

# Unit Testing Things That Are Hard To Test

---

David Berry

@DavidCBerry13

<https://github.com/DavidCBerry13/unit-testing-hard-things/>





# THANK YOU, THAT CONFERENCE SPONSORS!



Created by  
ThoughtWorks®



# Why Do We Write Unit Tests



Make sure code is  
working as intended



Helps us think about  
failure scenarios



Automate as much of  
testing as possible

This talk is about providing you different **options** to solve some of the harder to test scenarios in your code

# What Are Things That Are Hard To Test

Static Methods

Legacy Code

File Operations

HTTP Calls

Database Calls

# Testing Static Methods

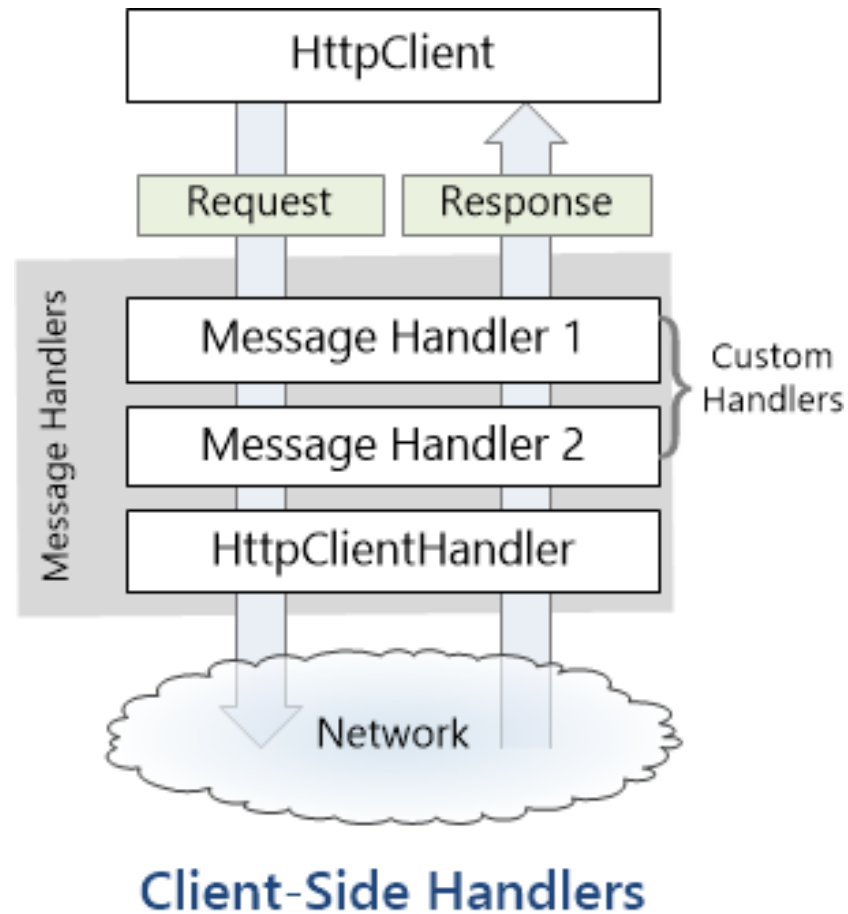
## Partial Mock Solution

- Extract the call to the static method into a helper method on the class under test
- Make sure the helper method is marked virtual
- Use a partial mock to substitute the value you want at test time

## Extract Interface Solution

- Create an interface to with the static methods on the class you need to call
- Create a default implementation of the interface that makes calls to the static methods
- Provide a constructor where the interface can be injected in

# HttpClient Architecture



Just because some scenarios appear hard to test doesn't mean they can't be tested.

Use the techniques and libraries discussed in this talk to drive automated testing across more of your code base.



# Resources

## System.IO.Abstractions (Library)

<https://github.com/tathamoddie/System.IO.Abstractions>

## Pose (Library)

<https://github.com/tonerdo/pose>

## Mocking Sequences Revisited (Phil Haack Blog Post)

<https://haacked.com/archive/2010/11/24/moq-sequences-revisited.aspx/>

# Thank You

David Berry

@DavidCBerry13

<https://github.com/DavidCBerry13/unit-testing-hard-things/>

