



# Testing

Vijaya Chandran Mani

@vijaycs85

# Overview

- Introduction
- Types
- Sample tests
- Lessons learned

# 1. Introduction

# What?

- Code to test code by coder for coder
- Lives and runs as part of project code base
- Quality assurance at developer level

# Why?

- SDLC - Requirements are not explicit
- Edge cases are \*\*\*\*
- Express understanding of system
- Developer Documentation
- Easy to write
- Cost effective (DEV -> QA -> PROD -> Bug -> Repeat)
- Impossible to do CI / CD
- Security updates
- Software version upgrade

## **2. Types of testing**

1. Unit

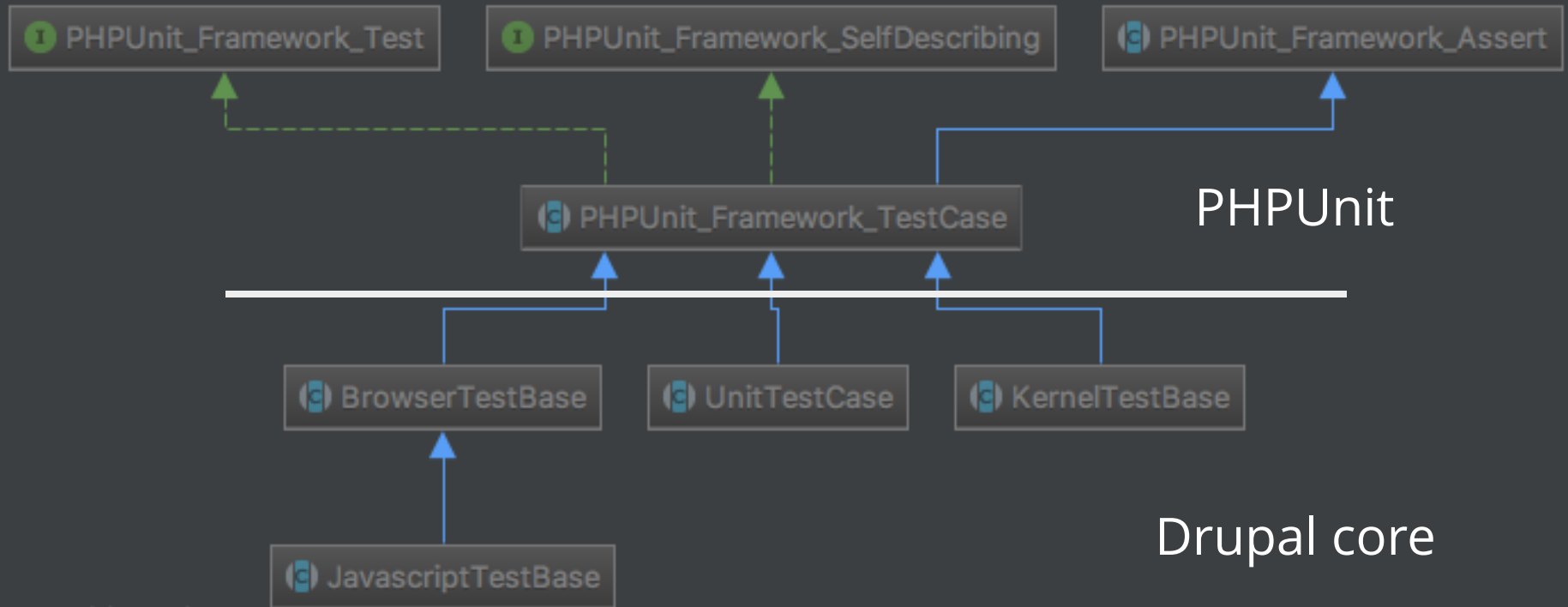
