IDisposable Best Practices

C# Tips

SWIPE >>>

Richard Crane

IDisposable Interface

IDisposable is a .NET interface used to release unmanaged resources like memory, file handles, network, or database connections, which are not handled by the .NET garbage collector, to prevent resource leaks.

When a class implements **IDisposable**, a **Dispose** method is necessary. This method implements the logic to close or release unmanaged resources.

SWIPE >>>

Richard Crane

IDisposable Example

```
public class SomeResource : IDisposable
{
    // ... some resources like a file handle, database
connection etc.

    public void Dispose()
    {
        // logic to close or release resources
    }
}
```

SWIPE >>>

Richard Crane

using

You typically use the **using** statement in **C#** when working with an **IDisposable** object. The **using** statement automatically calls the **Dispose** method, even if an exception occurs within the `using` block.

```
using (SomeResource resource = new SomeResource())
{
    // Use resource...
} // Dispose gets called automatically here
```

SWIPE >>>

Richard Crane

using var

using var is a feature that was introduced in **C# 8.0**. It's like the **using** statement, but with a slight difference: the scope of the object is not limited to the nearest enclosing block but extends to the end of the containing method. This can lead to more readable code since it avoids deeply nested blocks.

SWIPE >>>

```
public void Example()
{
    using var reader = new StreamReader("file.txt");
    // Use reader

    // Other code...

    // reader.Dispose() is called here, at the end of
the method.
}
```

Richard Crane

`using var` (cont'd)

The **`using var`** construct might delay resource disposal until a method's end, potentially causing resources to stay in memory longer if additional operations are run afterwards.

Delayed disposal with **`using var`** could cause issues if resources are scarce, costly, or if further long-running operations follow. It might be better to use the classic **`using`** statement if you want to ensure that resources are freed as soon as possible.

SWIPE >>>

Richard Crane

Get more tips at WickedProgrammer.com