Variety of automated tests

Artjom Kurapov

10 years in IT / PHP
lvl 65 elf/hunter

Kurapov.ee
@tot_ra

Fear of sudden death

QA engineer

Developer

https://pass.rzd.ru/ticket/secure/ru?STRUCTURE_ID=735&layer_id=5420&refererLayerId=5460&id=136113308

ПАССАЖИРАМ   ГРУЗОВЫЕ ПЕРЕВОЗКИ   РАБОТА В РЖД

Поиск

Пассажирам | Поезда дальнего следования | Расписание, стоимость билетов

› Мои заказы

› Правила покупки
электронных билетов

› Поезда дальнего следования

 › Расписание, стоимость
 билетов

  › Оферта

  › Оферта оформления
  электронной карты

  › Оферта страховой
  компании
  ОАО "ЖАСО"

  › Оферта страховой
  компании
  ООО "СО
  "Сургутнефтегаз"

  › Оферта страховой
  компании
  ООО "Росгосстрах"

  › Оферта страховой
  компании
  ООО "Группа
  Ренессанс
  Страхование"

› Дополнительные поезда

Error:
java.lang.NullPointerException

    com.ibm._jsp._finalTicketStep._jspService(_finalTicketStep.java:297)
    com.ibm.ws.jsp.runtime.HttpJspBase.service(HttpJspBase.java:99)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:831)
    com.ibm.ws.webcontainer.servlet.ServletWrapper.service(ServletWrapper.java:1658)
    com.ibm.ws.webcontainer.servlet.ServletWrapper.service(ServletWrapper.java:1598)
    com.ibm.ws.webcontainer.filter.WebAppFilterChain.doFilter(WebAppFilterChain.java:113)
    com.ibm.ws.webcontainer.filter.WebAppFilterChain._doFilter(WebAppFilterChain.java:80)
    com.ibm.ws.webcontainer.filter.WebAppFilterManager.doFilter(WebAppFilterManager.java:908)
    com.ibm.ws.webcontainer.servlet.ServletWrapper.handleRequest(ServletWrapper.java:935)
    com.ibm.ws.webcontainer.servlet.ServletWrapper.handleRequest(ServletWrapper.java:503)
    com.ibm.ws.webcontainer.servlet.ServletWrapperImpl.handleRequest(ServletWrapperImpl.java:181)
    com.ibm.wsspi.webcontainer.servlet.GenericServletWrapper.handleRequest(GenericServletWrapper.java:121)
    com.ibm.ws.jsp.webcontainerext.AbstractJSPExtensionServletWrapper.handleRequest(AbstractJSPExtensionServletWrapper.java:259)
    com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.include(WebAppRequestDispatcher.java:711)
    org.apache.jasper.runtime.JspRuntimeLibrary.include(JspRuntimeLibrary.java:1045)
    org.apache.jasper.runtime.PageContextImpl.include(PageContextImpl.java:527)
    ru.tehnosk.chameleon.layer.TagLayerRenderableClientImpl.includeJsp(TagLayerRenderableClientImpl.java:30)
    ru.tehnosk.chameleon.layer.render.JspLayerDataRendererImpl.renderLayerData(JspLayerDataRendererImpl.java:38)
    ru.tehnosk.chameleon.layer.render.LayerDataRenderer.render(LayerDataRenderer.java:46)
    ru.tehnosk.chameleon.layer.render.LayerDataRenderer.render(LayerDataRenderer.java:42)
    ru.tehnosk.chameleon.theme.LayerRendererTag.doStartTag(LayerRendererTag.java:54)
    com.ibm._jsp._Default._jspx_meth_wps_screenRender_0(_Default.java:2857)
    com.ibm._jsp._Default._jspService(_Default.java:2204)
    com.ibm.ws.jsp.runtime.HttpJspBase.service(HttpJspBase.java:99)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:831)
    com.ibm.ws.webcontainer.servlet.ServletWrapper.service(ServletWrapper.java:1658)
    com.ibm.ws.webcontainer.servlet.ServletWrapper.service(ServletWrapper.java:1598)
    com.ibm.ws.webcontainer.filter.WebAppFilterChain.doFilter(WebAppFilterChain.java:113)
    com.ibm.ws.webcontainer.filter.WebAppFilterChain._doFilter(WebAppFilterChain.java:80)
    com.ibm.ws.webcontainer.filter.WebAppFilterManager.doFilter(WebAppFilterManager.java:908)

REALITY
IS COMPLEX

Objectivity - state or quality of being true even **outside of a subject's individual biases,** interpretations, feelings, and imaginings.

Client – I *dropped* a *phone*
Test that it *works*

Awareness & Clarity

OVERVIEW

BLACK   GRAY   WHITE

TEST FAMILY

SUBJECTIVE IMPORTANCE

Unit tests : exercise the smallest pieces of testable software in the application to determine whether they behave as expected.

Integration tests : verify the communication paths and interactions between components to detect interface defects.

Public Resources

Internal Resources

Service Layer

Gateways

Stub HTTP Client

External Service

Domain

Live HTTP Client

Repositories

Data Mappers / ORM

External Datastore

In Memory Datastore

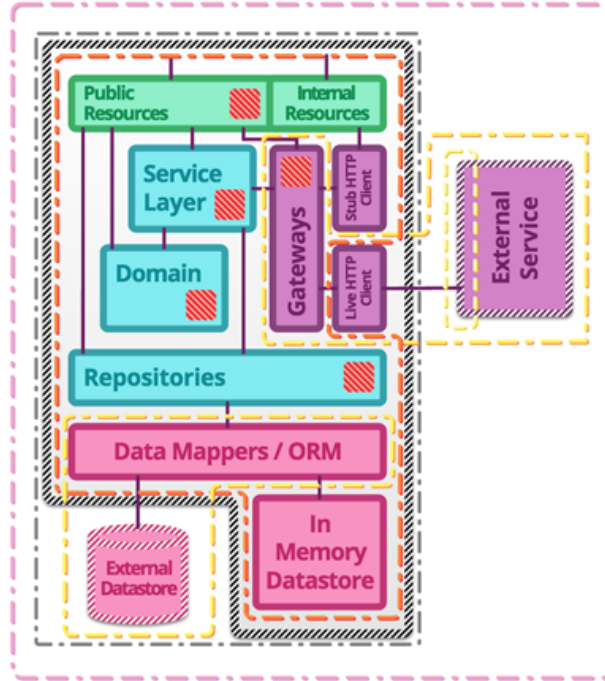**Component tests** : limit the scope of the exercised software to a portion of the system under test, manipulating the system through internal code interfaces and using test doubles to isolate the code under test from other components.

Contract tests : verify interactions at the boundary of an external service asserting that it meets the contract expected by a consuming service.

End-to-end tests : verify that a system meets external requirements and achieves its goals, testing the entire system, from end to end.