2. Kernel

3. Functional

4. Functional Javascript

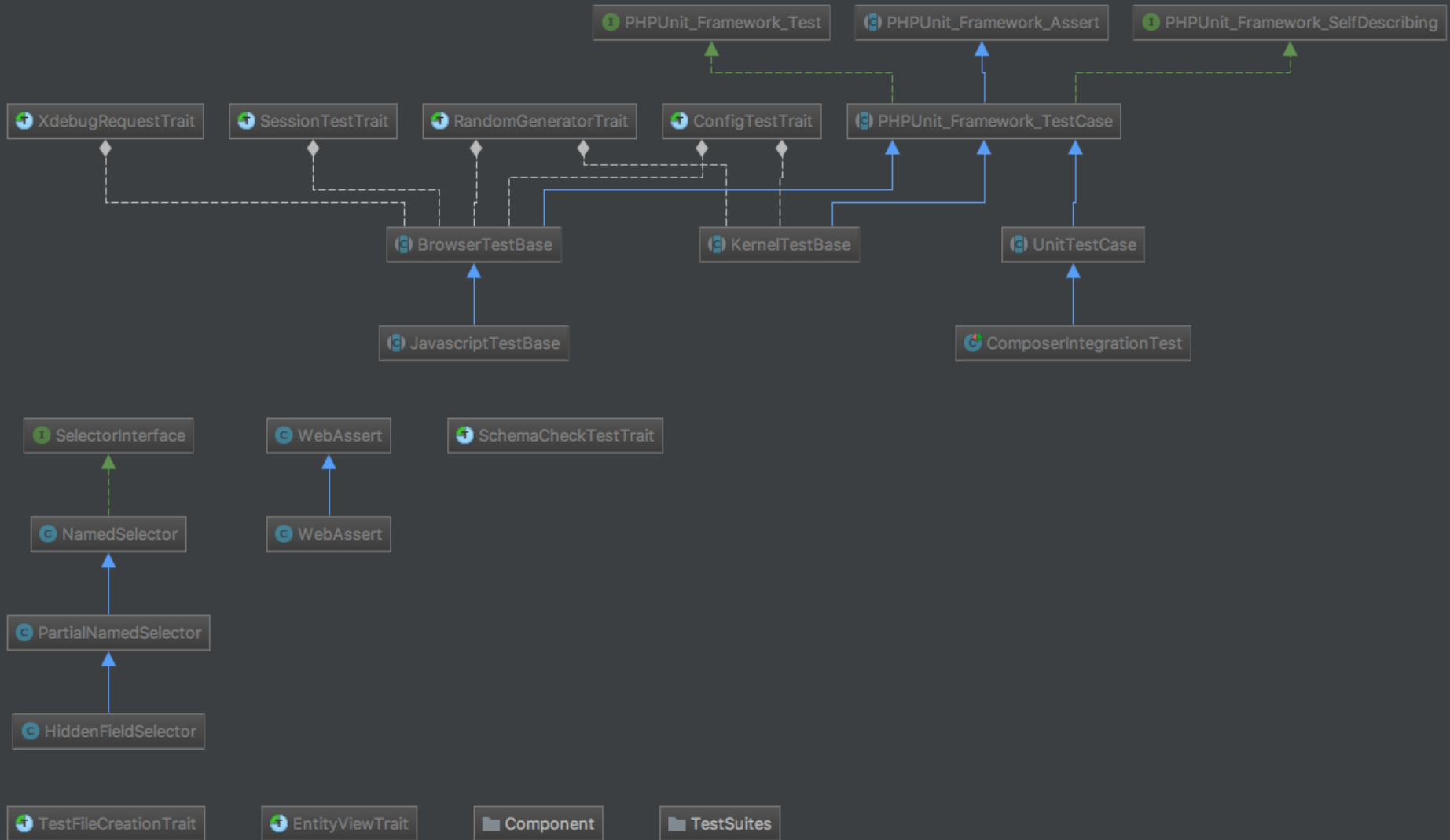
5. Behat

# Structure

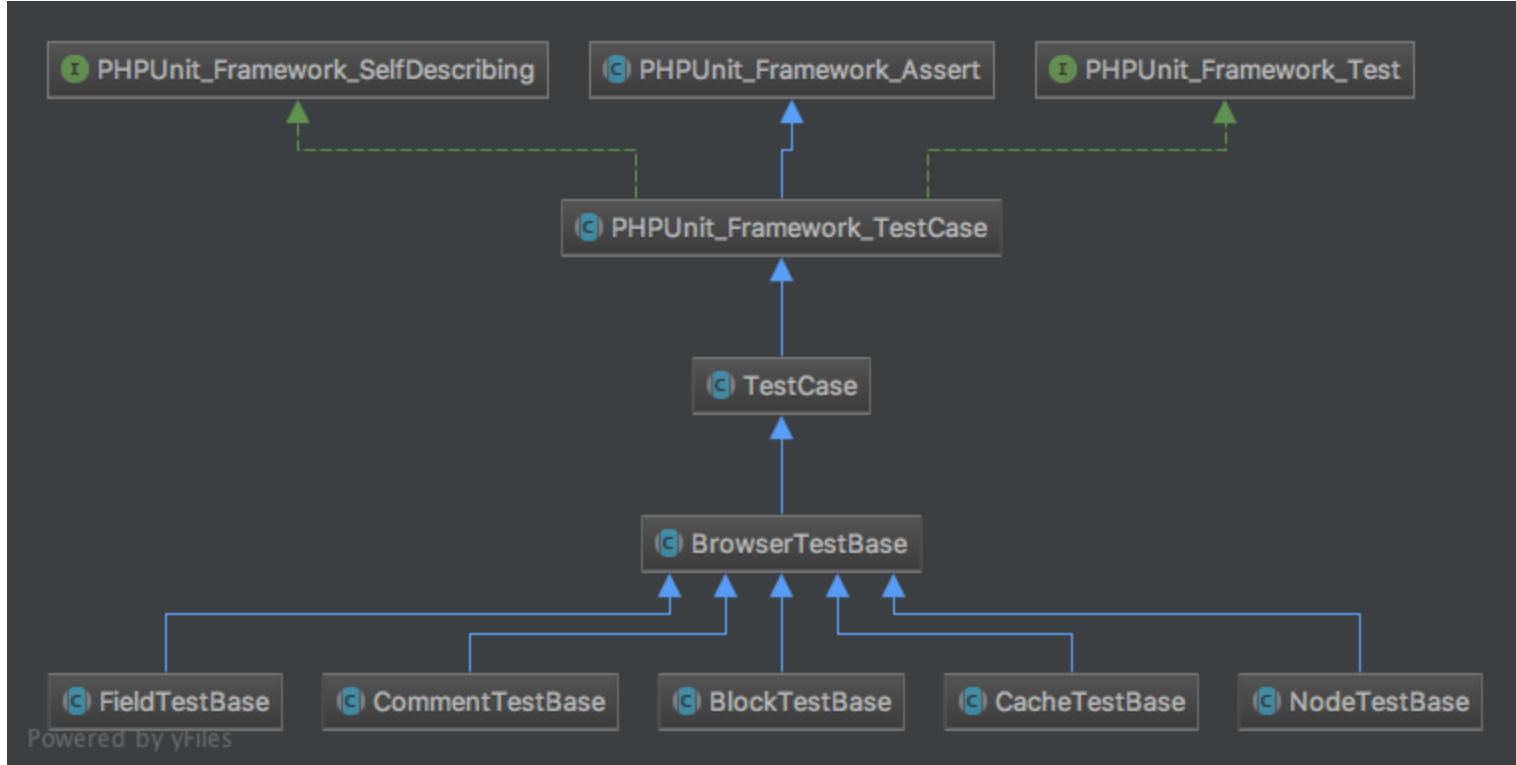




# Structure (expanded)

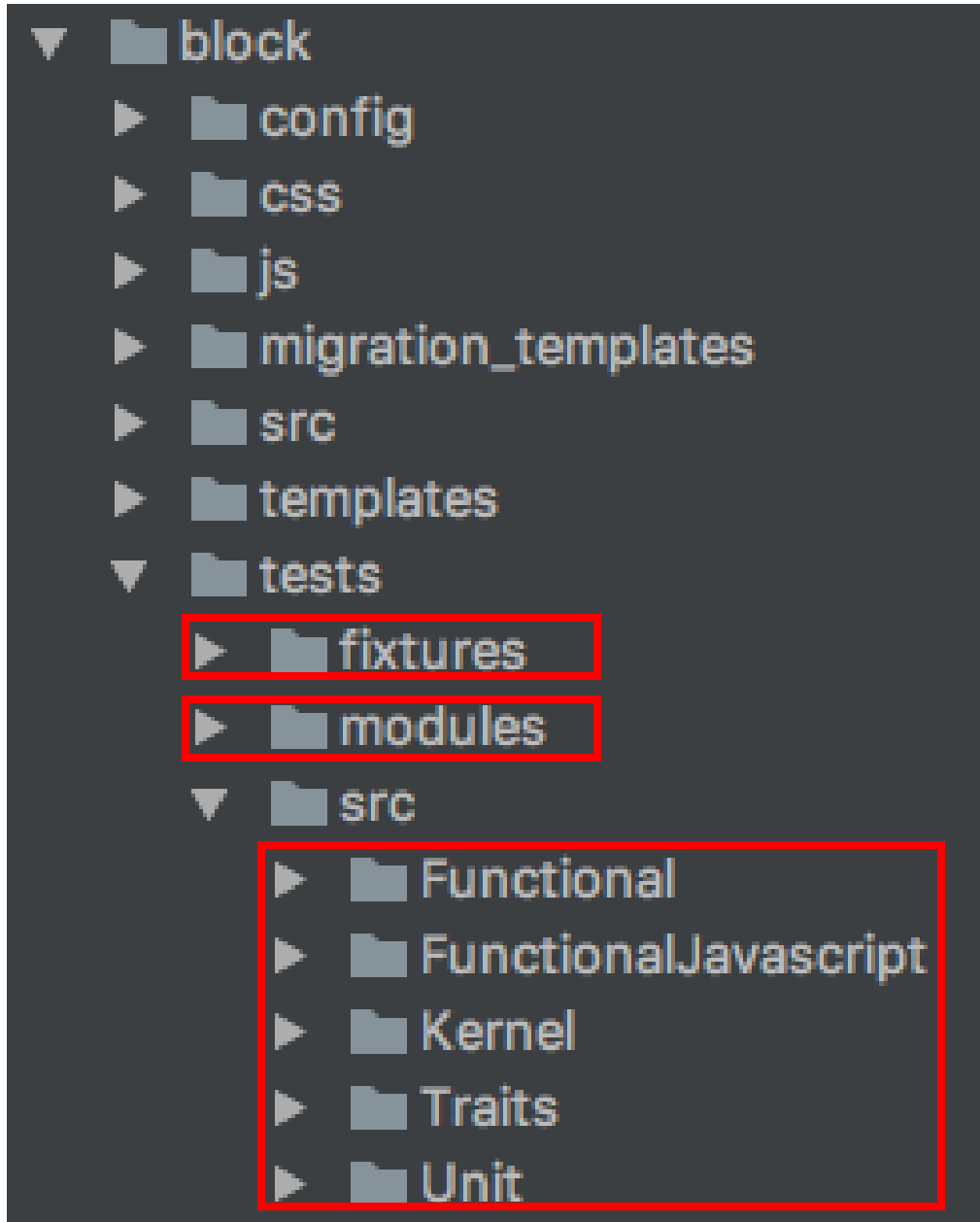


# Functional Test Base



Full list of base class is available [here](#)

# Folder Structure



**Use case**

# Contact module

The image displays three screenshots of the Drupal 8 Contact module interface:

- Left Screenshot:** Shows the 'Contact forms' administration page. It includes a breadcrumb trail 'Home » Administration » Structure' and a '+ Add contact form' button. A table lists existing forms:

FORM	RECIPIENTS	SELECTED	OPERATIONS
Personal contact form	Selected user	No	Manage fields
Website feedback	admin@example.com	Yes	Edit

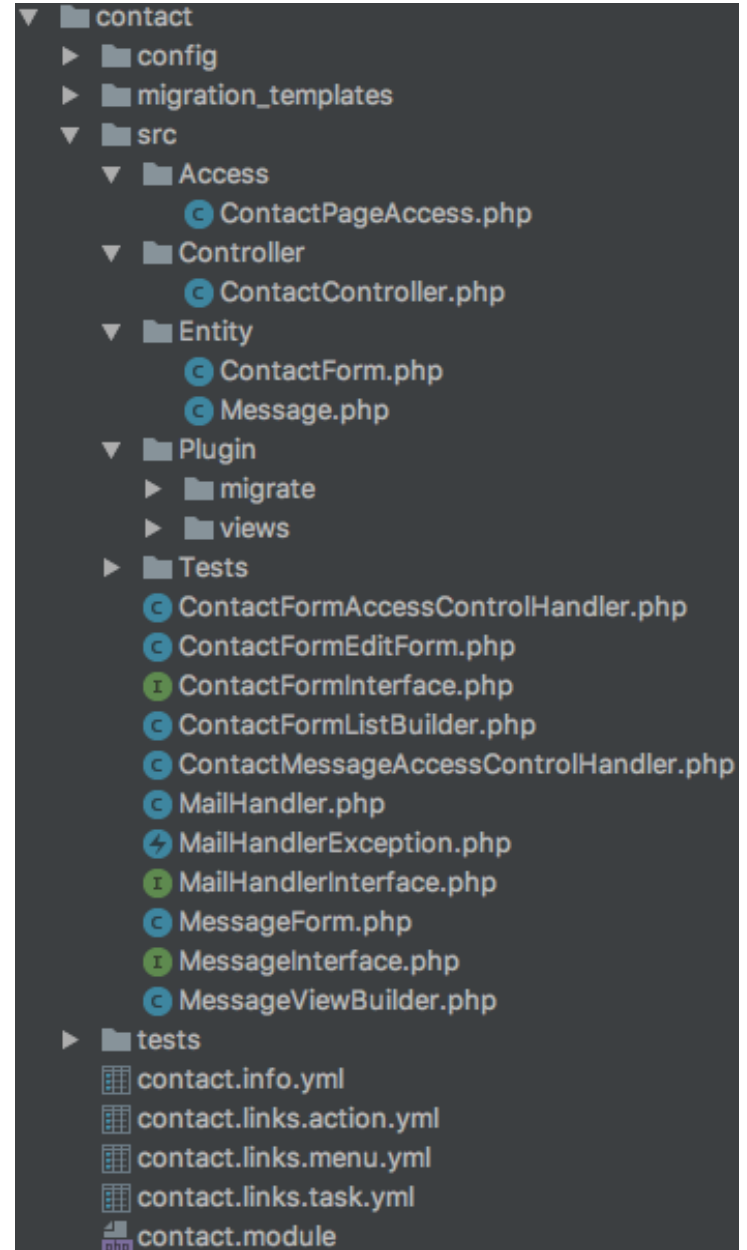
- Middle Screenshot:** Shows the configuration page for the 'Website feedback' form. It includes fields for 'Label', 'Machine name', 'Recipients', 'Message', 'Redirect path', and 'Auto-reply'. The 'Make this the default form' checkbox is checked.

- Right Screenshot:** Shows a preview of the 'Website feedback' form. It includes fields for 'Your name', 'Your email address', 'Subject', and 'Message', along with 'Send message' and 'Preview' buttons.

