**Implement Lazy Loading in C# Using Lazy<T> Class**

برای توسعه اپلیکیشن با کارایی و بهره وری بالا یک راه مناسب است.

lazyLoad زمانی بسیار واجب است که ایجاد یک object هزینه بالایی دارد و استفاده از آن شی بسیار نادر است و هدف اصلی ما این است که زمانی داده یا شی را load کنیم که به آن نیاز داریم.

این روش نه فقط جهت افزایش کارایی مناسب است بلکه همچنین به مدیریت حافظه و منابع نیز کمک می کند و زمانی که یک large object را ایجاد می کنیم کاربرد دارد.

راه های مختلفی وجود دارد مثل به تاخیر انداختن ایجاد شی توسط خودمان یا استفاده از lazy<>

نکته:

مقدار دهی اولیه در lazy<> زمانی اتفاق می افتد که برای اولین بار مقدار .Value صدا زده شود.

**IsValueCreated**

**مشخص می کند که آیا object ایجاد شده است یا نه؟**

public class Test

{

private List<string> list = null;

public Test()

{

Console.WriteLine("List Generated:");

list = new List<string>() {

"Sourav","Ram"

};

}

public List<string> Names

{

get

{

return list;

}

}

}

class Program

{

static void Main(string[] args)

{

Lazy<Test> lazy = new Lazy<Test>();

Console.WriteLine("Data Loaded : " + lazy.IsValueCreated);

Test t = lazy.Value;

foreach (string tmp in t.Names)

{

Console.WriteLine(tmp);

}

Console.ReadLine();

}

}

خروجی

False

List generated:

Sourav

Ram

نکته : به صورت پیشفرض اعضای آن threadSafe هستند.

روش دیگر مقدار دهی

Lazy<T> lazyObject = new Lazy<T>(() => {  
 // Code to initialize the value  
 return /\* initialized value \*/;  
});