Working with Data-driven Tests

Kevin Dockx
Architect

@KevinDockx https://www.kevindockx.com

Coming Up

`[Fact]` versus `[Theory]`

Introducing data-driven tests

- Inline data
- Member data
- Class data
- Type-safe approach
- Data from an external source

Introducing Theories and Data-driven Tests

```
[Fact]
public void
CreateInternalEmployee_InternalEmployeeCreated_MustHaveAttendedFirstObligatoryCourse()
{    ...
     // Assert
     Assert.Contains(internalEmployee.AttendedCourses, course =>
     course.Id == Guid.Parse("37e03ca7-c730-4351-834c-b66f280cdb01"));_
}

[Fact]
public void
CreateInternalEmployee_InternalEmployeeCreated_MustHaveAttendedSecondObligatoryCourse()
{    ...
     // Assert
     Assert.Contains(internalEmployee.AttendedCourses, course =>
     course.Id == Guid.Parse("1fd115cf-f44c-4982-86bc-a8fe2e4ff83e"));
}
```

Introducing Theories and Data-driven Tests

```
[Fact]
public void
CreateInternalEmployee_InternalEmployeeCreated_MustHaveAttendedFirstObligatoryCourse()
{    ...
     // Assert
     Assert.Contains(internalEmployee.AttendedCourses, course =>
     course.Id == Guid.Parse("37e03ca7-c730-4351-834c-b66f280cdb01"));
}

[Fact]
public void
CreateInternalEmployee_InternalEmployeeCreated_MustHaveAttendedSecondObligatoryCourse()
{    ...
     // Assert
     Assert.Contains(internalEmployee.AttendedCourses, course =>
     course.Id == Guid.Parse("1fd115cf-f44c-4982-86bc-a8fe2e4ff83e"));
}
```

Introducing Theories and Data-driven Tests

```
[Fact]
public void
CreateInternalEmployee_InternalEmployeeCreated_AttendedCoursesMustMatchObligatoryCourses()
{
    // Arrange
    var obligatoryCourses =
    _employeeServiceFixture.EmployeeManagementTestDataRepository.GetCourses(
        Guid.Parse("37e03ca7-c730-4351-834c-b66f280cdb01"),
        Guid.Parse("1fd115cf-f44c-4982-86bc-a8fe2e4ff83e"));

    // Act
    var internalEmployee =_employeeServiceFixture
        .EmployeeService.CreateInternalEmployee("Brooklyn", "Cannon");

    // Assert
    Assert.Equal(obligatoryCourses, internalEmployee.AttendedCourses);
}
```

Introducing Theories and Data-driven Tests

Adding additional courses would mean:

- Writing additional tests
- Changing the existing test

Fact

A test which is always true. They test invariant conditions.

Theory

A test which is only true for a particular set of data.

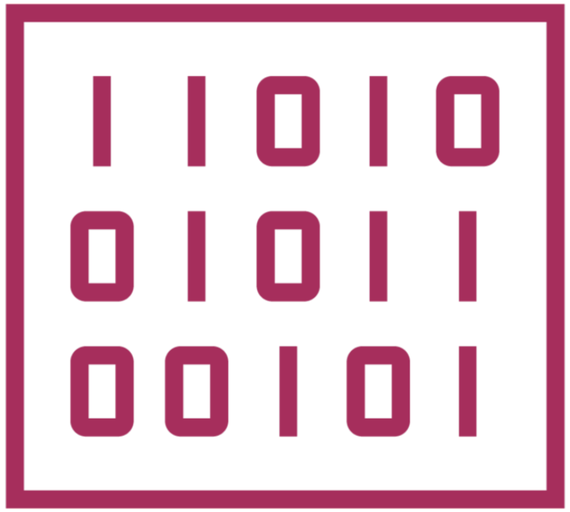
Introducing Theories and Data-driven Tests

```
[Fact]
public void
CreateInternalEmployee_InternalEmployeeCreated_MustHaveAttendedFirstObligatoryCourse()
{    ...
     // Assert
     Assert.Contains(internalEmployee.AttendedCourses, course =>
     course.Id == Guid.Parse("37e03ca7-c730-4351-834c-b66f280cdb01"));
}

[Fact]
public void
CreateInternalEmployee_InternalEmployeeCreated_MustHaveAttendedSecondObligatoryCourse()
{    ...
     // Assert
     Assert.Contains(internalEmployee.AttendedCourses, course =>
     course.Id == Guid.Parse("1fd115cf-f44c-4982-86bc-a8fe2e4ff83e"));
}
```

