More Create Blog Sign In

Sql server, .net and c# video tutorial

Free C#, .Net and Sql server video tutorial for beginners and intermediate programmers.

Support us .Net Basics C# SQL ASP.NET Aarvi MVC Slides C# Programs Subscribe Download

Lazy vs Eager loading in Singleton

Suggested Videos
Part 2 - Singleton Design Pattern - Text - Slides
Part 3 - Why is singleton class sealed - Text - Slides
Part 4 - Thread Safety in Singleton - Text - Slides

In this tutorial we will discuss the **difference between Lazy Initialization and Eager Initialization**

Lazy Initialization : The lazy initialization of an object improves the performance and avoids unnecessary computation till the point the object is accessed. Further, it reduces the memory footprint during the startup of the program. Reducing the memory print will help faster loading of the application.

Non-Lazy or Eager Loading : Eager loading is nothing but to initialize the required object before it's being accessed. Which means, we instantiate the object and keep it ready and use it when we need it. This type of initialization is used in lower memory footprints. Also, in eager loading, the common language runtime takes care of the variable initialization and its thread safety. Hence, we don't need to write any explicit coding for thread safety.

Singleton with Lazy keyword (.NET 4.0) : Lazy keyword provides support for lazy initialization. In order to make a property as lazy, we need to pass the type of object to the lazy keyword which is being lazily initialized.

By default, Lazy<T> objects are thread-safe. In multi-threaded scenarios, the first thread which tries to access the Value property of the lazy object will take care of thread safety when multiple threads are trying to access the Get Instance at the same time.

Therefore, it does not matter which thread initializes the object or if there are any thread race conditions that are trying to access this property.

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace SingletonDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            Parallel.Invoke(
```

Complete Tutorials
How to become a full stack web developer
Cloud computing complete tutorial
Healthy food for healthy mind and body
JavaScript tutorial
Bootstrap tutorial
Angular tutorial for beginners
Angular 5 Tutorial for beginners

Important Videos
The Gift of Education
Web application for your business
How to become .NET developer
Resources available to help you

Dot Net Video Tutorials
Blazor tutorial
C tutorial
ASP.NET Core Tutorial
ASP.NET Core Razor Pages Tutorial
Angular 6 Tutorial
Angular CRUD Tutorial
Angular CLI Tutorial
Angular 2 Tutorial
Design Patterns
SOLID Principles
ASP.NET Web API

```
                () => PrintStudentDetails(),
                () => PrintEmployeeDetails()
            );
            Console.ReadLine();
        }

        private static void PrintEmployeeDetails()
        {
            Singleton fromEmployee = Singleton.GetInstance;
            fromEmployee.PrintDetails("From Employee");
        }

        private static void PrintStudentDetails()
        {
            Singleton fromStudent = Singleton.GetInstance;
            fromStudent.PrintDetails("From Student");
        }
    }
}
```

Singleton.cs

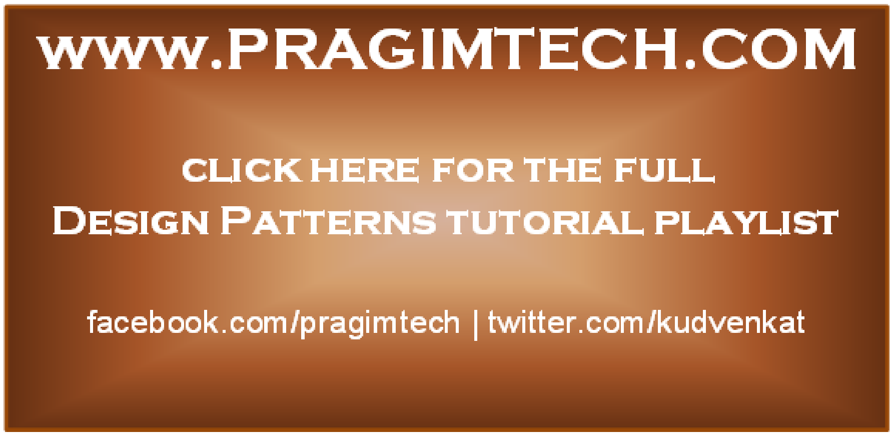
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace SingletonDemo
{

    public sealed class Singleton
    {
        private static int counter = 0;

        private Singleton()
        {
            counter++;
            Console.WriteLine("Counter Value " + counter.ToString());
        }
        private static readonly Lazy<Singleton> instance =
new Lazy<Singleton>(()=>new Singleton());

        public static Singleton GetInstance
        {
            get
            {
                return instance.Value;
            }
        }

        public void PrintDetails(string message)
        {
            Console.WriteLine(message);
        }
    }
}
```


www.PRAGIMTECH.COM
CLICK HERE FOR THE FULL
DESIGN PATTERNS TUTORIAL PLAYLIST
facebook.com/pragimtech | twitter.com/kudvenkat

Bootstrap

AngularJS Tutorial

jQuery Tutorial

JavaScript with ASP.NET Tutorial

JavaScript Tutorial

Charts Tutorial

LINQ

LINQ to SQL

LINQ to XML

Entity Framework

WCF

ASP.NET Web Services

Dot Net Basics

C#

SQL Server

ADO.NET

ASP.NET

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

Dot Net Interview Questions

Slides

Entity Framework

WCF

ASP.NET Web Services

Dot Net Basics

C#

SQL Server

ADO.NET

ASP.NET

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

1 comment:

Unknown April 13, 2018 at 8:21 AM

Excellent Explanation Sir !!! No Words

Reply

Enter Comment

It would be great if you can help share these free resources

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Java Video Tutorials

Part 1 : Video | Text | Slides

Part 2 : Video | Text | Slides

Part 3 : Video | Text | Slides

Interview Questions

C#

SQL Server

Written Test