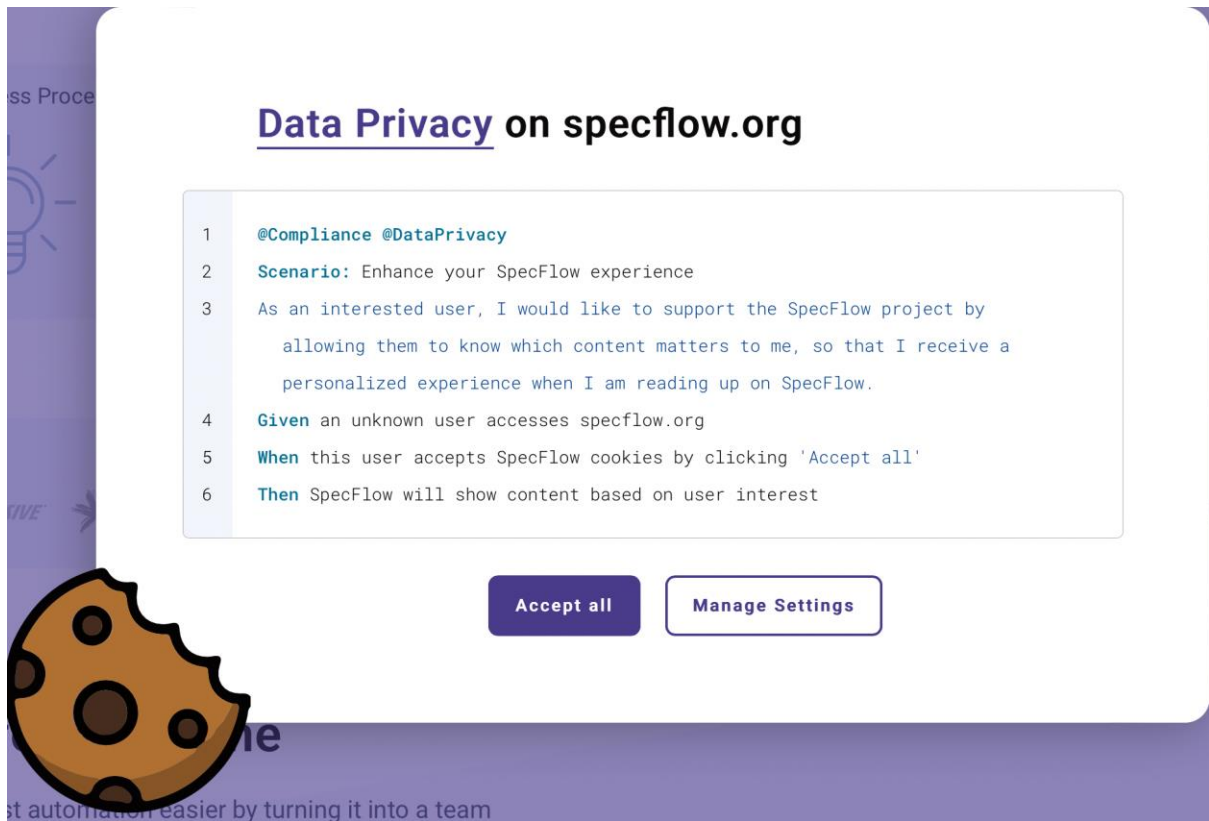


# Chapter 1: How Can Metaprogramming Benefit You?

*No images...*

## Chapter 2: Metaprogramming Concepts



ss Proce

### Data Privacy on specflow.org

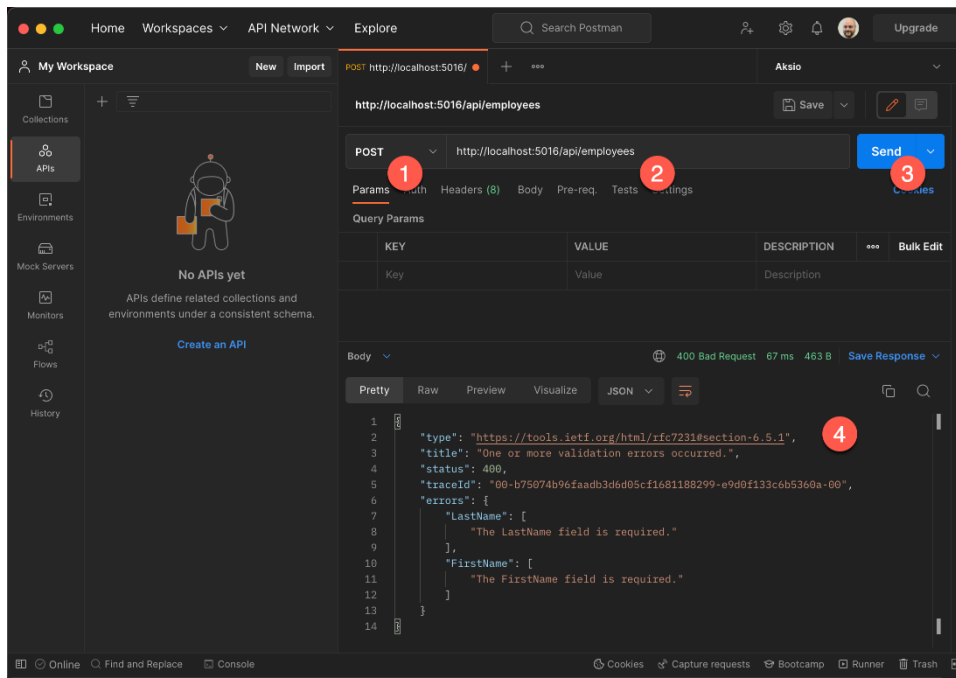
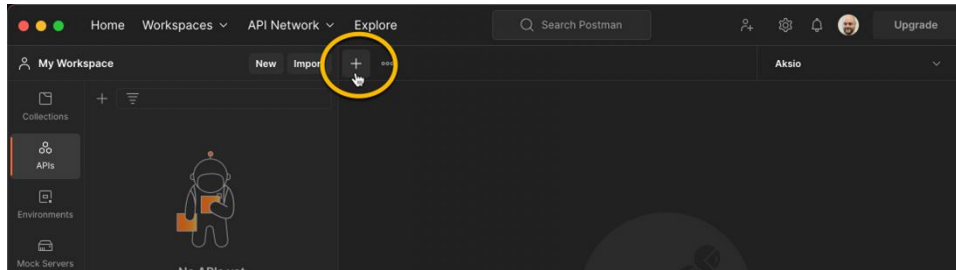
- 1 `@Compliance @DataPrivacy`
- 2 **Scenario:** Enhance your SpecFlow experience
- 3 As an interested user, I would like to support the SpecFlow project by  
allowing them to know which content matters to me, so that I receive a  
personalized experience when I am reading up on SpecFlow.
- 4 **Given** an unknown user accesses specflow.org
- 5 **When** this user accepts SpecFlow cookies by clicking 'Accept all'
- 6 **Then** SpecFlow will show content based on user interest

