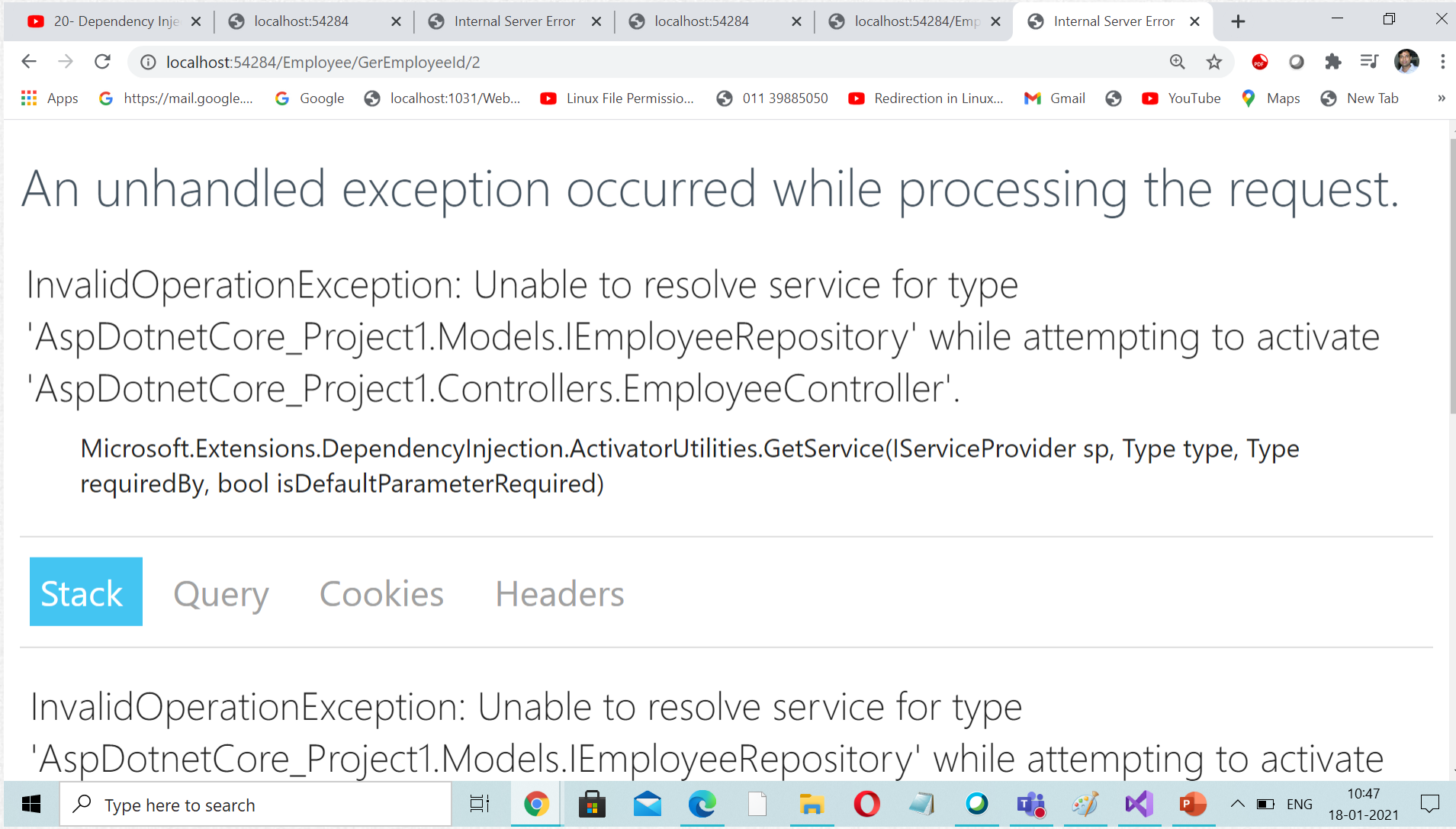
.

What is Dependency Injection?