# Contact

- CRUD contact form
- Compose message
- Send mail

Contact form view == message add



# 2.1 Unit tests

**Drupal\Tests\UnitTestCase**

# Mail handler

```
namespace Drupal\contact;

/**
 * Provides an interface for assembly and dispatch of contact mail messages.
 */
interface MailHandlerInterface {
  /**
   * Sends mail messages as appropriate for a given Message form submission.
   *
   * Can potentially send up to three messages as follows:
   * - To the configured recipient;
   * - Auto-reply to the sender; and
   * - Carbon copy to the sender.
   *
   * @param \Drupal\contact\MessageInterface $message
   *   Submitted message entity.
   * @param \Drupal\Core\Session\AccountInterface $sender
   *   User that submitted the message entity form.
   *
   * @throws \Drupal\contact\MailHandlerException
   *   When unable to determine message recipient.
   */
  public function sendMailMessages(MessageInterface $message, AccountInterface $sender);
}
```

```
// contact.services.yml
```

```
contact.mail_handler:
  class: Drupal\contact\MailHandler
  arguments: ['@plugin.manager.mail', '@language_manager',
             '@logger.channel.contact', '@string_translation', '@entity.manager']
```



# Setup

```
use Drupal\Core\Routing\RouteProvider;

class MailHandlerTest extends UnitTestCase {
    protected function setUp() {
        parent::setUp();
        $this->mailManager = $this->getMock(MailManagerInterface::class);
        $this->languageManager = $this->getMock(LanguageManagerInterface::class);
        $this->logger = $this->getMock(LoggerInterface::class);
        $this->entityManager = $this->getMock(EntityManagerInterface::class);
        $this->userStorage = $this->getMock(EntityStorageInterface::class);
        $this->entityManager->expects($this->any())
            ->method('getStorage')
            ->with('user')
            ->willReturn($this->userStorage);

        $string_translation = $this->getStringTranslationStub();
        $this->contactMailHandler = new MailHandler($this->mailManager,
            $this->languageManager, $this->logger, $string_translation, $this->entityManager);
        $language = new Language(['id' => 'en']);

        $this->languageManager->expects($this->any())
            ->method('getDefaultLanguage')
            ->will($this->returnValue($language));

        $this->languageManager->expects($this->any())
            ->method('getCurrentLanguage')
            ->will($this->returnValue($language));
    }
}
```

# Test 1

```
/**
 * Tests the children() method with an invalid key.
 *
 * @covers ::sendMailMessages
 */
public function testInvalidRecipient() {
    $message = $this->getMock('\Drupal\contact\MessageInterface');
    $message->expects($this->once())
        ->method('isPersonal')
        ->willReturn(TRUE);

    $sender = $this->getMock('\Drupal\Core\Session\AccountInterface');
    $this->userStorage->expects($this->any())
        ->method('load')
        ->willReturn($sender);

    $this->setExpectedException(MailHandlerException::class, 'Unable to determine message recipient');
    $this->contactMailHandler->sendMailMessages($message, $sender);
}
```

# Test 2

```
/**
 * Tests the sendMailMessages method.
 *
 * @dataProvider getSendMailMessages
 *
 * @covers ::sendMailMessages
 */
public function testSendMailMessages(MessageInterface $message, AccountInterface $sender, $results)
{
    $this->logger->expects($this->once())
        ->method('notice');
    $this->mailManager->expects($this->any())
        ->method('mail')
        ->willReturnCallback(
            function($module, $key, $to, $langcode, $params, $from) use (&$results) {
                $result = array_shift($results);
                $this->assertEquals($module, $result['module']);
                $this->assertEquals($key, $result['key']);
                $this->assertEquals($to, $result['to']);
                $this->assertEquals($langcode, $result['langcode']);
                $this->assertArrayEquals($params, $result['params']);
                $this->assertEquals($from, $result['from']);
            });
    $this->userStorage->expects($this->any())
        ->method('load')
        ->willReturn(clone $sender);
    $this->contactMailHandler->sendMailMessages($message, $sender);
}
```

# Test 2 - DataProvider

```
/**
 * Data provider for ::testSendMailMessages.
 */
public function getSendMailMessages() {
    $message = $this->getAnonymousMockMessage($recipients, '');
    $sender = $this->getMockSender();
    $result = [
        'key' => 'page_mail',
        'params' => [
            'contact_message' => $message,
            'sender' => $sender,
            'contact_form' => $message->getContactForm(),
        ],
    ];
    $results[] = $result + $default_result;
    $data[] = [$message, $sender, $results];
    ...
    ...
    return $data;
}
```

# MockBuilder vs Prophesize

```
// Create a route provider stub.
$provider = $this->getMockBuilder(RouteProvider::class)
    ->disableOriginalConstructor()
    ->getMock();

$provider->expects($this->any())
    ->method('getRouteByName')
    ->will($this->returnValueMap($route_name_return_map));

$this->container = new ContainerBuilder();
$this->container->set('router.route_provider', $provider);
```

```
// Create a route provider stub.
$provider = $this->prophesize(RouteProvider::class);

$provider->getRouteByName()->willReturn($route_name_return_map);

$this->container = new ContainerBuilder();
$this->container->set('router.route_provider', $provider->reveal());
```

# Summary

- Method level testing
- Easy to add multiple data set (@dataProvider)
- Fastest test runs.

```
>→ drupal8 git:(8.5.x) php core/scripts/run-tests.sh --verbose --url http://d8 --class "Drupal\Tests\permission_ui\Unit\Menu\PermissionUiLocalTasksTest"

Drupal test run
-----

Tests to be run:
- Drupal\Tests\permission_ui\Unit\Menu\PermissionUiLocalTasksTest

Test run started:
  Tuesday, October 24, 2017 - 15:12

Test summary
-----

Drupal\Tests\permission_ui\Unit\Menu\PermissionUiLocalTasksT  1 passes
Test run duration: 1 sec

Detailed test results
-----

---- Drupal\Tests\permission_ui\Unit\Menu\PermissionUiLocalTasksTest ----

Status  Group  Filename  Line Function
-----
Pass    Other  PermissionUiLocal  0 Drupal\Tests\permission_ui\Unit\Men
```

```
→ drupal8 git:(8.5.x) php core/scripts/run-tests.sh --verbose --url http://d8 --class "Drupal\Tests\Core\Ajax\AjaxCommandsTest"
```

```
Drupal test run
```

```
-----
```

```
Tests to be run:
```

```
- Drupal\Tests\Core\Ajax\AjaxCommandsTest
```

```
Test run started:
```

```
Tuesday, October 24, 2017 - 15:13
```

```
Test summary
```

```
-----
```

```
Drupal\Tests\Core\Ajax\AjaxCommandsTest
```

```
24 passes
```

```
Test run duration: 1 sec
```

```
Detailed test results
```

```
-----
```

```
---- Drupal\Tests\Core\Ajax\AjaxCommandsTest ----
```

