Bending AutoFixture to your will

Mark Seemann
@ploeh

http://www.flickr.com/photos/barnshaws/5757943464/
Agenda

- Slides
- Code
- Code
- Code
- Code
- Code
- Code
Goals

- AutoFixture provides feedback – listen to it
- You can modify AutoFixture’s behavior to fit your needs
- Favor conventions over specific modifications
public class ContactBuilder
{
    private readonly string n;
    private readonly string p;

    public ContactBuilder() : this("Jane Doe", "55512345") { }

    private ContactBuilder(string name, string phone)
    {
        this.n = name;
        this.p = phone;
    }

    public ContactBuilder WithName(string name)
    {
        return new ContactBuilder(name, this.p);
    }

    public ContactBuilder WithPhone(string phone)
    {
        return new ContactBuilder(this.n, phone);
    }

    public Contact Build()
    {
        return new Contact { Name = this.n, Phone = this.p };  
    }
}
ContactBuilder examples

```javascript
var contact = new ContactBuilder().Build();

var contact = new ContactBuilder().WithName("John Doe").Build();

var contact = new ContactBuilder().WithPhone("12345678").Build();
```
Test Data Builder

A dream to use

A pain to maintain

A TDB for every SUT
AutoFixture

Takes away the pain of maintaining TDBs

Reflection-based

Convention-based
Replace ContactBuilder with AutoFixture
What just happened?

The concrete ContactBuilder was replaced with AutoFixture
Customize the Fixture

DEMO
What just happened?

The phone rule was encapsulated in a class

This provides a central place to manage the rule
Move towards a declarative style
What just happened?

Taking advantage of xUnit.net’s extensibility model, creation of object instances were made declarative.

The rest of the talk doesn’t hinge on this xUnit.net feature at

It’s just a cool thing to do

h:p://www.flickr.com/photos/omcoc/6751047205/
Create unique phone numbers

DEMO
What just happened?

Phone numbers were made unique to a Fixture

Partial delegation to Fixture’s ability to create unique numbers

http://www.flickr.com/photos/omcoc/6751047205/
What just happened?

- Fixture was turned into an Auto-mocking Container
- Moq was used
- RhinoMocks also supported OOB
- FakeItEasy also supported OOB
- All are extensions

http://www.flickr.com/photos/omcoc/6751047205/
Circular references
What just happened?

Circular references are a design smell

However, they can be dealt with by changing the behavior of Fixture

http://www.flickr.com/photos/omcoc/6751047205/
Conventions for phone numbers

DEMO
What just happened?

Manual edit using the Fluid Builder API

Convention based on PropertyInfo

Value resolved via int

Value resolved via RangedNumberRequest

http://www.flickr.com/photos/omcoc/6751047205/
To go

- If AutoFixture can’t create an instance of your API, it’s probably your fault
- Use AutoFixture’s many extensibility points to make it behave like you’d like
- Strive towards a single set of conventions per test project
AutoFixture

- [http://stackoverflow.com/questions/tagged/autofixture](http://stackoverflow.com/questions/tagged/autofixture)
- Nuget: AutoFixture

Mark Seemann

- [http://blog.ploeh.dk/](http://blog.ploeh.dk/)
- @ploeh

Nikos Baxevanis

- @nikosbaxevanis

Enrico Campidoglio

- [http://megakemp.wordpress.com/](http://megakemp.wordpress.com/)
- @ecampidoglio