**Accept all** **Manage Settings**

ne

st automation easier by turning it into a team

## Chapter 3: Demystifying through Existing Real-World Examples



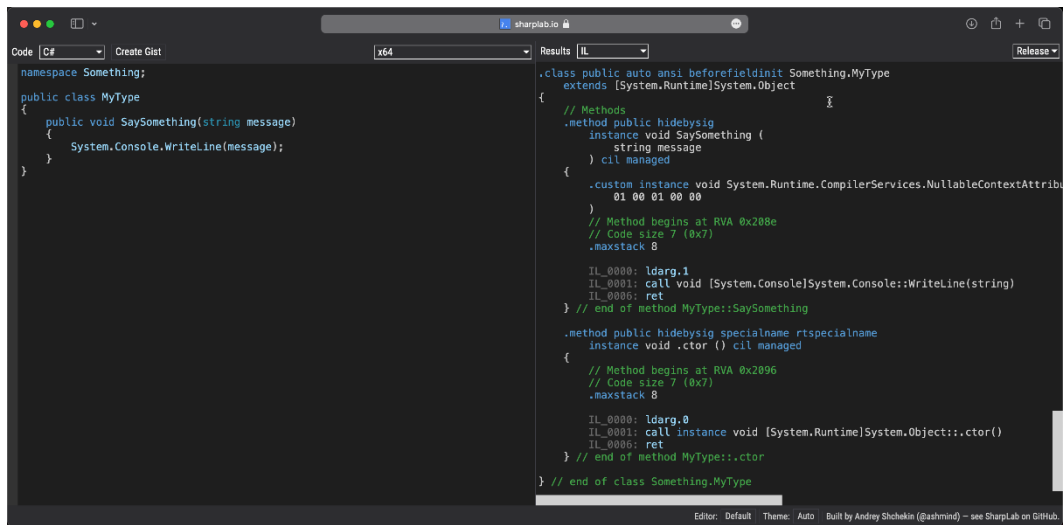
## Chapter 4: Reasoning about Types Using Reflection