- The Dependency Injection a process of injecting the object of a class into a class that depends on it.
- The Dependency Injection is the most commonly used design pattern nowadays to remove the dependencies between the objects. So, the Dependency Injection design pattern allows us to develop loosely coupled software components.

Implement DI:-




20- Dependency Inje
localhost:54284
Internal Server Error
localhost:54284
localhost:54284/Emp
Internal Server Error
localhost:54284/Employee/GerEmployeeId/2
Apps
https://mail.google....
Google
localhost:1031/Web...
Linux File Permissio...
011 39885050
Redirection in Linux...
Gmail
YouTube
Maps
New Tab
An unhandled exception occurred while processing the request.
InvalidOperationException: Unable to resolve service for type 'AspDotnetCore_Project1.Models.IEmployeeRepository' while attempting to activate 'AspDotnetCore_Project1.Controllers.EmployeeController'.
Microsoft.Extensions.DependencyInjection.ActivatorUtilities.GetService(IServiceProvider sp, Type type, Type requiredBy, bool isDefaultParameterRequired)
Stack
Query
Cookies
Headers
InvalidOperationException: Unable to resolve service for type 'AspDotnetCore_Project1.Models.IEmployeeRepository' while attempting to activate
Type here to search
ENG
10:47
18-01-2021

How to register a Service with ASP.NET Core Dependency Injection Container?

- We need to register a service with **ASP.NET Core Dependency Injection** Container within the **ConfigureServices()** method of the **Startup** class.
- Before we can discuss how to register a service with the Dependency Injection Container, it is important to understand the lifetime of service.
- When a class receives the dependency object through dependency injection, then whether the instance it receives is unique to that instance of the class or not depends on the lifetime of the service. Setting the lifetime of the dependency object determines how many times the dependency object needs to be created.

What are the different methods ASP.NET Core Provides to register a service with Dependency Injection Contains?

- The ASP.NET core provides 3 methods to register a service with the ASP.NET Core Dependency Injection container. The method that we use to register a service will determine the lifetime of that service.
  - AddSingleton
  - AddScoped
  - AddTransient

AddSingleton()