Status	Group	Filename	Line	Function
Pass	Other	AjaxCommandsTest.	39	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	53	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	70	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	83	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	100	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	117	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	132	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	152	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	168	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	185	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	202	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	218	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	235	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	249	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	266	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	280	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	295	Drupal\Tests\Core\Ajax\AjaxCommands
Pass	Other	AjaxCommandsTest.	330	Drupal\Tests\Core\Ajax\AjaxCommands

# 2.2 Kernel tests

`Drupal\KernelTests\KernelTestBase`



# Message Entity

```
/**
 * Defines the contact message entity.
 *
 * @EntityType(
 *   id = "contact_message",
 *   label = @Translation("Contact message"),
 *   handlers = {
 *     "access" = "Drupal\contact>ContactMessageAccessControlHandler",
 *     "storage" = "Drupal\Core\Entity\ContentEntityNullStorage",
 *     "view_builder" = "Drupal\contact\MessageViewBuilder",
 *     "form" = {
 *       "default" = "Drupal\contact\MessageForm"
 *     }
 *   },
 *   admin_permission = "administer contact forms",
 *   entity_keys = {
 *     "bundle" = "contact_form",
 *     "uuid" = "uuid",
 *     "langcode" = "langcode"
 *   },
 *   bundle_entity_type = "contact_form",
 *   field_ui_base_route = "entity.contact_form.edit_form",
 * )
 */
class Message extends ContentEntityBase implements MessageInterface {
```

# Setup

```
class MessageEntityTest extends EntityKernelTestBase {  
  
    public static $modules = [  
        'system',  
        'contact',  
        'field',  
        'user',  
        'contact_test',  
    ];  
  
    protected function setUp() {  
        parent::setUp();  
        $this->installConfig(['contact', 'contact_test']);  
    }  
}
```

- installSchema - DB tables
- installConfig - config/install
- installEntitySchema - Storage for entities

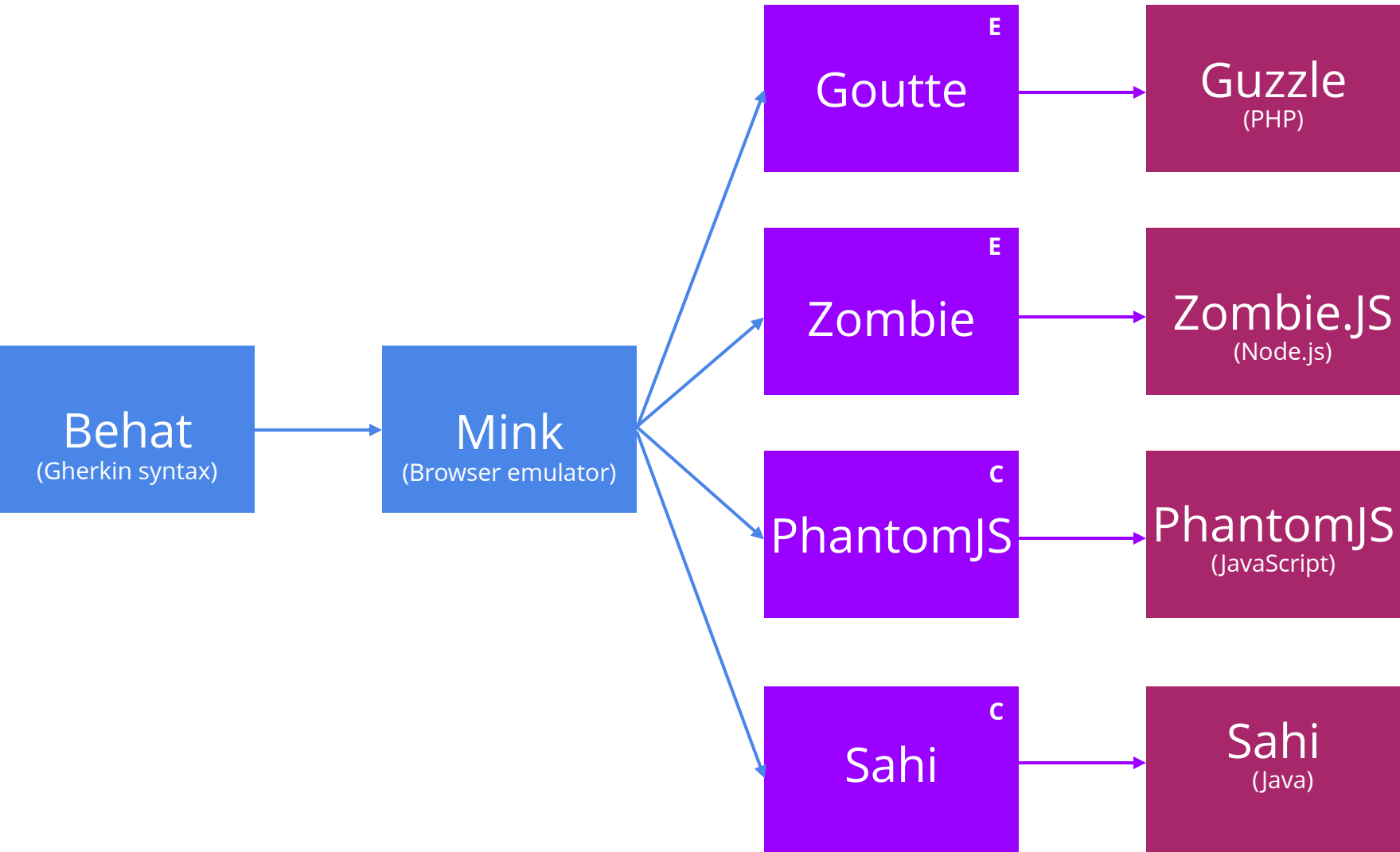
# Test 1

```
public function testMessageMethods() {  
  
    $message_storage = $this->container->get('entity.manager')->getStorage('contact_message');  
    $message = $message_storage->create(['contact_form' => 'feedback']);  
  
    // Check for empty values first.  
    $this->assertEquals($message->getMessage(), '');  
    $this->assertEquals($message->getSenderName(), '');  
  
    // Set some values and check for them afterwards.  
    $message->setMessage('welcome_message');  
    $message->setSenderName('sender_name');  
  
    // Check set values.  
    $this->assertEquals($message->getMessage(), 'welcome_message');  
    $this->assertEquals($message->getSenderName(), 'sender_name');
```