*No images...*

## Chapter 5: Leveraging Attributes

*No images...*

## Chapter 6: Dynamic Proxy Generation



The screenshot displays the SharpLab online compiler interface. The left pane shows the C# source code, and the right pane shows the generated IL code. The C# code defines a class `MyType` in the `Something` namespace with a single method `SaySomething` that writes a message to the console. The IL code shows the class structure, including the `SaySomething` method with its metadata, a custom instance void, and the `.ctor` method. The IL for `SaySomething` includes instructions for loading the argument, calling `System.Console.WriteLine`, and returning. The `.ctor` method includes instructions for loading `0`, calling `System.Runtime.System.Object::ctor`, and returning.

```
Code C# Create Gist x64 Results IL Release

namespace Something;
public class MyType
{
    public void SaySomething(string message)
    {
        System.Console.WriteLine(message);
    }
}

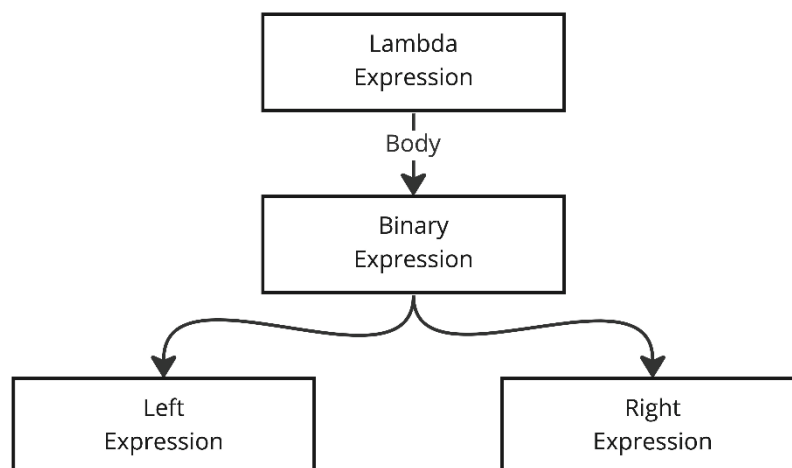
.class public auto ansi beforefieldinit Something.MyType
    extends [System.Runtime]System.Object
{
    // Methods
    .method public hidebysig
        instance void SaySomething (
            string message
        ) cil managed
    {
        .custom instance void System.Runtime.CompilerServices.NullableContextAttribute::+RTTID()=0x00000001 01 00 01 00 00
    }
    // Method begins at RVA 0x208e
    // Code size 7 (0x7)
    .maxstack 8
    IL_0000: ldarg.1
    IL_0001: call void [System.Console]System.Console::WriteLine(string)
    IL_0005: ret
} // end of method MyType::SaySomething

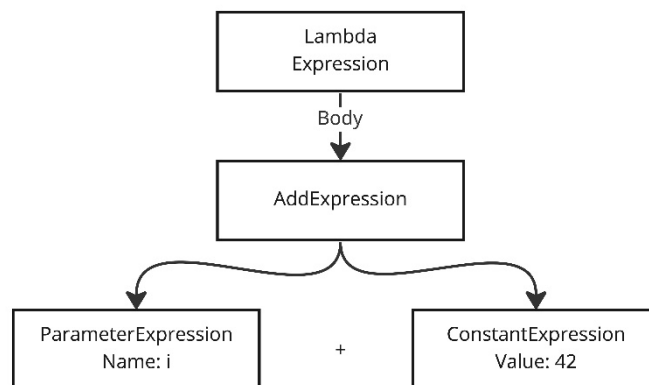
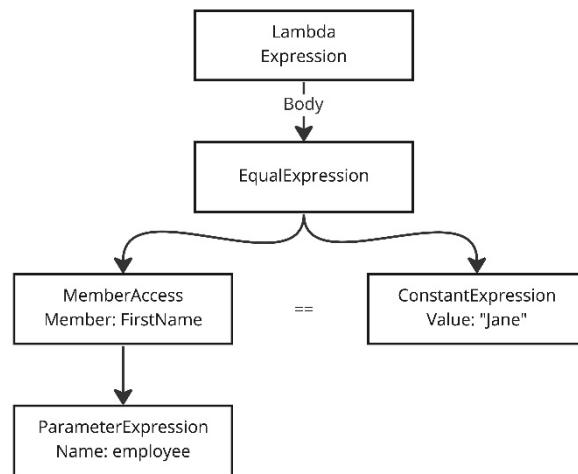
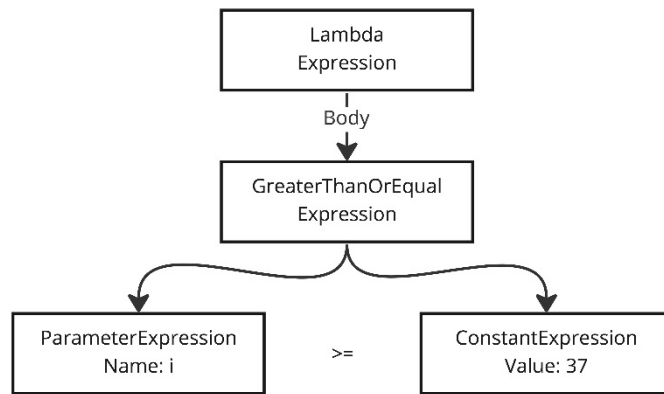
.method public hidebysig specialname rtspecialname
    instance void .ctor () cil managed
{
    // Method begins at RVA 0x2096
    // Code size 7 (0x7)
    .maxstack 8
    IL_0000: ldarg.0
    IL_0001: call instance void [System.Runtime]System.Object::.ctor()
    IL_0005: ret
} // end of method MyType::.ctor
} // end of class Something.MyType

Editor: Default Theme: Auto Built by Andrey Shchekin (@ashmind) - see SharpLab on GitHub.
```

## Chapter 7: Reasoning about Expressions

```
✓ filter [Expression]: {i => (i >= 37)}  
  ✓ Body [Expression]: {(i >= 37)}  
    CanReduce [bool]: false  
    > Conversion [LambdaExpression]: null  
    DebugView [string]: "$i >= 37"  
    IsLifted [bool]: false  
    IsLiftedToNull [bool]: false  
    > Left [Expression]: {i}  
    Method [MethodInfo]: null  
    NodeType [ExpressionType]: GreaterThanOrEqual  
    > Right [Expression]: {37}  
    > Type [Type]: {System.Boolean}  
    > Raw View  
      CanReduce [bool]: false  
      DebugView [string]: ".Lambda #Lambda1<System.Func`2[System.Int32,System.B...  
      Name [string]: null  
      NodeType [ExpressionType]: Lambda
```