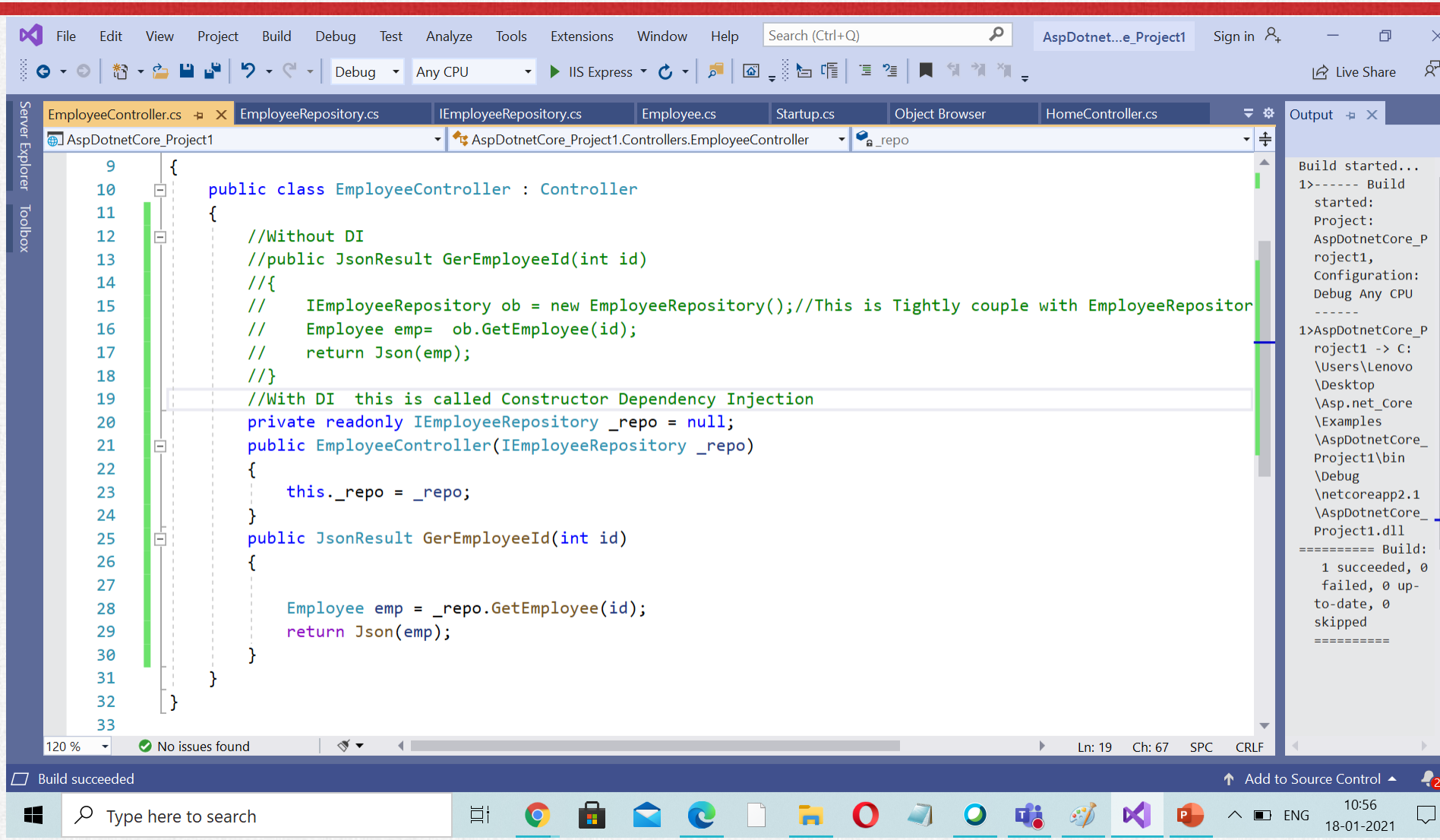
- When we use the AddSingleton() method to register a service, then it will create a singleton service.
- It means a single instance of that service is created and that singleton instance is shared among all the components of the application that require it.
- That singleton service is created when we requested for the first time.

AddScoped()

