

Fake It Easy

Philipp Dolder



What it is

Amazingly simple mocking framework

Created by Patrik Hägne

<https://github.com/FakeItEasy/FakeItEasy>

<http://www.nuget.org/packages/FakeItEasy>

Concepts

Everything is a fake

Setups are put on a stack

Default behavior

Default values for primitive types

Fakes for complex types

How to use it

Install-Package FakeItEasy

```
Interface f = A.Fake<Interface>();
```

```
A.CallTo(() => f.GetValue()).Returns(5);
```

```
A.CallTo(() => f.DoIt()).MustHaveHappened();
```

Create fakes

```
Interface f = A.Fake<Interface>();
```

```
IList<Interface> fakes =  
    A.CollectionOfFake<Interface>(3);
```

Setup (1)

```
f.Property = 5;
```

```
A.CallTo(() => f.ReadOnly).Returns(8);
```

```
A.CallTo(() => f.GetValue()).Returns(13);
```

Setup (2)

```
A.CallTo(() => f.GetValue())  
  .ReturnsNextFromSequence(5, 8, 13, 21);
```

```
int i = 0;
```

```
A.CallTo(() => f.GetValue())  
  .ReturnsLazily(() => i++);
```

Assert (1)

```
A.CallTo(() => f.DoIt()).MustHaveHappened();
```

```
A.CallTo(() => f.DoIt())
```

```
.MustHaveHappened(Repeated.AtLeast.Once);
```

```
A.CallTo(() => f.DoIt()).MustNotHaveHappened();
```

```
A.CallTo(() => f.DoIt())
```

```
.MustHaveHappened(Repeated.Never);
```


Assert (2)

- `A.CallTo(() => f.DoIt())
.MustHaveHappened(Repeated.AtLeast.Twice);`
- `A.CallTo(() => f.DoIt())
.MustHaveHappened(Repeated.Exactly.Times(3));`
- `A.CallTo(() => f.DoIt())
.MustHaveHappened(Repeated.NoMoreThan.Once);`

Argument Matching (1)

```
A.CallTo(() => f.Request(3))  
  .Returns(5);
```

```
A.CallTo(() => f.Request(3))  
  .MustHaveHappened();
```

Argument Matching (2)

```
A<int>._
```

```
A<int>.Ignored
```

```
A<int>.That.Matches(i => i == 5)
```

```
A<string>.That.StartsWith("g")
```

```
A<string>.That.Not.IsEmpty()
```

Raise events

```
f.Trigger += Raise.WithEmpty().Now;
```

```
f.Trigger += Raise.With(new EventArgs()).Now;
```

```
f.Trigger +=  
    Raise.With(this, new EventArgs()).Now;
```

Access invocation arguments (1)

```
A.CallTo(() => f.Request(A<int>._))  
  .ReturnsLazily((int i) => i + 5);
```

Supports up to 4 arguments

Signature has to match

Access invocation arguments (2)

```
A.CallTo(() => f.Request(A<int>._))  
  .ReturnsLazily(call =>  
    call.GetArgument<int>(0) + 5);
```

Invocation actions (1)

```
A.CallTo(() => f.Request(A<int>._))  
  .Invokes(() => i++);
```

```
A.CallTo(() => f.Request(A<int>._))  
  .Invokes((int a) => i += a);
```

Invocation actions (2)

```
A.CallTo(() => f.Request(A<int>._))  
  .Invokes(call => i +=  
    call.GetArgument<int>(0));
```


Throw exceptions

```
A.CallTo(() => f.GetValue())  
  .Throws<Exception>();
```

```
A.CallTo(() => f.GetValue())  
  .Throws(new Exception());
```

```
A.CallTo(() => f.Request(A<int>._))  
  .Throws((int i) => new MyException(i));
```

The winner fakes it all

Philipp Dolder

philipp.dolder@bbv.ch

Get latest packages

Latest release from NuGet

Install-Package FakeItEasy

Latest pre release from CodeBetter

Feed url: <http://teamcity.codebetter.com/guestAuth/app/nuget/v1/FeedService.svc>

Install-Package FakeItEasy -IncludePrerelease