

AUTOMATTIC  
WordPress, Akismet, WooCommerce, LinkedIn, SoundCloud, Email

# Thanks to Our Sponsors

**RingCentral**<sup>®</sup>

**twilio**

**nexmo**<sup>®</sup>  
The Vonage<sup>®</sup> API Platform

**OSMI**

**mailchimp**

**wayfair**<sup>®</sup>

## Stay Connected



## Wifi

Network: LCC  
PW: iluvmeeting

## #phptek

WELCOME TO  
THE TESTING  
TRACK

EGO-CENTRIC  
INTRODUCCION  
GOES HERE

WHAT DO TESTS  
LOOK LIKE?

# WHAT DO TESTS LOOK LIKE?

THEY ARE CODE THAT USES OTHER CODE

# WHAT DO TESTS LOOK LIKE?

THEY CAN BE EXECUTED BY AUTOMATED SYSTEMS

**WHAT DO TESTS LOOK LIKE?**

**THEY CLEARLY REPORT A PASS OR FAIL**

HOW ARE TESTS  
STRUCTURED?



# HOW ARE TESTS STRUCTURED?

USE THE ARRANGE-ACT-ASSERT PATTERN

# ARRANGE

ALL THE SETUP WORK THAT IS REQUIRED IN  
ORDER TO EXECUTE YOUR TESTING SCENARIO

# ARRANGE

- > TEST DATA
- > EXPECTED RESULTS
- > OBJECTS UNDER TEST

# ARRANGE

USUALLY THE LARGEST PART OF ANY TESTING SCENARIO

# ACT

WHERE YOU GO THROUGH ALL THE STEPS OF YOUR  
TESTING SCENARIO

# ACT

- > USES CODE (HOPEFULLY) AS IT WOULD BE USED IN THE APPLICATION
  - > ACTS AS STEALTH DOCUMENTATION

# ACT

USUALLY THE SHORTEST PART OF A TESTING SCENARIO

# ASSERT

WHERE YOU VERIFY THAT YOUR TESTING SCENARIO IS  
BEHAVING AS EXPECTED



# ASSERT

- > BUILT-IN ASSERTIONS
- > VERIFYING EXCEPTIONS

# CLEAN-UP

**ALWAYS MAKE SURE TO CLEAN UP ANY SIDE-EFFECTS OF TESTING SCENARIOS**

# CLEAN-UP

- > ANY TEXT FILES THAT WERE CREATED
- > ANY NEW OR UPDATED RECORDS IN DATA SOURCES

# TESTING BUILDING BLOCKS

- > ASSERTIONS
- > TEST DOUBLES

# TESTING BUILDING BLOCKS

## ASSERTIONS

CREATED BY THE FOLKS WHO INVENTED NUNIT STYLE TESTING IN  
THE JAVA WORLD

# TESTING BUILDING BLOCKS

## ASSERTIONS

REVOLVES AROUND A VERY SIMPLE CONCEPT: PROVE THAT THE OBSERVED BEHAVIOUR IS WHAT YOU EXPECT

# TESTING BUILDING BLOCKS

## ASSERTIONS

```
$this->assertTrue(<something>);  
$this->assertFalse(<something>);  
$this->assertEquals($foo, $bar);
```

# TESTING BUILDING BLOCKS

## ASSERTIONS

THESE COVER 99% OF EVERYTHING I'VE EVER NEEDED TO ASSERT



TESTING  
BUILDING  
BLOCKS  
DOUBLES

# TESTING BUILDING BLOCKS

(YOU PROBABLY CALL

# BUILDING BLOCKS

THEY ARE USED WHEN  
YOU NEED A DEPENDENCY  
FOR A TEST AND DON'T

# TESTING BUILDING BLOCKS

## DOUBLES

- > FAKES
- > DUMMIES
- > STUBS
- > MOCKS

# TESTING BUILDING BLOCKS

## DOUBLES

A FAKE IS WHEN YOU CREATE YOUR OWN VERSION OF A DEPENDENCY AND DON'T CARE IF IT MATCHES THE CLASS SIGNATURE

# TESTING BUILDING BLOCKS

## DOUBLES

A DUMMY IS WHEN YOU CREATE A VERSION OF A DEPENDENCY THAT HAS THE SAME CLASS SIGNATURE BUT HAS NO FUNCTIONALITY

# TESTING BUILDING BLOCKS

## STUBS

A STUB IS WHEN YOU CREATE A VERSION OF A DEPENDENCY THAT HAS THE SAME CLASS SIGNATURE AND THE METHODS YOUR TEST NEEDS DEFINED

# TESTING BUILDING BLOCKS

## MOCK

A MOCK IS WHEN YOU CREATE A VERSION OF A DEPEDENCY THAT HAS THE SAME CLASS SIGNATURE, THE METHODS YOUR TEST NEEDS DEFINED, AND METHODS RETURN VALUES



TESTING  
BUILDING  
BLOCKS  
DATA PROVIDERS

# TESTING BUILDING BLOCKS

## DATA PROVIDERS

- > **ALLOWS THE CREATION OF MORE GENERIC TESTS**

# TESTING BUILDING BLOCKS

## DATA PROVIDERS

- > **ALLOWS THE CREATION OF MORE GENERIC TESTS**

# TESTING BUILDING BLOCKS

## DATA PROVIDERS

- > **ALLOWS THE CREATION OF MORE GENERIC TESTS**
- > **ALLOWS YOU TO ADD MORE DATA SETS WITHOUT ADDING TESTS**

# TESTING HELPERS

- > TESTS ARE JUST CODE! APPLY THE SAME RULES TO TESTS YOU WRITE

# TESTING HELPERS

- > LOOK FOR DUPLICATION IN YOUR TESTS AND CREATE HELPERS INSTEAD

# LOOPING CONSTRUCTS

- > **LOOPS ARE OKAY IF THE SCENARIO CALLS FOR IT**

# CONDITIONAL STATEMENTS

- > AN IF STATEMENTS MEANS YOU REALLY NEED TWO TESTS



# TL:DR

- > TESTS ARE JUST CODE WITH SOME DIFFERENT RULES AND STRUCTURES APPLIED TO THEM

# SLIDES

[HTTPS://GRUMPY-LEARNING.COM/PRESENTATIONS/TEK2019](https://grumpy-learning.com/presentations/tek2019)