public interface IService

{

void Info();

}

public interface ISingleton : IService { }

public interface IScoped : IService { }

public interface ITransient : IService { }

public abstract class Operation : ISingleton, IScoped, ITransient

{

private Guid \_operationId;

private string \_lifeTime;

public Operation(string lifeTime)

{

\_operationId = Guid.NewGuid();

\_lifeTime = lifeTime;

Console.WriteLine($"{\_lifeTime} Service Created.");

}

public void Info()

{

Console.WriteLine($"{\_lifeTime}: {\_operationId}");

}

}

public class SingletonOperation : Operation

{

public SingletonOperation() : base("Singleton") { }

}

public class ScopedOperation : Operation

{

public ScopedOperation() : base("Scoped") { }

}

public class TransientOperation : Operation

{

public TransientOperation() : base("Transient") { }

}

class Program

{

static void Main(string[] args)

{

var serviceProvider = new ServiceCollection()

.AddTransient<ITransient, TransientOperation>()

.AddScoped<IScoped, ScopedOperation>()

.AddSingleton<ISingleton, SingletonOperation>()

.BuildServiceProvider();

Console.WriteLine("========== Request 1 ============");

serviceProvider.GetService<ITransient>().Info();

serviceProvider.GetService<IScoped>().Info();

serviceProvider.GetService<ISingleton>().Info();

Console.WriteLine("========== ========= ============");

Console.WriteLine("========== Request 2 ============");

serviceProvider.GetService<ITransient>().Info();

serviceProvider.GetService<IScoped>().Info();

serviceProvider.GetService<ISingleton>().Info();

Console.WriteLine("========== ========= ============");

using (var scope = serviceProvider.CreateScope())

{

Console.WriteLine("========== Request 3 ============");

scope.ServiceProvider.GetService<IScoped>().Info();

scope.ServiceProvider.GetService<ITransient>().Info();

scope.ServiceProvider.GetService<ISingleton>().Info();

Console.WriteLine("========== ========= ============");

Console.WriteLine("========== Request 4 ============");

scope.ServiceProvider.GetService<IScoped>().Info();

scope.ServiceProvider.GetService<ISingleton>().Info();

Console.WriteLine("========== ========= ============");

}

using (var scope = serviceProvider.CreateScope())

{

Console.WriteLine("========== Request 5 ============");

scope.ServiceProvider.GetService<IScoped>().Info();

scope.ServiceProvider.GetService<ISingleton>().Info();

Console.WriteLine("========== ========= ============");

}

Console.WriteLine("========== Request 6 ============");

serviceProvider.GetService<IScoped>().Info();

Console.WriteLine("========== ========= ============");

Console.ReadKey();

}

}

Sample Output

========== Request 1 ============

Transient Service Created.

Transient: 6958dca3-05a7-4322-82b6-dec6fcacefeb

Scoped Service Created.

Scoped: c1098528-5855-46f7-a284-ba2a1dff0769

Singleton Service Created.

Singleton: 5ca7706b-5181-49ff-be09-5493f0decc0d

========== ========= ============

========== Request 2 ============

Transient Service Created.

Transient: c7710765-0b09-43bd-86f4-0ef022ed5220

Scoped: c1098528-5855-46f7-a284-ba2a1dff0769

Singleton: 5ca7706b-5181-49ff-be09-5493f0decc0d

========== ========= ============

========== Request 3 ============

Scoped Service Created.

Scoped: 6d5d7842-468f-4040-907c-6b2ab907df61

Transient Service Created.

Transient: 1ec15435-dda2-4fe0-8e29-c154c0352c95

Singleton: 5ca7706b-5181-49ff-be09-5493f0decc0d

========== ========= ============

========== Request 4 ============

Scoped: 6d5d7842-468f-4040-907c-6b2ab907df61

Singleton: 5ca7706b-5181-49ff-be09-5493f0decc0d

========== ========= ============

========== Request 5 ============

Scoped Service Created.

Scoped: 2ad1a715-09b0-4aac-9b84-2df376cc2223

Singleton: 5ca7706b-5181-49ff-be09-5493f0decc0d

========== ========= ============

========== Request 6 ============

Scoped: c1098528-5855-46f7-a284-ba2a1dff0769

========== ========= ============

Output Analysis

Transient - (Request 1,2,3)

In the above output, we can see that a new instance transient service is created whenever the service is requested.

Singleton - (Request 1,2,3,4,5)

In the above output, we can see that a singleton service is only created when it is called for the first time. In the next subsequent requests, the same instance is provided.

Scope - (Request 1,2,3,4,5,6)

In the sample program, we have 3 scopes,

For the Main function

For Request 3 & 4

For Request 5