# Summary

- Functional integration testing
- Minimal/on-demand bootstrap
- Prefilled service container & DB
- Much faster compared to Browser test

**Background**

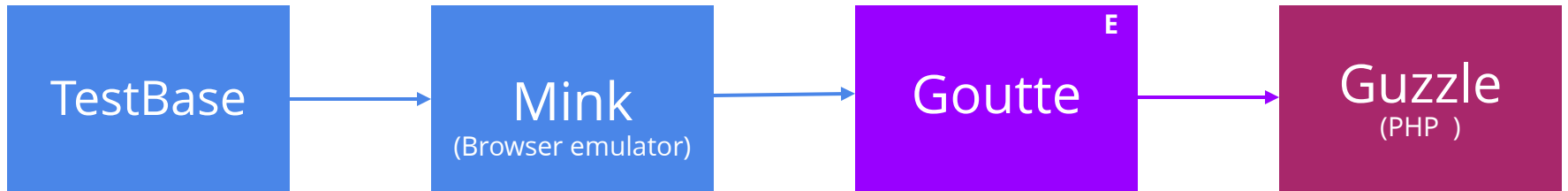


Driver

Ctrl / Emulator

# 2.3 Function Tests

`Drupal\Tests\BrowserTestBase`



# HEADLESS BEHAT!

Drivers

Ctrl / Emulator



# Setup

```
class ContactSitewideTest extends BrowserTestBase {

    use FieldUiTestTrait;
    use AssertMailTrait;

    public static $modules = [
        'text',
        'contact',
        'field_ui',
        'contact_test',
        'block',
        'error_service_test',
        'dblog',
    ];

    protected function setUp() {
        parent::setUp();
        $this->drupalPlaceBlock('system_breadcrumb_block');
        $this->drupalPlaceBlock('local_actions_block');
        $this->drupalPlaceBlock('page_title_block');
    }
}
```

# Test 1

```
public function testSiteWideContact() {
    // Create and log in administrative user.
    $admin_user = $this->drupalCreateUser([
        'access site-wide contact form',
        ...
        ...
    ]);
    $this->drupalLogin($admin_user);

    // Check the presence of expected cache tags.
    $this->drupalGet('contact');
    $this->assertCacheTag('config:contact.settings');

    $edit['contact_default_status'] = TRUE;
    $this->drupalPostForm('admin/config/people/accounts', $edit, t('Save configuration'));

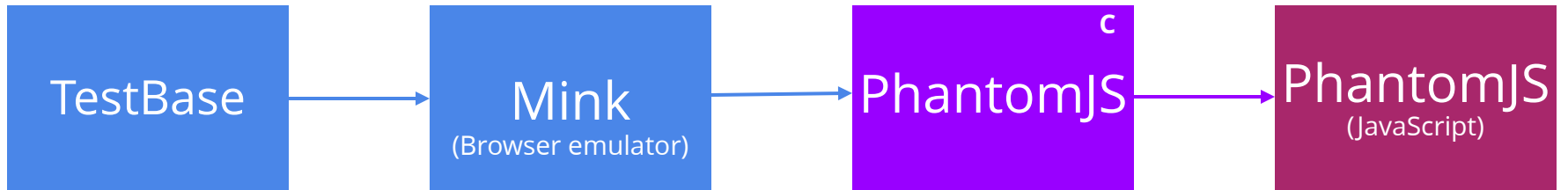
    $this->drupalGet('admin/structure/contact');
    // Default form exists.
    $this->assertLinkByHref('admin/structure/contact/manage/feedback/delete');
```

# Summary

- Run in fresh installation
- Full bootstrap
- Less code to cover most of functional elements

# 2.4 Function Javascript Tests

Drupal\FunctionalJavascriptTests\JavascriptTestBase



Drivers

Ctrl / Emulator

# Setup

```
use Drupal\FunctionalJavascriptTests\JavascriptTestBase;

/**
 * Tests for the machine name field.
 *
 * @group field
 */
class MachineNameTest extends JavascriptTestBase {

    public static $modules = ['node', 'form_test'];

    protected function setUp() {
        parent::setUp();

        $account = $this->drupalCreateUser([
            'access content',
        ]);
        $this->drupalLogin($account);
    }

}
```

# Test

```
public function testMachineName() {  
  
    // Visit the machine name test page which contains two machine name fields.  
    $this->drupalGet('form-test/machine-name');  
  
    // Get page and session.  
    $page = $this->getSession()->getPage();  
  
    $title = $page->findField('machine_name_1_label');  
    $machine_name_field = $page->findField('machine_name_1');  
  
    $machine_name_wrapper = $machine_name_field->getParent();  
    $machine_name_value = $page->find('css', '#edit-machine-name-1-label-machine-name-suffix .machine-name-v  
  
    // Assert field is initialized correctly.  
    $this->assertNotEmpty($machine_name_value, 'Machine name field must be initialized');  
  
    // Set the value for the field, triggering the machine name update.  
    $title->setValue('Test value !0-9@');  
  
    // Wait the set timeout for fetching the machine name.  
    $this->assertJsCondition('jQuery("#edit-machine-name-1-label-machine-name-suffix .machine-name-v  
  
    // Validate the generated machine name.  
    $this->assertEquals('test_value_0_9_', $machine_name_value->getHtml(), $test_info['message']);  
}
```

# Summary

- Tests browser level interactions
- Needs PhantomJS
- Generates Screenshots!