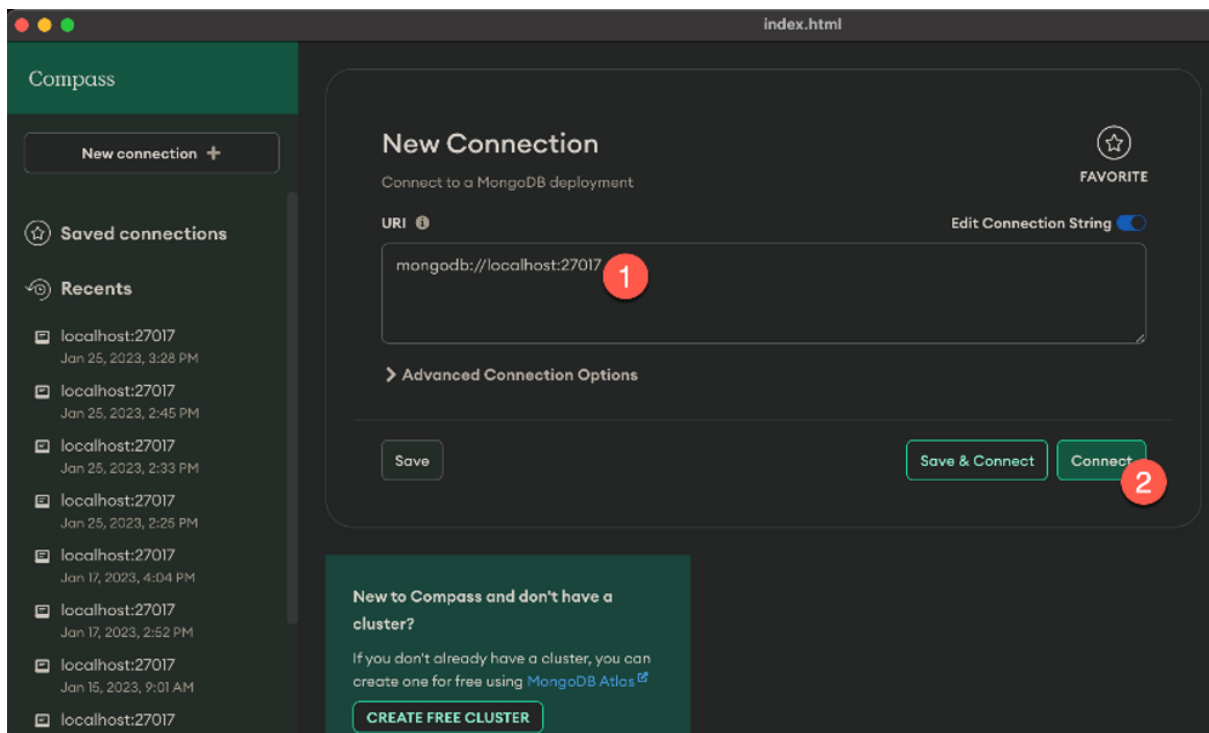
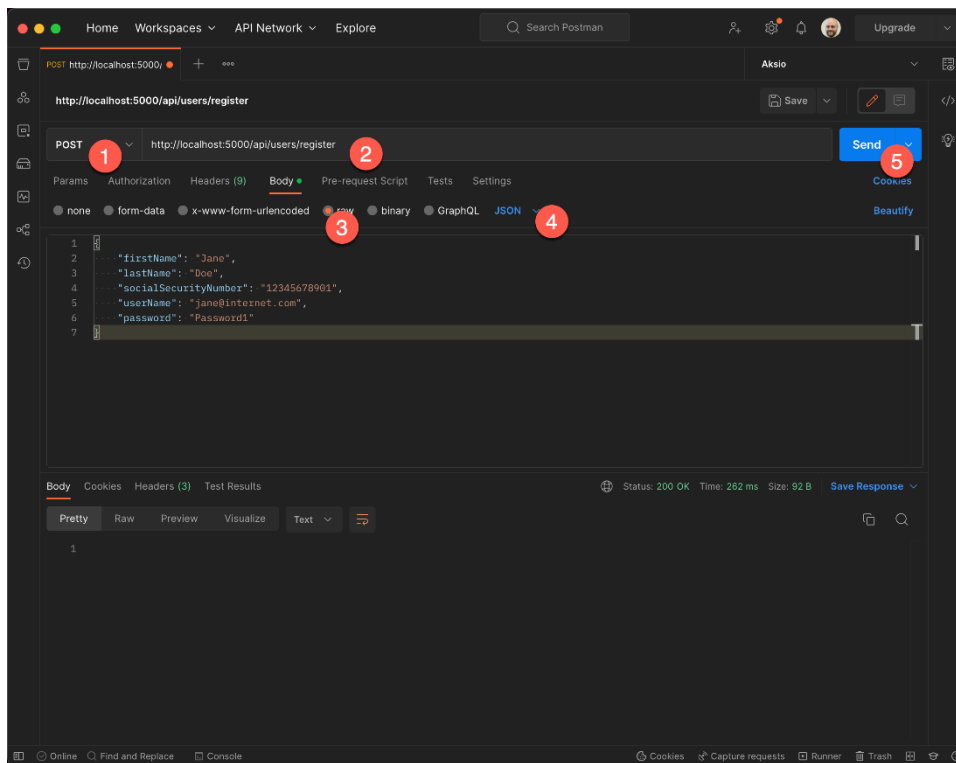
## Chapter 8: Building and Executing Expressions