Martin Fowler http://martinfowler.com/articles/microservice-testing/

GITHUB LINKS

PHP testing tools & authors

Sebastian Bergmann
@s_bergmann

Ciaran McNulty
@ciaranmcnulty

Michael Bodnarchuk
@davert

Konstantin Kudryashov
@everzet

atoum/atoum
mlively/Phake
ribozz/PhpSpock
marco-fiset/Testify.php
SimpleTest
inclusive-design/AChecker

LEVEL 1 **BLACK BOX**

Scenario tests

Basic scenario tests

- Create a feature list
  - Register as new user
  - Register with coupon
  - Login
  - Checkout
- Timeouts between commands
- Hard-coded data

Codeception/Codeception
giorgiosironi/phpunit-selenium
minkphp/MinkSelenium2Driver

UI testing tools

- Browser controllers – Selenium, Sahi
- Headless browsers
  - CasperJS, PhantomJS, Goutte
- Custom stuff - Watir, TestComplete, iMacros

Codeception/Codeception
giorgiosironi/phpunit-selenium
minkphp/MinkSelenium2Driver

```php
class RegistrationNormalTest extends PHPUnit_Extensions_Selenium2TestCase {
    protected function setUp() {
        $this->setBrowser('firefox');
        $this->setBrowserUrl('http://fake.gratheon.com/rg');
    }

    protected function loginAs($login, $pass) {
        $this->url('http://fake.gratheon.com/en/login');
        $this->byId('inputEmail')->value($login);
        $this->byId('inputPassword')->value($pass);
        $this->byCssSelector('.btn-primary')->click();
        $this->assertEquals('Logout', $this->byCssSelector('.exit')->text());
    }

    /** @test ... */
    public function registerNormal() {
        $this->url('http://fake.gratheon.com/en/rg');

        $this->byId('fake_register_referralCode')->value("001");
        $this->byId('fake_register_email')->value("artkurapov+7@gmail.com");
        $this->byId('fake_register_firstName')->value("Temporary");
        $this->byId('fake_register_lastName')->value("User");
        $this->byId('fake_register_middleName')->value("Mid");
```

Sum of complexity

- **Not informative** if fails

- **Environment** maintenance

- New tech
  - dragging, canvas, svg,
  - touch, multitouch, 3d-touch

Codeception/Codeception
giorgiosironi/phpunit-selenium
minkphp/MinkSelenium2Driver

Instability & non-determinism

- **Async** - ajax, animations, upload
- **HTML-dependent**
- **Hard-coded data & dependent tests**
- **External websites**
- **Drivers**

Codeception/Codeception
giorgiosironi/phpunit-selenium
minkphp/MinkSelenium2Driver
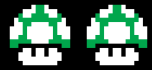
Abstracting QA framework

- Syntax improvement
- Page objects + Workflows
- API for data reset / VM per process
- Repeating functions
  - Login, Messages
  - Form saving, Layout mode

Codeception/Codeception
giorgiosironi/phpunit-selenium
minkphp/MinkSelenium2Driver

Behaviour tests

- Human-readable specification
- Needs parsing
- Business people-oriented

Behat/Behat

Feature: Addition
In order to avoid silly mistakes
As a math idiot
I want to be told the sum of two numbers

Scenario:
Given I have an calculator
When I have entered 30 as first number
And I have entered 20 as second number
And I press 'Add'
Then The result should be 50

```php
$steps->Given('/^I have an calculator$/', function ($world) {
    //implement
});

$steps->When('/^I have entered (\d+) as first number$/', function ($world, $
    //implement
});

$steps->And('/^I have entered (\d+) as second number$/', function ($world, $
    //implement
});

$steps->And('/^I press \'([^\']*)\'$/', function ($world, $arg1) {
    //implement
});

$steps->Then('/^The result should be (\d+)$/', function ($world, $arg1) {
    //implement
});
```

```php
class Calc {
    protected $first = 0;
    protected $second = 0;
    protected $result = 0;

    public function setFirst($num) {
        $this->first = $num;
    }

    public function setSecond($num) {
        $this->second = $num;
    }

    public function add() {
        $this->result = $this->first + $this->second;
    }

    public function getResult() {
        return $this->result;
    }
}
```

Screenshots / CSS regression

Pixel-by-pixel comparison

Similarity in %

`imagick::compareImages()`
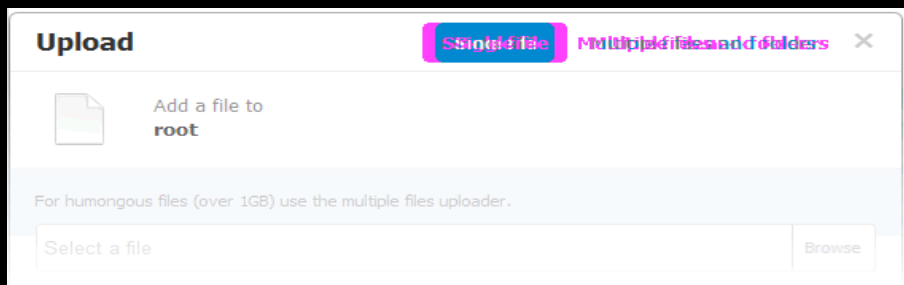
Huddle/PhantomCSS

LEVEL 2 GRAY BOX

Environment diagnostics

- RAM / HDD / file permissions

- Packages, extensions, libraries

- Services / docker containers

```php
class MongoIOTest extends \PHPUnit_Framework_TestCase {

    function VoidConnection() {
        $this->setExpectedException('MongoConnectionException');
        new \MongoClient("mongodb://10.10.10.10:27018");
    }

    function Connection() {
        $m           = new \MongoClient("mongodb://10.10.10.10:27017");
        $db          = $m->selectDB("Gratheon");
        $collection = new MongoCollection($db, 'user_log');

        $this->assertTrue($collection->count() >= 0);
    }
}
```

```php
trait EmailServerDeliveryTest {

    protected $email_test_email = 'noreply@fake.gratheon.com';

    public function receiveEmail() {
        $parser = new PlancakeEmailParser(file_get_contents('/tmp/test-mail'));

        $result = [
            'title' => $parser->getSubject(),
            'body'  => $parser->getPlainBody()
        ];
        $this->emptyMailFile();

        return $result;
    }

    protected function emptyMailbox() {
        $fp = fopen('/tmp/test-mail', 'w');
        fwrite($fp, '');
        fclose($fp);
    }
}
```

Integration external APIs

- Availability - FTP / REST API
  - server is down
  - no file / API 404

- Compatibility
  - empty file
  - wrong json/XML schema

guzzle/guzzle
Codeception/Codeception

```php
class IntegrationTestBase extends \PHPUnit_Framework_TestCase {
    protected $phpErrorDetection = 'error';
    protected $baseURL = 'https://local/';
    protected $cookieFile = '';


    protected function setUp() {...}


    /** @param      $url ...*/
    protected function curlPost($url, $data, $useCookie = true) {...}


    protected function curlPostRawJson($url, $data) {...}


    /** @param string $url ...*/
    protected function curlFileUpload($url, $filePath, $postFilename, $post = [], $useCook


    /** @param      $url ...*/
    protected function curlGET($url, $useCookie = true) {
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_SSLVERSION, 3);
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
```