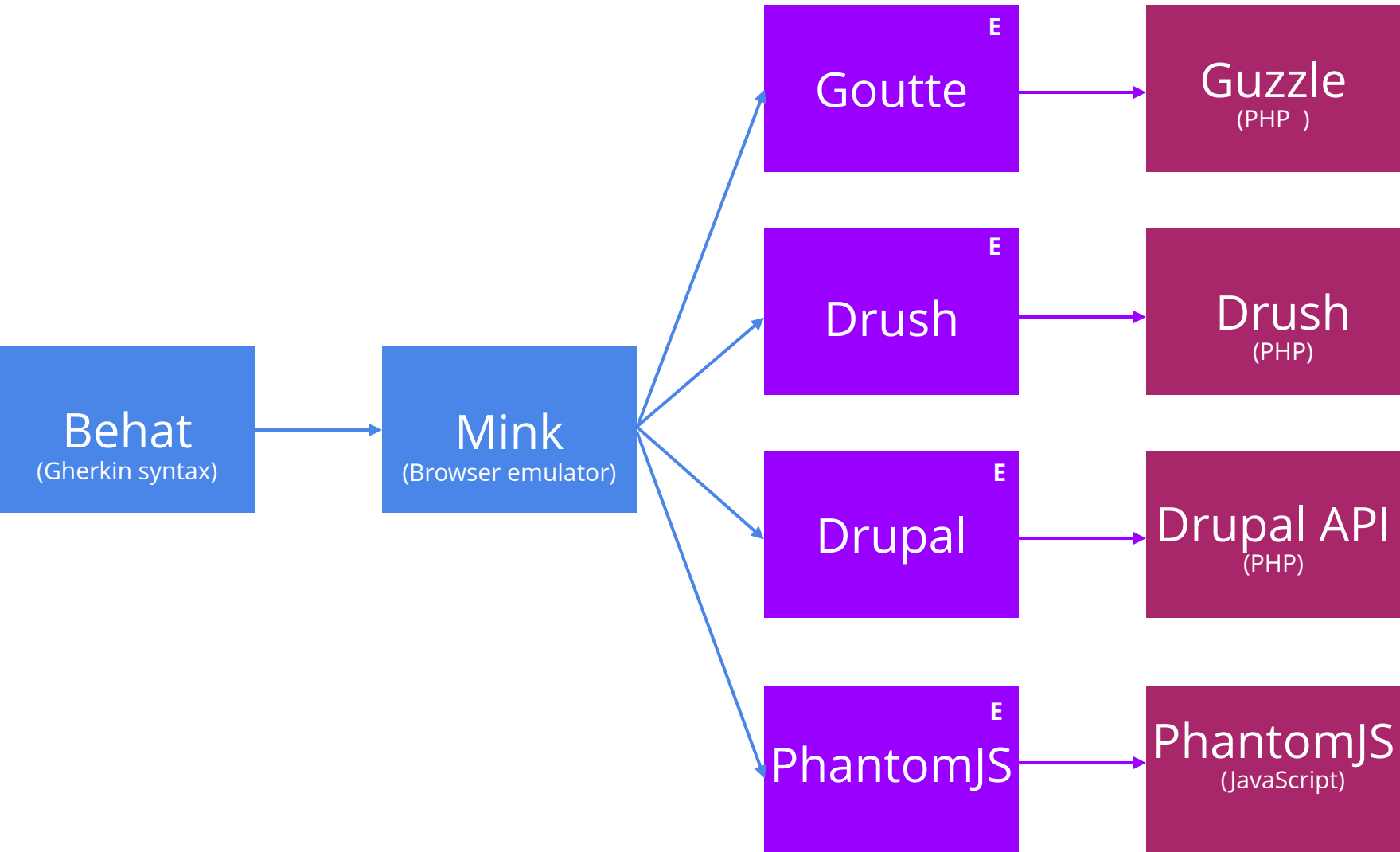
## Machine name fields

Machine name 1 label  Machine name: test\_value\_0\_9\_Edit

Machine name 2 label

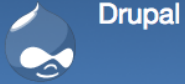


## **2.5 Behat Tests**



Driver

Ctrl / Emulator



Drupal

[Home](#)

[Home](#)

## Website feedback

**Your name**  
user

**Your email address**  
user@example.com

**Subject \***

**Message \***

Send yourself a copy

[Send message](#)

[Preview](#)



Drupal

[Home](#)

[Home](#)

## Website feedback

**Your name \***

**Your email address \***

**Subject \***

**Message \***

[Send message](#)

[Preview](#)

**1. behat.yml**

```
default:
  suites:
    contact:
      contexts:
        - DrupalCoreExtension\Context\DrupalContext
        - Drupal\DrupalExtension\Context\ConfigContext
        - Drupal\DrupalExtension\Context\DrupalContext
        - Drupal\DrupalExtension\Context\MinkContext
        - Drupal\DrupalExtension\Context\MarkupContext
        - Drupal\DrupalExtension\Context\MessageContext
      filters:
        tags: "@contact"
  extensions:
    Behat\MinkExtension:
      base_url: http://localhost/
      sessions:
        default:
          goutte: ~
    Drupal\DrupalExtension:
      api_driver: "drupal"
      drupal:
        drupal_root: "/www/htdocs/docroot"
      region_map:
        left_sidebar: "#sidebar-first"
        content: "#content"
      selectors:
        error_message_selector: '.messages--error'
```

## **2. Feature**

@d8 @api @contact

Feature: Contact

In order to prove the Drupal contact module is working properly for Drupal 8

As a site administrator

I need to check end user and contact form administrator functionality.

Scenario: Visit feedback form as a logged in user.

Given I am logged in as a user with the "authenticated user" role

When I am on `"/contact"`

Then I should see the text "Website feedback"

And I should see the text "Your name"

And I should see the text "Your email address"

And I should see the text "Subject"

And I should see the text "Message"

And I should see the text "Send yourself a copy"

And I should not see "name" field

And I should not see "mail" field

And I should see "subject[0][value]" field

And I should see "message[0][value]" field

And I should see "Preview" button

And I should see "Send message" button

Scenario: Visit feedback form as a anonymous user.

Given I am an anonymous user

When I am on `"/contact"`

Then I should see the text "Website feedback"

And I should see the text "Your name"

And I should see the text "Your email address"

And I should see the text "Subject"

And I should see the text "Message"

And I should not see the text "Send yourself a copy"

And I should see "name" field

And I should see "mail" field

And I should see "subject[0][value]" field

And I should see "message[0][value]" field

And I should see "Preview" button

And I should see "Send message" button

# 3. Context





# 4. Extension

# Drupal Core Extension

The image displays a GitHub repository for 'vijaycs85 / drupal-core-extension'. It is divided into three vertical panels showing different parts of the project.