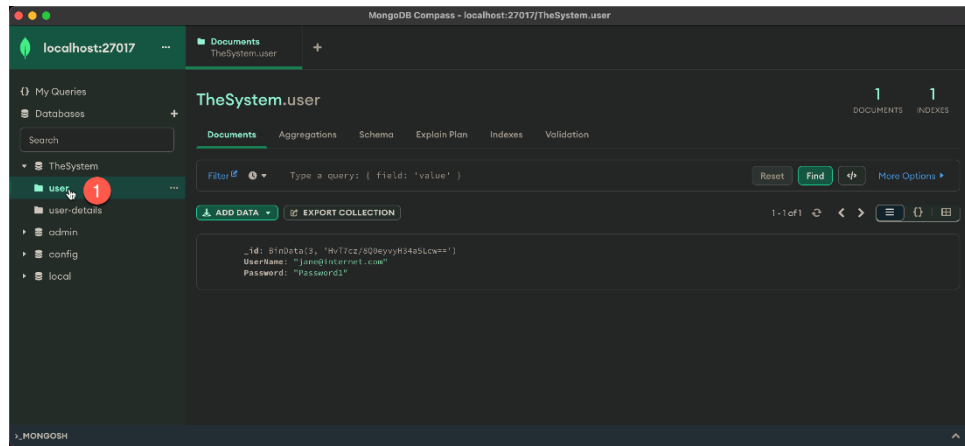
*No images...*

## Chapter 9: Taking Advantage of the Dynamic Language Runtime

*No images...*

## Chapter 10: Convention over Configuration

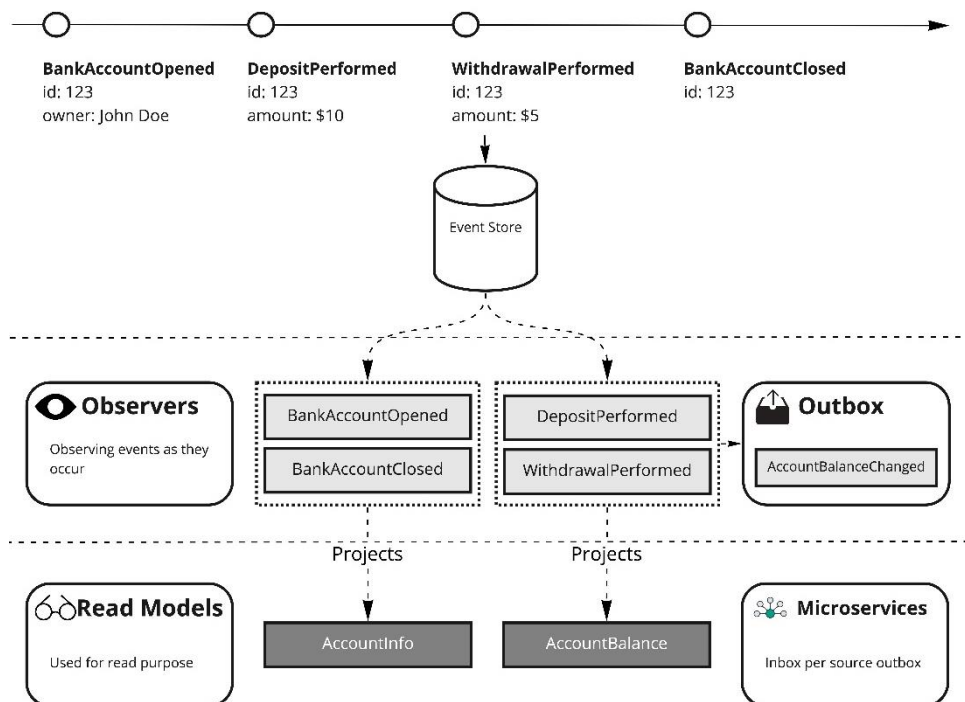
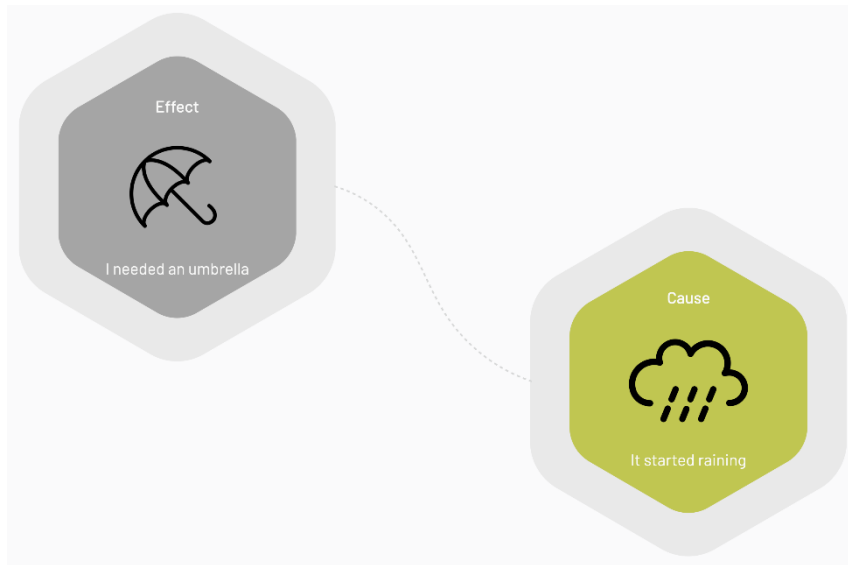