Single Page Application

Frontend / UI

API

Backend / DB
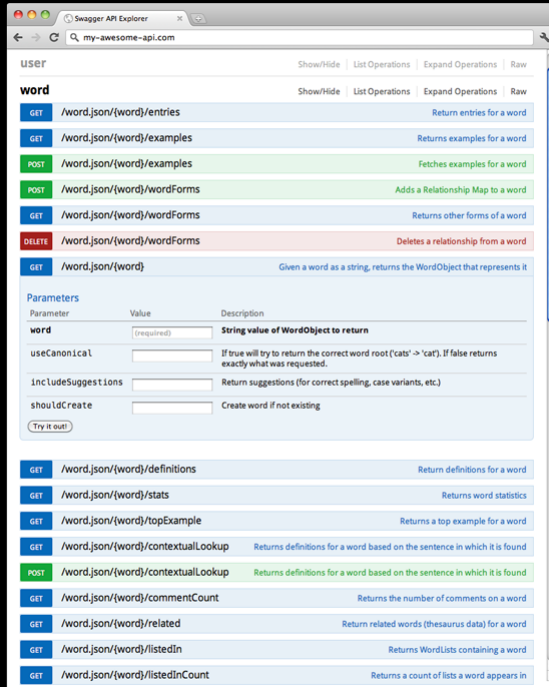
Own internal API

- HTTP GET on all endpoints
  No "Stack trace", "error", "exception" in response

- POST/PUT for data change
  - Mobile app scenarios
  - UDP + binary data
  - Image upload

guzzle/guzzle
Codeception/Codeception

Why you need to test with real DB

- DB specifics – dates / autoincrements
- Stored procedures
- Transactions
- Handling large data
- ORM mapping

Internal API + separate DB

- DB isolation for each test
- Slow & SQL fixtures harder to maintain
- Alternatives
  - transaction & rollback
  - Virtual machine
  - raw db file copy

```php
class AuthEndpointIsolatedDataTest extends IsolatedDataIntegrationTestBase {
    /** @test ...*/
    function postRestoreMultipleTimes() {
        $this->prepareAccountUser();

        for ($i = 1; $i <= 6; $i++) {
            $result = $this->curlPOST( $this->baseURL . 'auth/Restore', ['email' => 'pilot@gratheon.com'| );
        }
        $this->assertNoPHPErrors($result);
        $json = json_decode($result, true);
        $this->assertEquals(false, $json['success'], print_r($result, true));

        $this->db->execute("UPDATE user_restoration SET added = DATE_SUB(added, INTERVAL 20 MINUTE); ");
        $result = $this->curlPOST($this->baseURL . 'auth/Restore',['email' => 'pilot@gratheon.com'|);
        $this->assertNoPHPErrors($result);
        $json = json_decode($result, true);
        $this->assertEquals(false, $json['success'], print_r($result, true));

        $this->db->execute("UPDATE user_restoration SET added = DATE_SUB(added, INTERVAL 10 MINUTE); ");
        $result = $this->curlPOST($this->baseURL . 'auth/Restore',['email' => 'pilot@gratheon.com']);
        $this->assertNoPHPErrors($result);
        $json = json_decode($result, true);
        $this->assertEquals(true, $json['success'], print_r($result, true));
}
```

Security + fuzzing tests

Bruteforce own API

- Injections - sql, xss
- Argument types
  - bool, array, string, json
- Binary/Images
  - Different types, sizes

drodil/PHP-Fuzz-tester

```
trait ArgumentVariatorControl{
    private $injectionTypeTokens = [
        'null',
        '',
        'longstringlongstringlongstringlongstringlongstrin
        0,
        123,
        12.30,
        '0b10100111001',
        '+0123.45e50',
        'true',
        'false',
        '[]',//json array
        '{}',//json array
    ];
```

```php
protected function checkEndpointForErrorsOnVariousGET($URL, $fields = ['id']){
    foreach ($this->injectionTypeTokens as $injection) {
        foreach($fields as $k){
            $map[$k]=$injection;
        }


        $query = http_build_query($map);


        $result = $this->curlGET($URL.'?'.$query);


        $this->assertNotContains('SQLSTATE', $result);
        $this->assertNotContains('error', $result);
        $this->assertNotContains('exception', $result);
    }
}
```

```php
$orderBy = true;
// note in_array is not in strict mode
if (!in_array($orderBy, ['createdAt', 'name'])) {
    return false;
}


//will fail and spill out what SQL was executed
$db->execute("SELECT * FROM `articles` ORDER BY :o",['o'=>$orderBy]);
```

```php
trait InjectionControl {
    private $injectionAttackTokens = [
        '1 OR 1=1', "1' OR '1'='1", "1'1", "1 EXEC SP_ (or EXEC XP_
        "1' AND 1=(SELECT COUNT(*) FROM users); --", "1 AND USER_N
        "\'; DESC users; --", "1\'1", "1' AND non_existant_table =
        "' OR username IS NOT NULL OR username = '", "' or 1=1#",
        "' or 1=1/* (MySQL < 5.1)", "' or 1=1;%00", "' or 1=1 union
        "1='1", "' or--newline", "+ - ~ !", "' or —+2=--!!!'2",
        "%20 %09 %0a %0b %0c %0d %a0 /**/", '1" OR 1=1',
        '<script>alert(1337);</script>'
    ];
```

```php
protected function checkPingInjections($saveURL, $fields = ['login', 'pas

    foreach ($this->injectionAttackTokens as $injection) {
        $data = [];

        $this->resetCookie();
        $this->curlPost($this->baseURL . 'auth/login', []);

        list($data, $result, $resultJSON) = $this->postFieldsToURL(
            $saveURL, $fields,
            $injection, $data,
            true
        );

        $comparisonFunction($injection, $result);
    }
}
```

Performance tests - backend

- Response time < 1s
- DB queries < 15
- Memory load < 10MB
- Is the app getting slower?

jokkedk/webgrind

Data integrity/sync tests

- Check data in N services
  - files in Mysql, mongo, AWS
- JSON schema, NULLs, Dates

Example – translations export-import in UTF16 CSV

Application migration/upgrade tests

- If you run delta scripts / migrations (like phinx)
- Migration script = unit tests
- Data integrity after run

Configuration matrix tests

If you support multiple environment combinations (CMS)

- Browsers
- PHP versions
- Servers
- OS

Virtualize! – docker, vagrant

LEVEL 3 WHITE BOX

Unit tests

- Limited to isolated function or class
- Data processing confirmation
  - math / logic / price calculators
  - regexp / parsers / validators
  - random() ratio

sing()

Unit tests of private methods

"you should not test privates, its a bad practice"

