

Coming Up



Security implications from past to present

**Working towards cleaner code with
ReflectionMagic**

CAS (Code Access Security)

A security mechanism which enables different levels of trust for your code to help protect systems from malicious code

Security Implications from Past to Present

Reflection was influenced by CAS

- **SecurityCritical** attribute
- **ReflectionPermissionFlag** enumeration

.NET 5 no longer supports or honors these

Security Implications from Past to Present

The reflection-only context allowed loading assemblies for inspection but not execution

This is no longer supported in .NET 5

```
var person = new Person("Kevin");  
var privateField = person.GetType().GetField("_aPrivateField",  
                                             BindingFlags.Instance | BindingFlags.NonPublic);  
privateField.SetValue(person, "New private field value");
```

Working Towards Cleaner Code with ReflectionMagic

Reflection-based code can become bloated

```
var person = new Person("Kevin");  
var privateField = person.GetType().GetField("_aPrivateField",  
                                             BindingFlags.Instance | BindingFlags.NonPublic);  
privateField.SetValue(person, "New private field value");
```

Working Towards Cleaner Code with ReflectionMagic

Reflection-based code can become bloated

Working Towards Cleaner Code with ReflectionMagic

ReflectionMagic is a library that helps with writing cleaner code

- **Uses a custom `DynamicObject` underneath the covers**

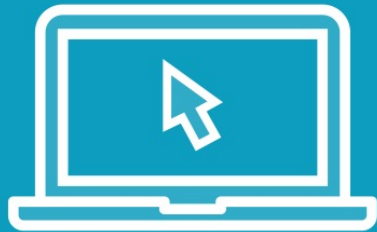
```
var person = new Person("Kevin");  
person.AsDynamic()._aPrivateField = "Updated value via ReflectionMagic";
```

Working Towards Cleaner Code with ReflectionMagic

Better developer experience

Cleaner codebase

Demo



**Working towards cleaner code with
ReflectionMagic**

Summary



Code Access Security and the reflection-only context don't exist in .NET 5

ReflectionMagic helps keep your codebase clean



@KevinDockx