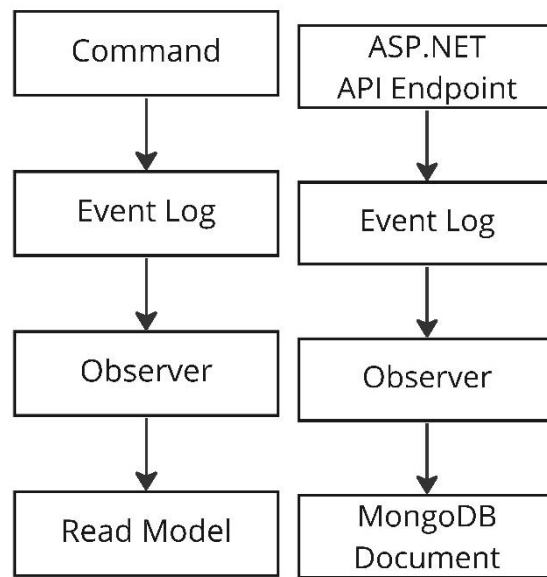


## Chapter 11: Applying the Open-Closed Principle

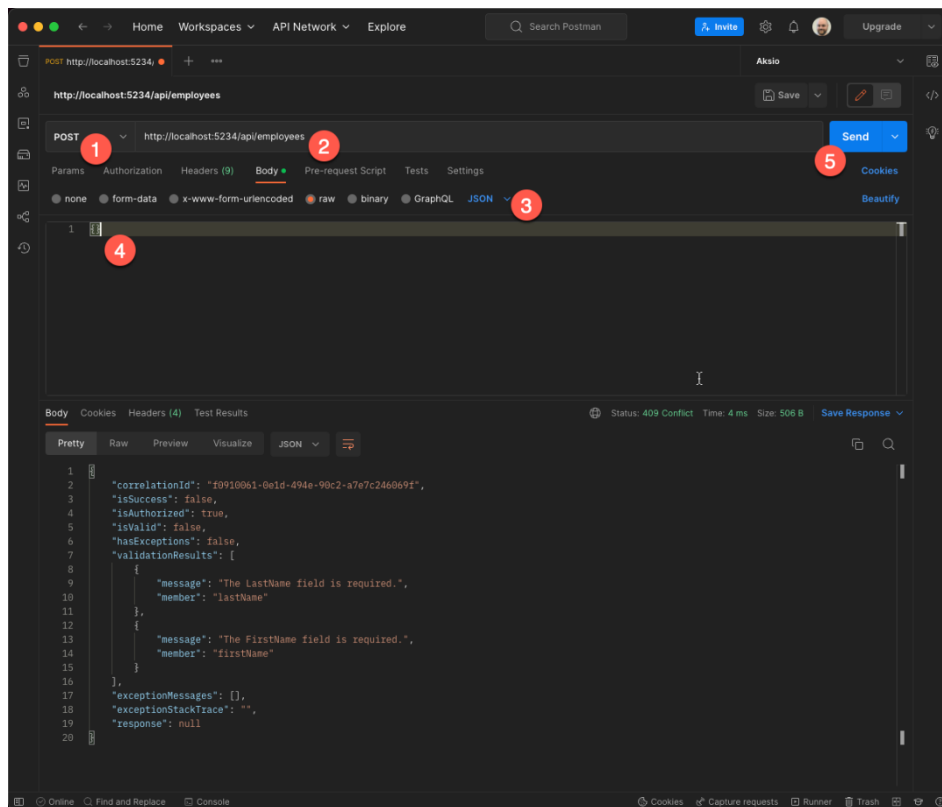
*No images...*

## Chapter 12: Go Beyond Inheritance





## Chapter 13: Applying Cross-Cutting Concerns



HomeWorkspacesAPI NetworkExplore

Search Postman

Invite

Upgrade

POST http://localhost:5234/

\*\*\*

Aksio

http://localhost:5234/api/employees

Save

POST

http://localhost:5234/api/employees

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

1

2

3

4

Body

Cookies

Headers (4)

Test Results

Status: 200 OK

Time: 5 ms

Size: 362 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

Online

Find and Replace

Console

Cookies

Capture requests

Runner

Trash

```
1 {
2   "firstName": "Jane",
3   "lastName": "Doe"
4 }
```

```
1 {
2   "correlationId": "f44600ee-02f2-4d0c-9187-f02ff02c9353",
3   "isSuccess": true,
4   "isAuthorized": true,
5   "isValid": true,
6   "hasExceptions": false,
7   "validationResults": [],
8   "exceptionMessages": [],
9   "exceptionStackTrace": "",
10  "response": 1
11 }
```