- Abstract class
- Public efficiency / cycles
- Debug
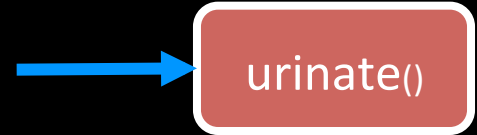
urinate()

```
new \ReflectionClass('Man')->getMethod('urinate')
        ->setAccessible(true)
```

Unit tests calling private methods

"(In phpspec) you cannot mock system under test. This is a bad practice"

runkit::method_redefine

PHPUnit
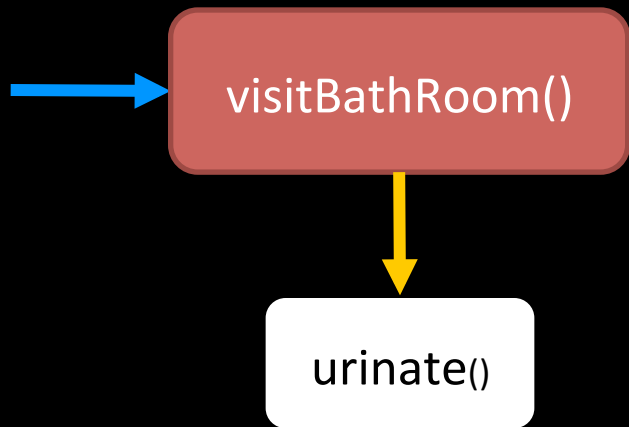
```php
class PhoneTest extends \PHPUnit_Framework_TestCase
    protected $o;

    function setUp() {
        $this->o = new Phone();
    }


    /** @test ...*/
    function validate_mix() {
        $r = $this->o->isValid(" + ().((()))");
        $this->assertEquals(false, $r, $r);
    }


    /** @test ...*/
    function validate_mix2() {
        $r = $this->o->isValid("-- +()()())()))");
        $this->assertEquals(false, $r, $r);
    }
}
```

```php
use PhpSpec\ObjectBehavior;
use Prophecy\Argument;

/**
 * @see |Gratheon\validators\Phone
 */
class PhoneSpec extends ObjectBehavior {
    function it_is_initializable() {
        $this->shouldHaveType('Gratheon\validators\Phone');
    }

    function it_fails_mixed_brackets() {
        $this->isValid(" + ().((()))")->shouldEqual(false);
        $this->isValid("-- +()()()()()))")->shouldEqual(false);
    }
}
```

Unit tests calling other classes

Test doubles:
- Dummy – argument that's not used (during test)
- Stub – hard-coded value
- Spy – did methods got called?
- Fake – dynamic stubs
- Mock – all-in-one

padraic/mockery
phpspec/prophecy
Codeception/AspectMock
sebastianbergmann/phpunit-mock-objects

Mocks

Expectations of coupling dynamics:
- What objects/functions to be called
- Order & timing
- Arguments
- Emulates returns
- Exceptions

padraic/mockery
phpspec/prophecy
Codeception/AspectMock
sebastianbergmann/phpunit-mock-objects

Mocks

sleep() → **man** → useBed() → **home** protect()
warm()
store()
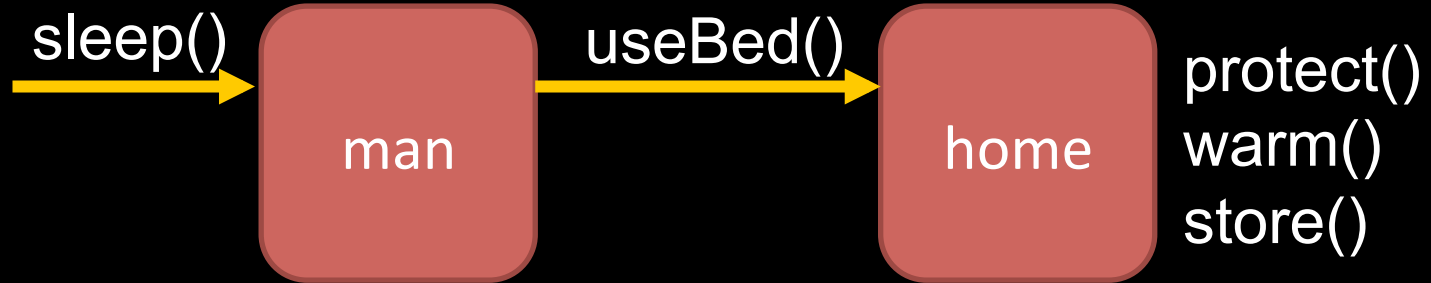
padraic/mockery
phpspec/prophecy
Codeception/AspectMock
sebastianbergmann/phpunit-mock-objects

Mocks – does it call something?
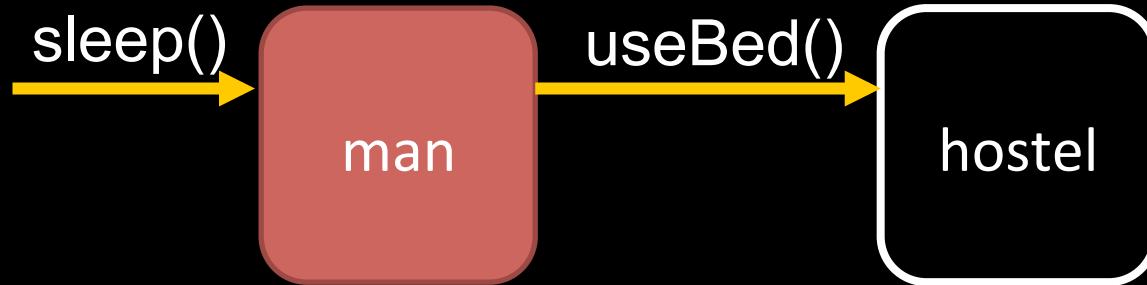
sleep() → man → useBed() → hostel

padraic/mockery
phpspec/prophecy
Codeception/AspectMock
sebastianbergmann/phpunit-mock-objects

```php
ss FileResourceTest extends \PHPUnit_Framework_TestCase {

    private $dbMock;

    public function setUp() {
        $this->dbMock = $this->getMock(
            'Database',
            ['execute', 'insert', 'select', 'delete', 'executeOrGetCached']
        );

        $this->o = new \GratheonCMS\model\FileResource($this->dbMock);
    }

    /**
     * @test
     */
    public function get_none() {
        $this->dbMock->expects($this->at(0))->method('execute')->will($this->returnValue([]

        $r = $this->o->getByID(1);
        $r = json_decode(json_encode($r), true);
```

2

Mocks with magic syntax

PHPUnit
+
prophecy

```php
public function setStatus() {
    $dbProphecy = $this->prophesize('Gratheon\Database');
    $o = new \Gratheon\model\Article($dbProphecy);

    //note how magic methods & chaining is used
    $dbProphecy->execute(
        Argument::type('string'),
        Argument::type('array')
    )->willReturn(true);
    $o->db = $this->dbProphecy->reveal();

    $result = $o->setStatus('live', 4, 3);
    $this->assertEquals(true, $result);
}
```