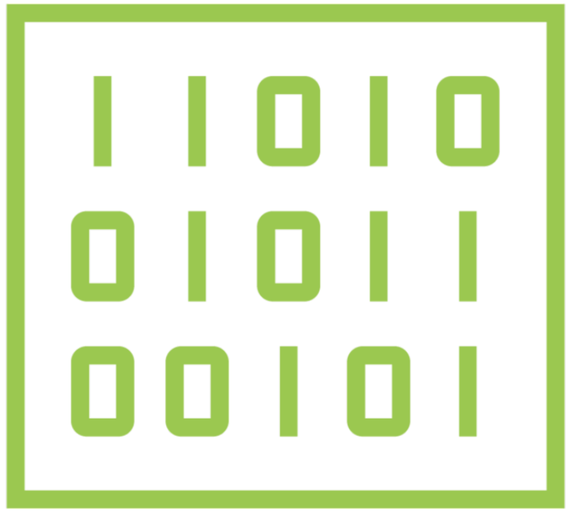
Introducing Theories and Data-driven Tests

```
[Theory]
public void
CreateInternalEmployee_InternalEmployeeCreated_MustHaveAttendedObligatoryCourse(
Guid courseId)
{
    // Arrange

    // Act
    var internalEmployee = _employeeServiceFixture
        .EmployeeService.CreateInternalEmployee("Brooklyn", "Cannon");

    // Assert
    Assert.Contains(internalEmployee.AttendedCourses,
        course => course.Id == courseId);
}
```

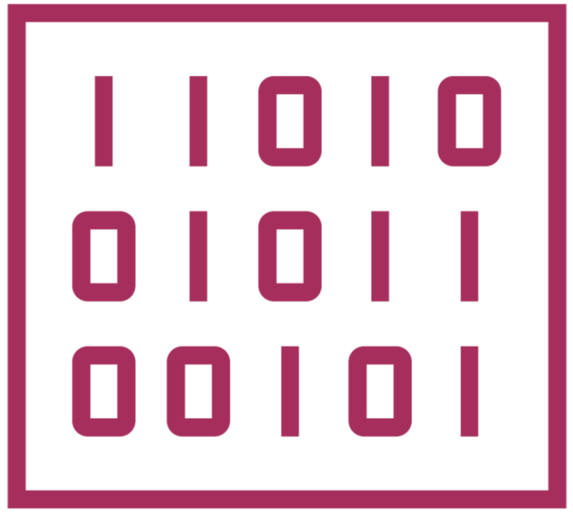
Introducing Theories and Data-driven Tests



[InlineData]



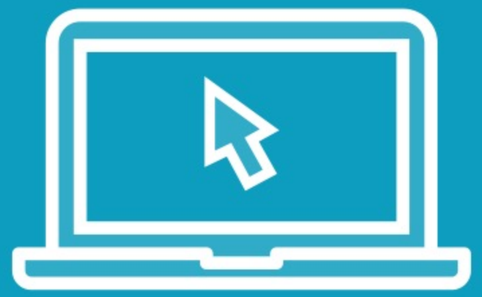
[MemberData]



[ClassData]

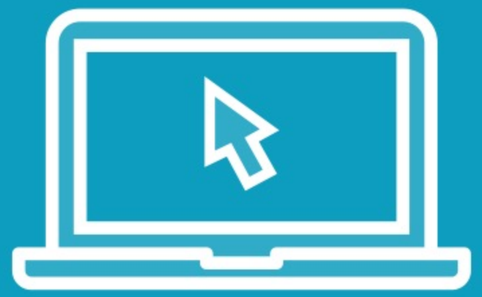
Use `TheoryData` for type-safe data

Demo



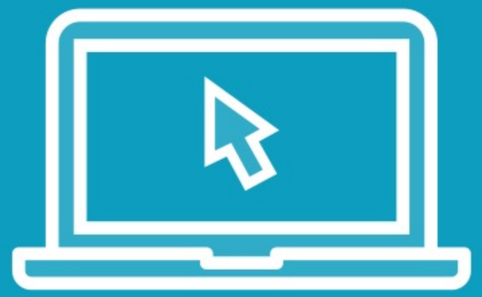
Testing a theory with inline data

Demo



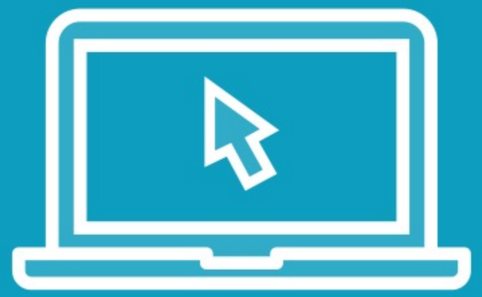
Testing a theory with member data

Demo



Testing a theory with class data

Demo



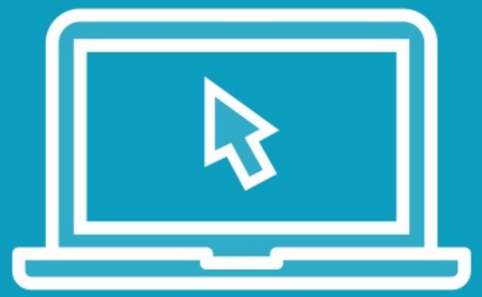
Testing a theory with strongly-typed test data

Getting Data From an External Source

Test data can come from an external source

- Other people can manage it
- Convenient for, for example, the QA team

Demo



Getting data from an external source

Summary

`[Theory]` enables data-driven tests

- Inline data
- Member data
- Class data
- Strongly-typed data with `TheoryData`
- Getting data from an external source

Up Next:

Isolating Unit Tests with ASP.NET Core Techniques and Mocking