Home Workspaces API Network Explore Search Postman Invite Upgrade

POST http://localhost:5234/ + \*\*\*

Aksio

Save

Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

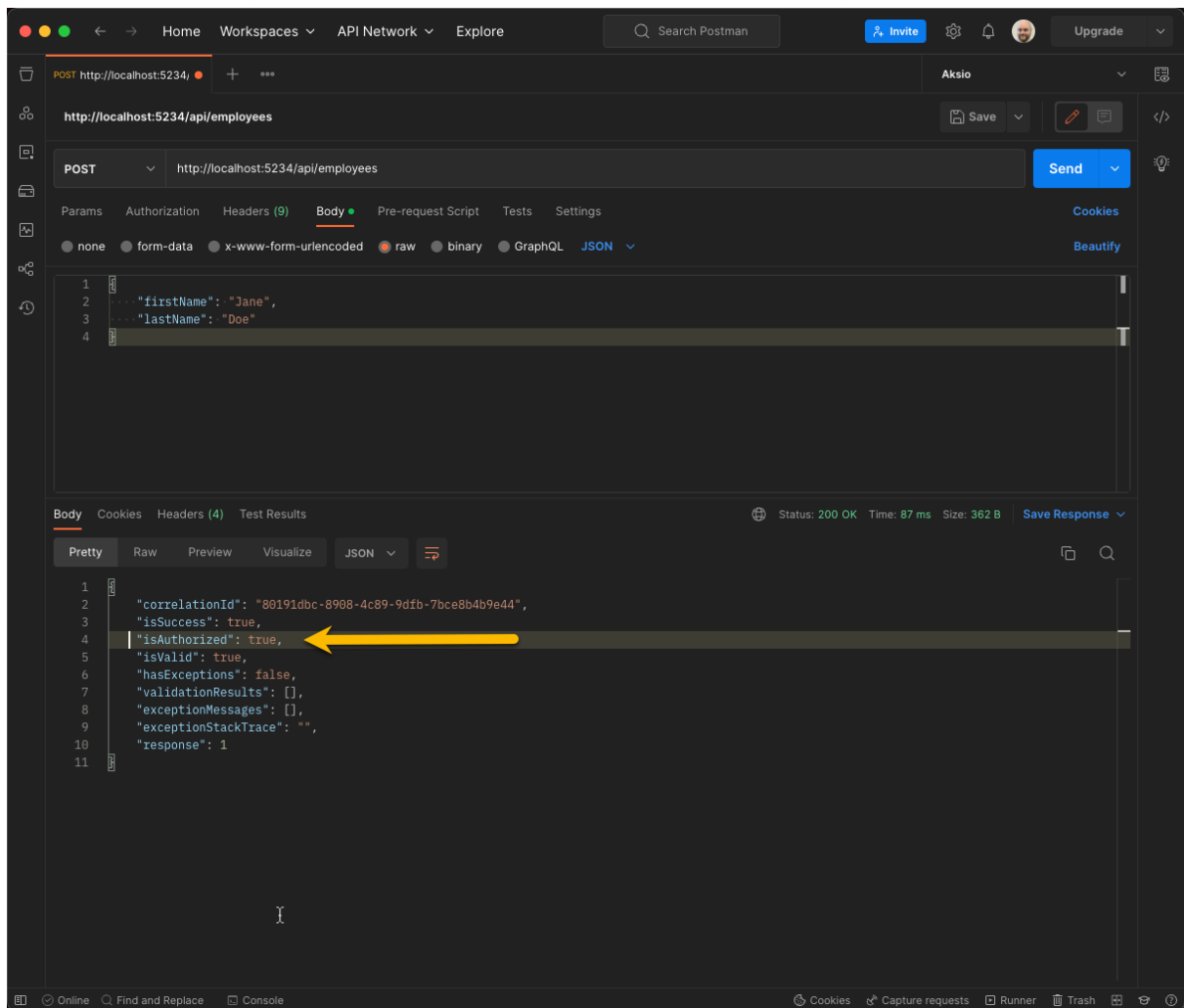
```
1 {
2   "firstName": "Jane",
3   "lastName": "Doe"
4 }
```

Body Cookies Headers (4) Test Results Status: 401 Unauthorized Time: 92 ms Size: 377 B Save Response

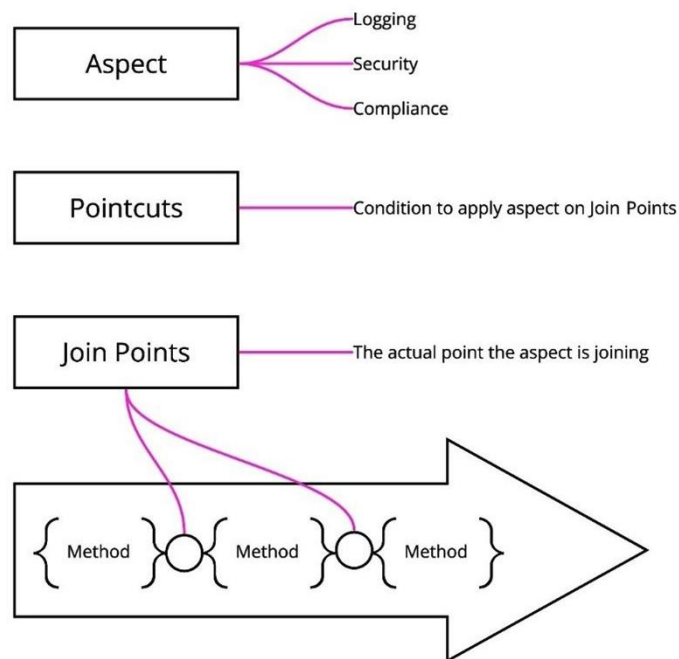
Pretty Raw Preview Visualize JSON

```
1 {
2   "correlationId": "eb26b553-45f3-43a4-9d92-9860b2fe541e",
3   "isSuccess": false,
4   "isAuthorized": false,
5   "isValid": true,
6   "hasExceptions": false,
7   "validationResults": [],
8   "exceptionMessages": [],
9   "exceptionStackTrace": "",
10  "response": null
11 }
```