Mocks chains

```php
$statement = $this->getMock('Statement', ['execute', 'rowCount']);
$pdo       = $this->getMock('PDOMock', ['prepare']);

$statement->expects($this->once())->method('execute')->will($this->returnValue(true));
$statement->expects($this->once())->method('rowCount')->will($this->returnValue(1));

$pdo->expects($this->once())->method('prepare')->with(
    $this->stringContains(
        'UPDATE `user` SET `name`=:name,`updatedAt`=NOW() ' .
        'WHERE `id`=:id AND `email`=:email'
    )
)->will($this->returnValue($statement));

$o = new Database($pdo);
$r = $o->update('user',
    ['name' => 'asd', 'updatedAt' => Database::NOW],
    ['id' => 5, 'email' => 'ads@asd']
);

$this->assertEquals(1, $r);
```
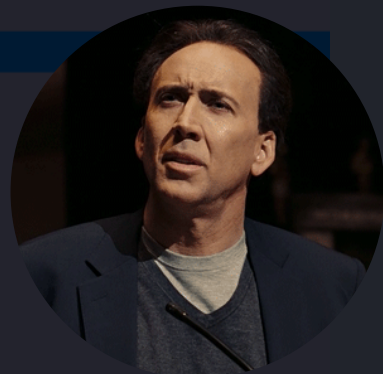
Mock magic chains

```php
class DatabaseSpec extends ObjectBehavior {
    /**
     * @param |PDO          $pdo
     * @param |PDOStatement $statement
     */
    function it_should_update_table_row($pdo, $statement) {
        $this->beConstructedWith($pdo);

        $statement->execute(Argument::any())->willReturn(true);
        $statement->rowCount()->willReturn(1);

        $pdo->prepare(
            'UPDATE `user` SET `name`=:name,`updatedAt`=NOW() '.
            'WHERE `id`=:id AND `email`=:email'
        )->willReturn($statement);

        $this->update(
            'user',
            ['name' => 'asd', 'updatedAt' => \Gratheon\Database::NOW],
            ['id' => 5, 'email' => 'ads@asd']
        )->shouldEqual(1);
    }
}
```

Mocks – backward thinking

Code

- Update()

- Pdo prepare()

- Statement execute()

- Statement rowCount()

Test

- Statement execute()

- Statement rowCount()

- Pdo prepare()

- Update()

Contracts

Side 1 has expectations

man

sex()

getNewKid()

