

Unit Testing Your Code



Joe Eames

WEB DEVELOPER

@josepheames www.joeeames.me



Agenda

Introduction

Karma

Services

Isolated Component Tests



Unit Test

A test of a single “unit” of code.



Unit Tests vs End to End Tests

Unit Test

Fast

**Involve Isolated Pieces of
Code**

End to End Test

Slow

Exercises the Entire System



Good Unit Tests



Fast

Cheap to Write

Single State Change

Assert 1 Thing

Doesn't Cross Process Boundaries

Reliable



Unit Test Structure

AAA

Arrange

Act

Assert

Damp

Can Have Some Duplication

Tells the Story



```
var user = new User("Sally");  
user.friends = ["Ralph"]  
  
user.addFriend("John");  
  
expect(user.friends.length)  
    .toBe(2);
```

◀ Arrange

◀ Act

◀ Assert

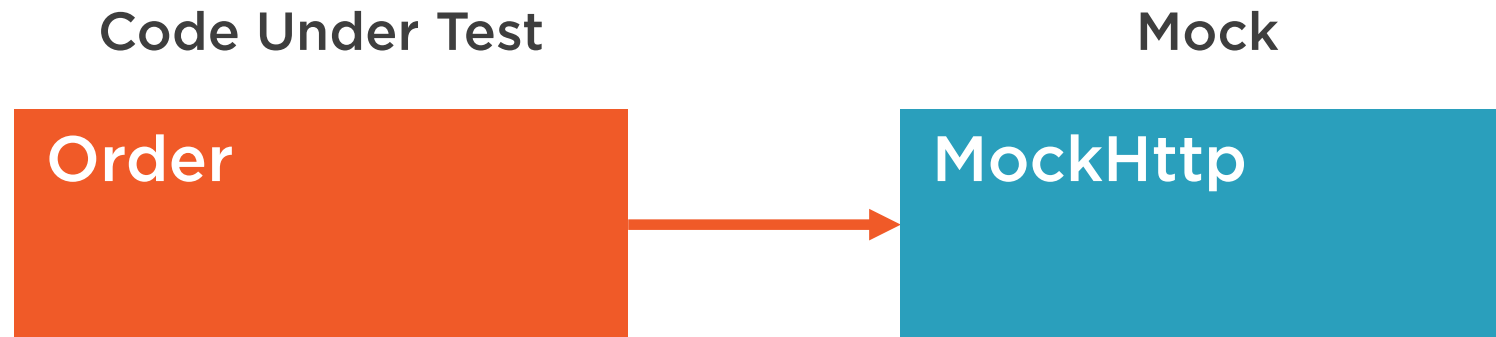


```
var user = new User("Sally");  
user.friends = ["Ralph"]  
  
user.addFriend("Ralph");  
  
expect(user.friends.length)  
    .toBe(1);
```

◀ Remove Duplication?



Mocking



Jasmine

describe()

beforeEach()

it()

expect()

Matchers

- **toBe()**
- **toContain()**
- **toBeDefined()**



Karma

Runs Tests in Browser

Multiple Browsers

Reports Test Run Results



Isolated vs Integrated Tests

Isolated Tests

Test Class Only – No Template

Constructed in Test

Simple

Best for Services & Pipes

Appropriate for Components &
Directives

Integrated Tests

Test Class & Template

Constructed by Framework

Complex

Mainly Used for Components & Directives

Sometimes Used for Services

Deep or Shallow



Summary



Karma & Jasmine
Services
Mocking
Isolated Component Tests

