



Armen Melkumyan • 1st
Technical / Solutions Architect
3mo •

...

💡 From .Net C# Interview Questions: Explaining Lazy Loading in C#:

Lazy Loading is a design pattern in C# for deferring object initialization until needed.

It boosts performance and optimizes resource usage, especially for costly initializations.

🔧 How it's Implemented:

Utilize the `Lazy<T>` type in C#.

Object instances are created only at their first access, not at declaration.

Use Cases:

Large resource-intensive objects.

Enhancing startup time and responsiveness.

Resources used only occasionally .

⚠️ Caution:

Beware of complex debugging and performance impacts in multi-threaded environments.



Example Scenario:

ExpensiveResource is created not at MyClass instantiation but when UseResource is called the first time.

[#CSharp](#) [#LazyLoading](#) [#DotNetDevelopment](#) [#PerformanceOptimization](#)

[#TechnicalInterview](#)

```
1 public class ExpensiveResource
2 {
3     public ExpensiveResource()
4     {
5         Console.WriteLine("ExpensiveResource created");
6     }
7 }
8
9 public class MyClass
10 {
11     private Lazy<ExpensiveResource> _resource = new Lazy<ExpensiveResource>();
12
13     public void UseResource()
14     {
15         var resource = _resource.Value; // ExpensiveResource is created here, on first access
16         // Use the resource
17     }
18 }
19
20 // Usage
21 MyClass myClass = new MyClass();
22 myClass.UseResource(); // "ExpensiveResource created" is printed here
```

Armen Melkumyan



Armen Melkumyan and 139 others

7 comments 4 reposts