Perfect wife

sex_anyTime()
sex_noCondom()
getNewKid_hasMyHair()
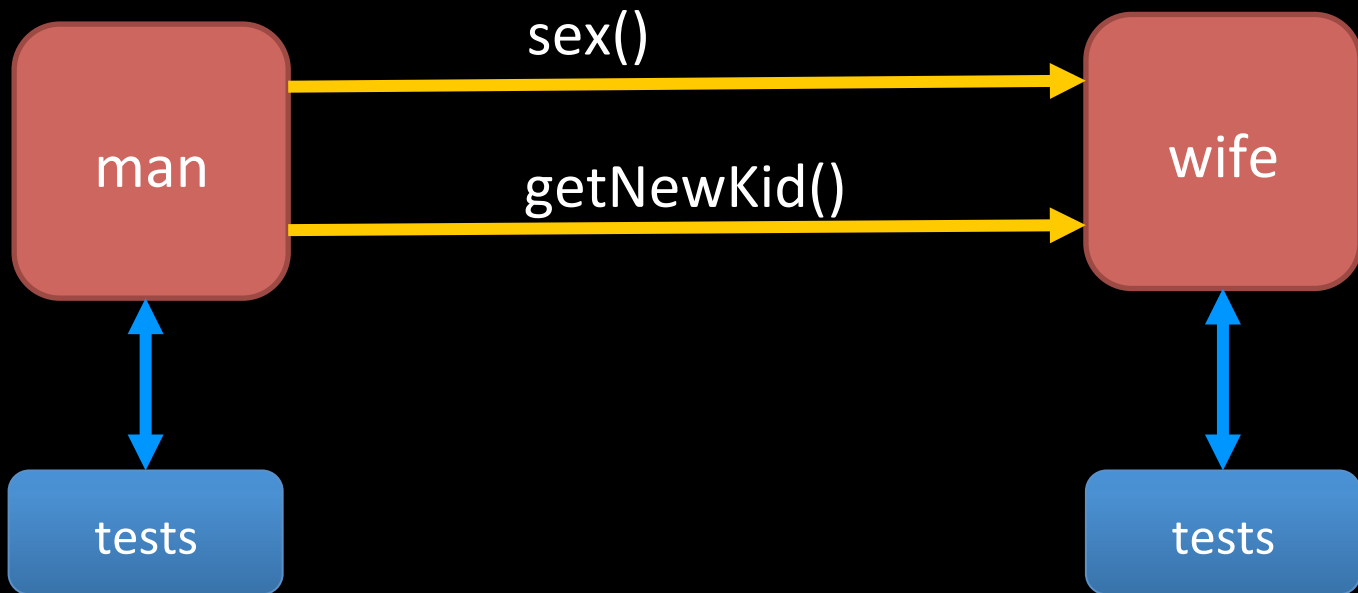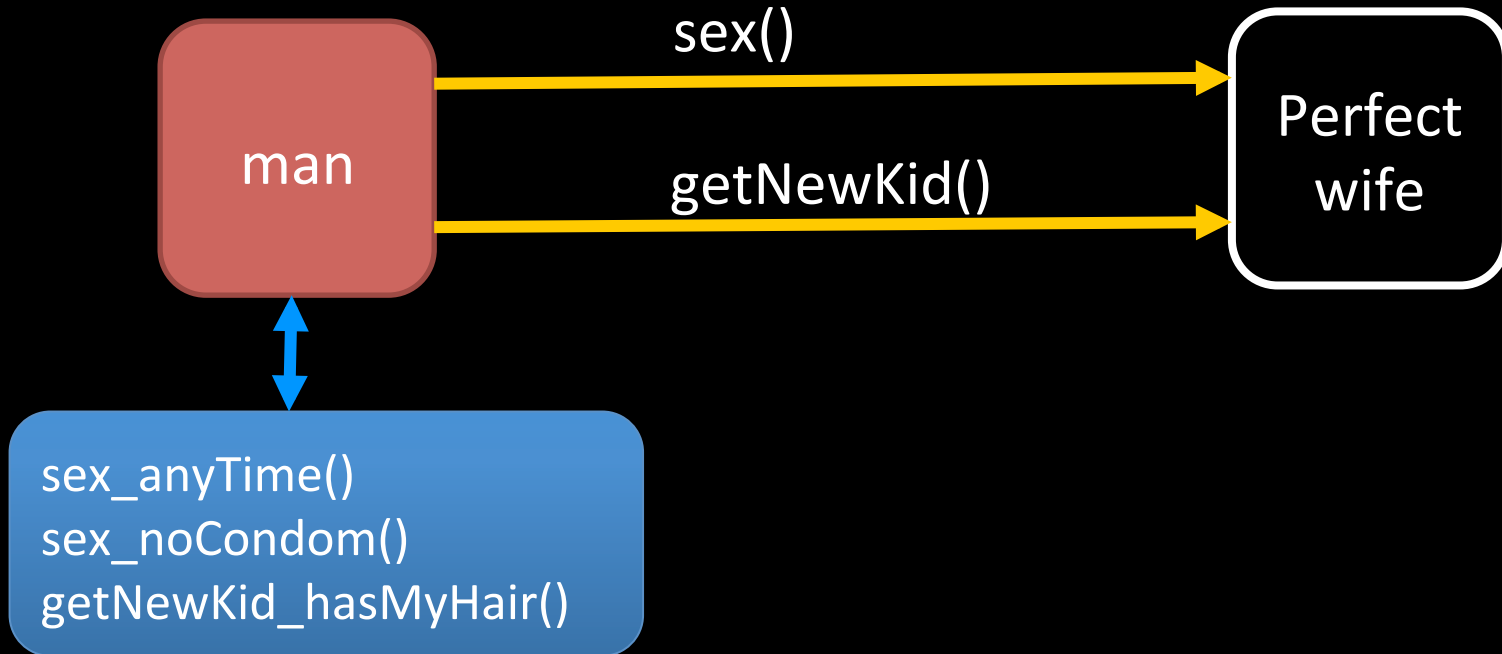
Side 2 has **different** expectations
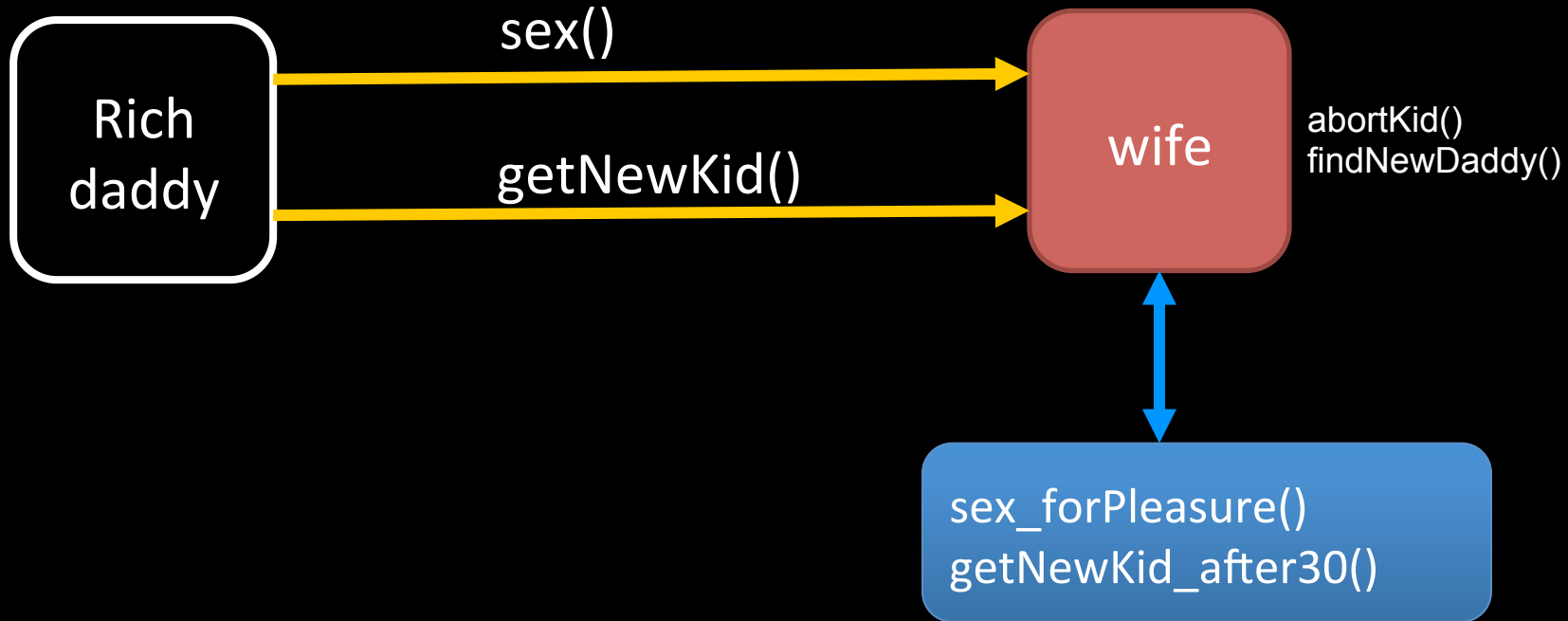
Contracts – a smarter interface

- Input validation
- Method call sequence
- Internal state limitations
- Output expectations
  empty array, false, Null, array, stdClass, Inherited
  entity, id-based map, pagination object…