Online Find and Replace Console Cookies Capture requests Runner Trash



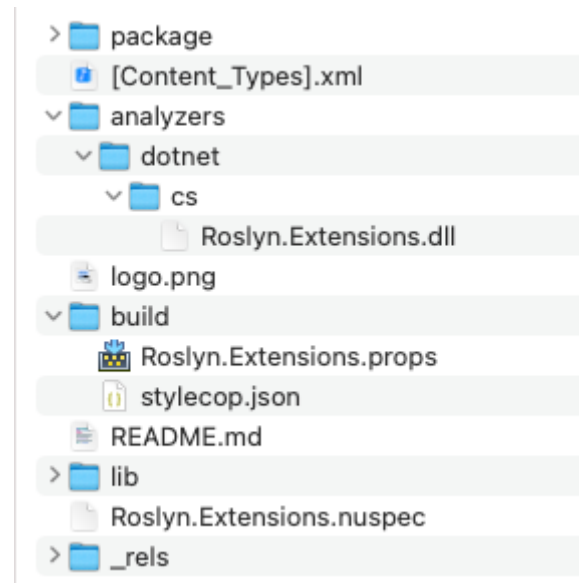
## Chapter 14: Aspect Oriented Programming



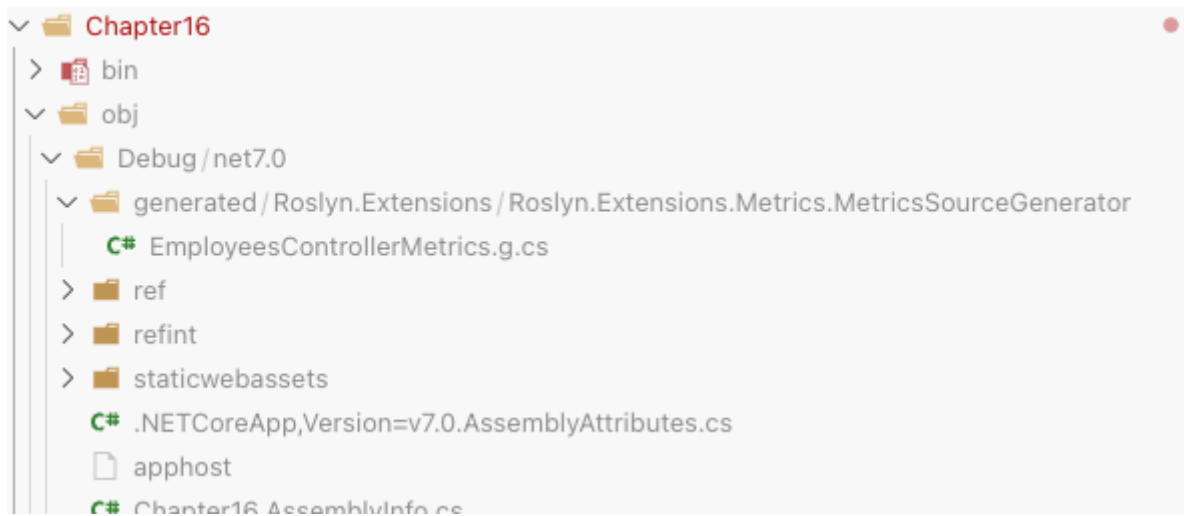
```
31 var userService = container.Resolve<IUserService>();  
32 var result = await userService.Register("jane@doe.io", "Password1");  
33
```

```
→ userService.GetType().GetInterfaces()  
  √ {System.Type[4]}  
    > [0] [Type]: {Chapter14.IUserService}  
    > [1] [Type]: {Chapter14.IAuthenticator}  
    > [2] [Type]: {Chapter14.IAuthorizer}  
    > [3] [Type]: {Castle.DynamicProxy.IProxyTargetAccessor}
```

## Chapter 15: Roslyn Compiler Extensions



## Chapter 16: Generating Code



dotnet 5608

VBCSCompiler.dll -pipename:Lun2X+71zJcK+oOxL0zg8291WixFLGXxWR0Hju6iJgU

## Chapter 17: Static Code Analysis

C# MyException.cs 1 X

Chapter17 > C# MyEx

You, yesterday | 1 a

1 namespace Chapter17

2 class Chapter17.MyException

You, yesterday | 1 a

3 public class MyException : Exception

4

5

6

The use of the word 'Exception' should not be added as a suffix – create a well understood and self explanatory name for the exception [Chapter17] csharp(PP0001)

View Problem (\F8) Quick Fix... (%.)

You, yesterday \* Adding batchfixer ...

C# MyException.cs 1 X

Chapter17 > C# MyException.cs > {} Chapter17 > MyException

You, yesterday | 1 author (You)

1 namespace Chapter17;

2

You, yesterday | 1 author (You)

3 public class MyException : Exception

4 {

5 }

6

Quick Fix...

Remove Exception suffix

More Actions...

- Generate base constructors
- Implement IEquatable<MyException>
- Generate overrides...
- Generate Equals(object)...
- Generate Equals and GetHashCode...
- Generate constructor 'MyException()'
- Generate constructor 'MyException(info, context)'
- Generate constructor 'MyException(message)'
- Generate constructor 'MyException(message, innerException)'
- Generate all
- Add 'DebuggerDisplay' attribute

## Chapter 18: Caveats and Final Words

*No images...*