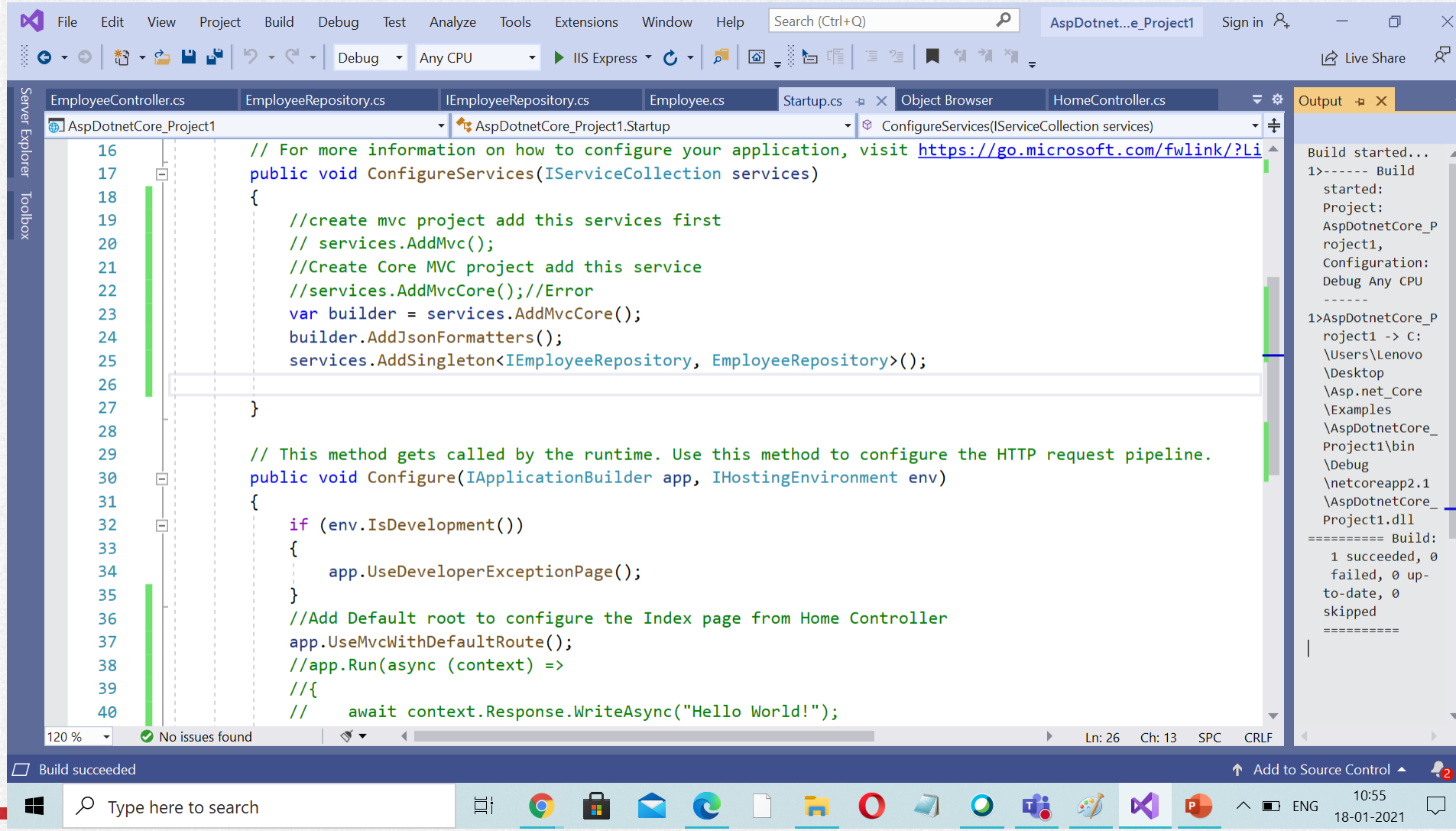
- Scoped means instance per request.
- When we use the AddScoped() method to register a service, then it will create a Scoped service.
- It means, an instance of the service is created once per each HTTP request and uses that instance in other calls of the same request.

AddTransient()

- When we use the AddTransient() method to register a service, then it will create a Transient service.
- It means a new instance of the specified service is created each time when it is requested and they are never shared.


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
AspDotnet...e_Project1
Sign in
Debug
Any CPU
IIS Express
Live Share
EmployeeController.cs
EmployeeRepository.cs
IEmployeeRepository.cs
Employee.cs
Startup.cs
Object Browser
HomeController.cs
AspDotnetCore_Project1
AspDotnetCore_Project1.Controllers.EmployeeController
_repo
9 {
10 public class EmployeeController : Controller
11 {
12 //Without DI
13 //public JsonResult GerEmployeeId(int id)
14 //{
15 // IEmployeeRepository ob = new EmployeeRepository();//This is Tightly couple with EmployeeRepositor
16 // Employee emp= ob.GetEmployee(id);
17 // return Json(emp);
18 //}
19 //With DI this is called Constructor Dependency Injection
20 private readonly IEmployeeRepository _repo = null;
21 public EmployeeController(IEmployeeRepository _repo)
22 {
23 this._repo = _repo;
24 }
25 public JsonResult GerEmployeeId(int id)
26 {
27
28 Employee emp = _repo.GetEmployee(id);
29 return Json(emp);
30 }
31 }
32 }
33
120 %
No issues found
Ln: 19 Ch: 67 SPC CRLF
Output
Build started...
1>------ Build started: Project: AspDotnetCore_Project1, Configuration: Debug Any CPU ------
1>AspDotnetCore_Project1 -> C:\Users\Lenovo\Desktop\Asp.net_Core\Examples\AspDotnetCore_Project1\bin\Debug\netcoreapp2.1\AspDotnetCore_Project1.dll
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
Build succeeded
Add to Source Control
Type here to search
ENG
10:56
18-01-2021

- Add SingleTon service inside the Startup.cs



Thanks