Contracts – mock matcher
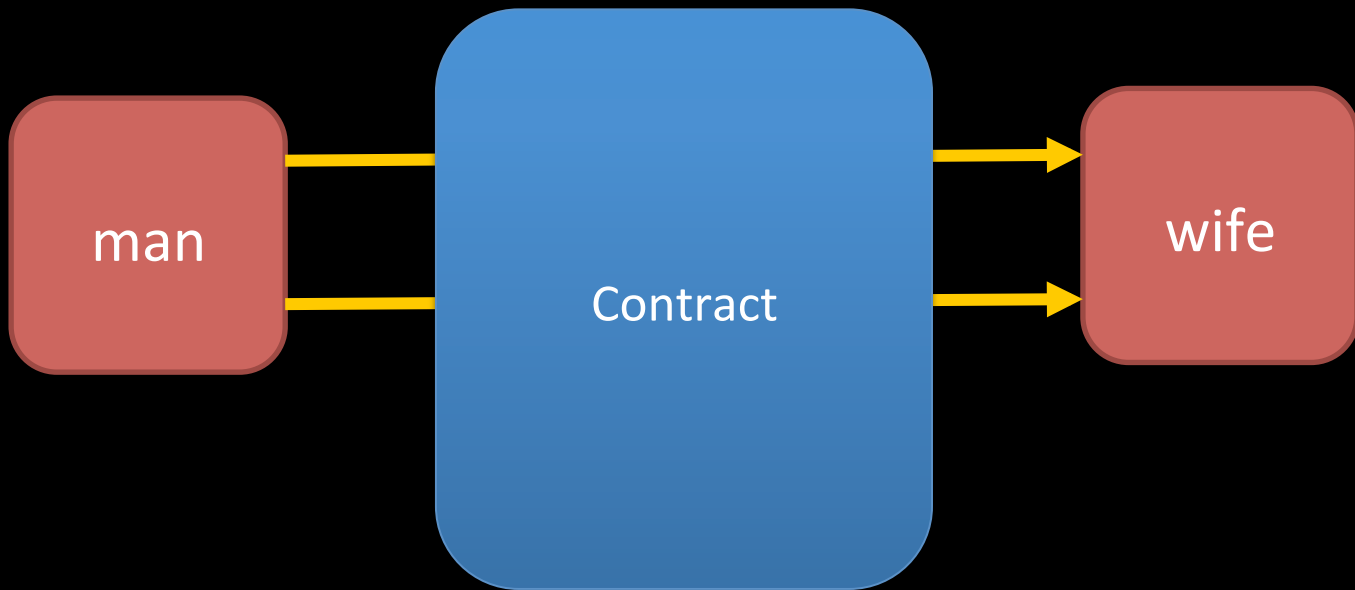
Contract tests

Parties have obligations

• Preconditions – before running method

• Postconditions

• Invariants – limitations for the entire class

lisachenko/php-deal
stuartherbert/ContractLib

```php
function foo($string) {
    //we must tell ContractLib to enforce contracts
    Contract::EnforceWrappedContracts();

    Contract::Preconditions(function() use ($string) {
        Contract::RequiresValue($string, is_string($string), '$string must be a string');
        Contract::RequiresValue($string, !empty($string), '$string must not be empty');
    });

    echo "this will not run\n";
}
```

```php
use PhpDeal\Annotation as Contract;

/**
 * Simple trade account contract
 */
interface AccountContract
{
    /**
     * Deposits fixed amount of money to the account
     *
     * @param float $amount
     *
     * @Contract\Verify("$amount>0 && is_numeric($amount)")
     * @Contract\Ensure("$this->balance == $__old->balance+$amount")
     */
    public function deposit($amount);

    /**
     * Returns current balance
     *
     * @Contract\Ensure("$__result == $this->balance")
     *
     * @return float
     */
    public function getBalance();
}
```

In-memory filesystem tests

- built-in streaming function support
  - file_put_contents, fwrite
- permissions & viewer

- large file support

- result structure assertion
- relative paths are not supported
  - chdir(), realpath() too

mikey179/vfsStream

```php
class BucketTest extends PHPUnit_Framework_TestCase {

    private $rootDir;

    public function setUp() {
        $this->o = new Bucket('/tmpLibrary');

        vfsStream::setup();
        vfsStreamWrapper::register();
        vfsStreamWrapper::setRoot(vfsStream::newDirectory('root', 0755));
        $this->rootDir = vfsStreamWrapper::getRoot();

        $this->o->pathRewrite = function ($path) {
            return vfsStream::url('root/' . $path);
        };
    }

    /** @test ... */
    public function get() {
        vfsStream::create(
            ['tmpLibrary' => ['bb2075d7d7023ebd5929f6a3f4c4d499' => ['original.jpg' => '42']]],
            $this->rootDir
        );

        $r = $this->o->get('/bb2075d7d7023ebd5929f6a3f4c4d499/original.jpg');
        $e = '42';
        $this->assertEquals($e, $r, $r);
    }
}
```

In-memory database tests

Function call expected, not entire stack. Fast, but narrow

PHPUnit_Extensions_Database_TestCase - too complex
- Data sources: XML, CSV, mysql dump
- DB virtualization: PDO + SQLite

See also: H2 Database, HyperSQL, Apache Derby

sebastianbergmann/dbunit
liip/LiipFunctionalTestBundle

```php
class ArticleDBTest extends PHPUnit_Framework_TestCase {
    protected $db;

    protected function setUp() {
        $this->db = new PDO('sqlite::memory:');
        $this->db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        $this->db->exec('CREATE TABLE article (title VARCHAR(100))');

        $this->o = new ArticleModel($this->db);
    }

    public function testRowCountForTableWithTwoRowsReturnsTwo() {
        $this->db->exec('INSERT INTO article (title) VALUES (\'asd\')');
        $this->db->exec('INSERT INTO article (title) VALUES (\'xcv\')');

        $this->assertEquals(2, $this->o->getArticleCount());
    }
}
```

Component testing

Component / library - coherent, replaceable utility classes

For example:
- Form builder (please don't)
- Validator
- Logger
- Tree builder
- Image resizer
- File storage abstraction
- CSV/XML generator/parser
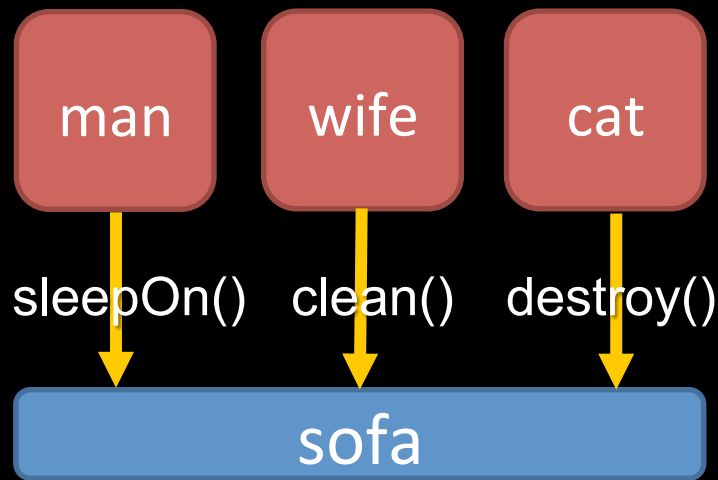
man	wife	cat
sleepOn()	clean()	destroy()

sofa

TEST DESIGN

Speed - order by types

PHPUNIT.xml

- Unit tests
  - General project unit tests (validators, DB adapter)
  - Unit tests per component (models)
- Integration tests
  - External
  - Internal API
  - Internal with DB isolation
  - Performance + Security
  - Data integrity
- UI tests
  - Screenshotting

Time
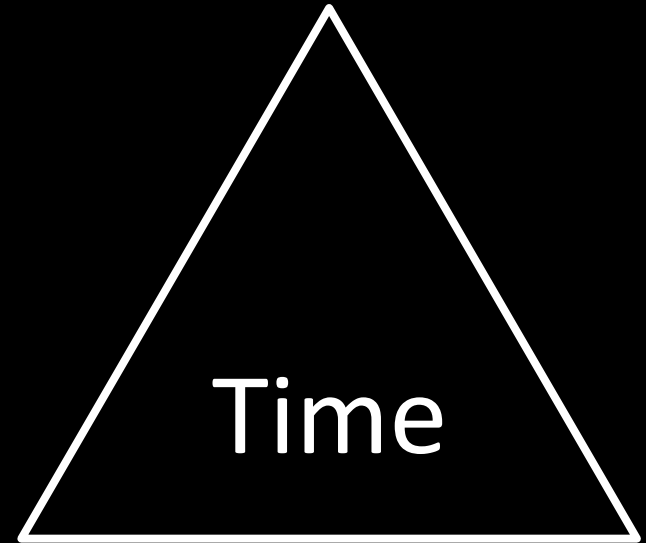
Modularity

Divide your app into modules - see what tests you lack where

/**controllers**/article.php
/**models**/article.php
/**tests**/integration/articleControllerTest.php
/**tests**/unit/articleModelTest.php

/article/**controller**/
/article/**model**/
/article/**modelTests**/
/article/**controllerTests**/

Annotations

jump between function and test
@see ArticleController::postCreate

@ticket T-1000 + JIRA API
(PHPUnit supports)

@group production
@group failed
phpunit --testsuite unit

Type Declaration ⇧⌘B
Super Method ⌘U
Test ⇧⌘T
Related Symbol... ⌃⌘↑

MyApp / MY-389
Add article sql-injection tests for backend

Edit   Comment   Assign   More ▾   Close Issue   Reopen Issue   Admin ▾

Details

Type:	↗ Improvement	Status:	RESOLVED (View Workflow)
Priority:	↑ Major	Resolution:	Fixed
Component/s:	Backend	Fix Version/s:	None
Labels:	backend security tests		
Sprint:	November end 2014		
Tests:	⌄ ok — ArticleEndpointTest::postAdd_SQLInjections		
	ok — ArticleEndpointTest::postSaveSettings_SQLInjections		
	ok — ArticleEndpointSecurityTest::postAdd_SQLInjections		
	ok — ArticleEndpointSecurityTest::postAddAccepted_SQLInjections		
	ok — ArticleVersionEndpointSecurityTest::postAdd_SQLInjections		
	ok — ArticleVersionEndpointSecurityTest::postAddAccepted_SQLInjections		
	ok — ArticleEndpointSecurityTest::postSaveSettings_SQLInjections		

More than code coverage

Code coverage = you tried to test

Delta coverage – what test adds

Mutations help to find

- tests that lack enough asserts
- fragile code

	Lines		
Total		69.56%	994 / 1429
📁 article		70.67%	212 / 300
📁 news		67.98%	138 / 203
📁 slides		77.30%	109 / 141
📁 video		61.77%	244 / 395
📁 maps		100.00%	45 / 45
📁 images		55.47%	71 / 128
📁 file		78.24%	151 / 193
📁 users		100.00%	24 / 24

henrikbjorn/PhpSpecCodeCoverageExtension

```
   __    __                __
  / /   / /_  _____ ___  / /_  __  _____
 / /___/ __ \/ / / / __ `__ \/ __ \/ / / / __ `/
/_____/_/ /_/\__,_/_/ /_/ /_/_.___/\__,_/\__, /
                                        /____/
```

Humbug version 1.0-dev

Humbug running test suite to generate logs and code coverage data...

     0 [>--] 23 secs

Humbug has completed the initial test run successfully.
Tests: 0 Line Coverage: 36.72%

Humbug is analysing source files...

Mutation Testing is commencing on 101 files...
(.: killed, M: escaped, S: uncovered, E: fatal error, T: timed out)

```
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS |   60 (  1/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS |  120 (  8/101)
SSSSSSSSSSSSSSSSSSMMSSSSSSSSSSSSSSSMMSMSMSSMMSMMSSMSSSS |  180 ( 14/101)
SSSMSMSMSSSSSSSSMSMS SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS |  240 ( 16/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS |  540 ( 33/101)
MSMMSSSMMMMMMMMSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS |  600 ( 36/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS |  660 ( 37/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS |  720 ( 40/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSMMSSSSSSSMSMMSS |  780 ( 49/101)
SSSSSSSSSSSSSSSSSMSSMSSSSMSSMSSSSSSSSSSSSSSSSSSSSSS |  840 ( 54/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS |  900 ( 54/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS |  960 ( 55/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS | 1020 ( 55/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSM | 1080 ( 58/101)
MMMMSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSM | 1140 ( 68/101)
MMSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS | 1200 ( 70/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS | 1260 ( 71/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSMMSMSSSMMSSSSSMMMMMM | 1320 ( 81/101)
MMSSSSSSSSSMSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS | 1380 ( 82/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS | 1440 ( 84/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS | 1500 ( 84/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS | 1560 ( 87/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS | 1620 ( 98/101)
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
```

1676 mutations were generated:
       0 mutants were killed
    1595 mutants were not covered by tests
      81 covered mutants were not detected
       0 fatal errors were encountered
       0 time outs were encountered

Metrics:
    Mutation Score Indicator (MSI): 0%
    Mutation Code Coverage: 5%
    Covered Code MSI: 0%

Remember that some mutants will inevitably be harmless (i.e. false positives).

Time: 2.01 minutes Memory: 15.25MB
Humbug results are being logged as JSON to: humbuglog.json
Humbug results are being logged as TEXT to: humbuglog.txt

```
27) \Humbug\Mutator\Boolean\TrueValue
Diff on \Gratheon\model\Article::getByName() in /Users/artjom/Php
--- Original
+++ New
@@ @@
        SELECT * FROM view_article WHERE name = :name LIMIT 1;',
-       ['name' => $name], true
+       ['name' => $name], false
    );

    if ($row) {
        return new \Gratheon\entity\Article($row);
    }
    else {
```
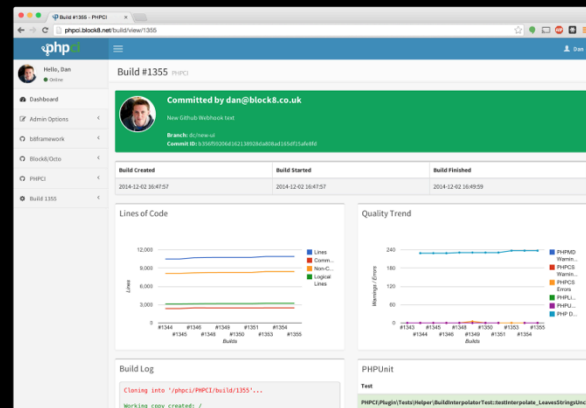
What else to test?

- Frontend
- Stress / Load
- SQL procedures
- Accessibility tests
- Globalization

Track your mistakes
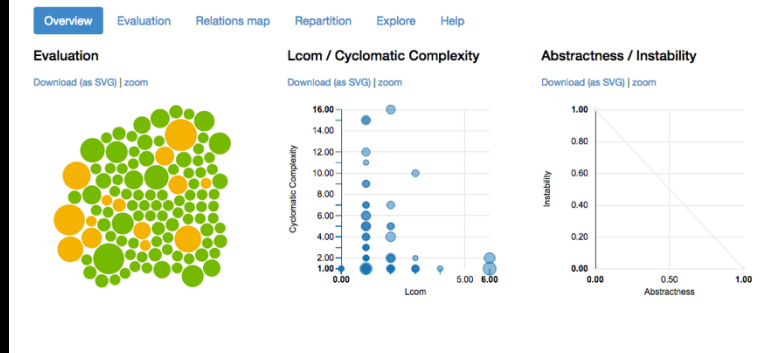
- Monitoring tests failures
- Coverage trend
- Test quality

Monitoring code health:

- Complexity
- Code standard compliance

Block8/PHPCI
Halleck45/PhpMetrics
squizlabs/PHP_CodeSniffer

Awareness

Clarity

Tests

Architecture

QUESTIONS?