- Left Panel:** Shows the repository overview. The README.md file is selected, displaying the title 'Behat Drupal Core E...', a description 'Provides behat tests for Drupal 8 core the scenarios.', and a 'Coverage' section.
- Middle Panel:** Shows the 'composer.json' file in the 'master' branch. The commit message is 'vijaycs85 Add PSR-4 mapping'. The file content is as follows:

```
1 {
2   "name": "vijaycs85/drupal-core-extension",
3   "type": "behat-extension",
4   "description": "Behat test for Drupal core functionality",
5   "keywords": ["drupal", "web", "test"],
6   "homepage": "http://github.com/vijaycs85/drupal-core-extens",
7   "license": "GPL-2.0+",
8   "authors": [
9     {
10      "name": "Vijaya Chandran Mani",
11      "email": "vijaycs85@gmail.com"
12    }
13  ],
14  "require": {
15    "drupal/drupal-extension": "^3.2"
16  },
17  "require-dev": {
18    "phpunit/phpunit": "3.7.*"
19  },
20  "autoload": {
21    "psr-4": {
22      "DrupalCoreExtension\\": "src/"
23    }
24  }
25 }
```
- Right Panel:** Shows the 'contact.feature' file in the 'features' directory. The commit message is 'vijaycs85 Adding WebAssert from Drupal 8 core'. The file content is as follows:

```
1 @d8 @api @contact
2 Feature: Contact
3 In order to prove the Drupal contact module is working properly
4 As a site administrator
5 I need to check end user and contact form administrator function
6
7 Scenario: Visit feedback form as a logged in user.
8 Given I am logged in as a user with the "authenticated user" r
9 When I am on "/contact"
10 Then I should see the text "Website feedback"
11 And I should see the text "Your name"
12 And I should see the text "Your email address"
13 And I should see the text "Subject"
14 And I should see the text "Message"
15 And I should see the text "Send yourself a copy"
16 And I should not see "name" field
17 And I should not see "mail" field
18 And I should see "subject[0][value]" field
19 And I should see "message[0][value]" field
20 And I should see "Preview" button
21 And I should see "Send message" button
22
23 Scenario: Visit feedback form as a anonymous user.
24 Given I am an anonymous user
25 When I am on "/contact"
26 Then I should see the text "Website feedback"
27 And I should see the text "Your name"
28 And I should see the text "Your email address"
```

# 5. Output

```
→ php vendor/bin/behat -c behat.yml -s contact
```

```
@d8 @api @contact
```

```
Feature: Contact
```

```
In order to prove the Drupal contact module is working properly for Drupal 8
```

```
As a site administrator
```

```
I need to check end user and contact form administrator functionality.
```

```
Scenario: Visit feedback form as a logged in user.
```

```
Given I am logged in as a user with the "authenticated user" role
```

```
When I am on "/contact"
```

```
Then I should see the text "Website feedback"
```

```
And I should see the text "Your name"
```

```
And I should see the text "Your email address"
```

```
And I should see the text "Subject"
```

```
And I should see the text "Message"
```

```
And I should see the text "Send yourself a copy"
```

```
And I should not see "name" field
```

```
And I should not see "mail" field
```

```
And I should see "subject[0][value]" field
```

```
And I should see "message[0][value]" field
```

```
And I should see "Preview" button
```

```
And I should see "Send message" button
```

```
Scenario: Visit feedback form as a anonymous user.
```

```
Given I am an anonymous user
```

```
When I am on "/contact"
```

```
Then I should see the text "Website feedback"
```

```
And I should see the text "Your name"
```

```
And I should see the text "Your email address"
```

```
And I should see the text "Subject"
```

```
And I should see the text "Message"
```

```
And I should not see the text "Send yourself a copy"
```

```
And I should see "name" field
```

```
And I should see "mail" field
```

```
And I should see "subject[0][value]" field
```

```
And I should see "message[0][value]" field
```

```
And I should see "Preview" button
```

```
And I should see "Send message" button
```

```
Scenario: Visit contact form admin pages as administrator.
```

```
Given I am logged in as a user with the "administer contact forms" permission
```

```
When I am on "/admin/structure/contact"
```

```
Then I should see the heading "Contact forms"
```

```
Then I should see the text "Personal contact form"
```

```
3 scenarios (3 passed)
```

```
32 steps (32 passed)
```

```
0m5.49s (30.77Mb)
```

# **3. Lessons learned**

# Code

- Not public property - public vs private vs **protected**
- No utility class/service
- Write code -> write test -> refactor -> repeat

# Run Tests

```
$ phantomjs --ssl-protocol=any --ignore-ssl-errors=true vendor/jcalderonzumba/gastonjs/src/Client/ma
```

```
$ php core/scripts/run-tests.sh --verbose --url http://d8 --class "Drupal\Tests\permission_ui\Unit\
```



# Annotations

- @group
- @runTestsInSeparateProcesses
- @preserveGlobalState



VijayaChandran Mani @vijaycs85 · 21h

Behat testing #Drupal core's Contact module. Equivalent of BrowserTest in < 6 sec! ❤️

```

→ PHP vendor/bin/behat -c behat.yml -s contact
@id @api @contact
Features: Contact
  In order to prove the Drupal contact module is working properly for Drupal 8
  As a site administrator
  I need to check and user and contact form administrator functionality.

  Scenario: Visit feedback form as a logged in user.
    Given I am logged in as a user with the "authenticated user" role
    When I am on "/contact"
    Then I should see the text "Website feedback"
    And I should see the text "Your name"
    And I should see the text "Your email address"
    And I should see the text "Subject"
    And I should see the text "Message"
    And I should see the text "Send yourself a copy"
    And I should not see "name" field
    And I should not see "mail" field
    And I should see "subject[0][value]" field
    And I should see "message[0][value]" field
    And I should see "Preview" button
    And I should see "Send message" button

  Scenario: Visit feedback form as a anonymous user.
    Given I am an anonymous user
    When I am on "/contact"
    Then I should see the text "Website feedback"
    And I should see the text "Your name"
    And I should see the text "Your email address"
    And I should see the text "Subject"
    And I should see the text "Message"
    And I should not see the text "Send yourself a copy"
    And I should see "name" field
    And I should see "mail" field
    And I should see "subject[0][value]" field
    And I should see "message[0][value]" field
    And I should see "Preview" button
    And I should see "Send message" button

  Scenario: Visit contact form admin pages as administrator.
    Given I am logged in as a user with the "administer contact forms" permission
    When I am on "/admin/structure/contact"
    Then I should see the heading "Contact forms"
    Then I should see the text "Personal contact form"

3 scenarios (3 passed)
32 steps (32 passed)
@vS.49s (30.77%)

```

3 8 17



dawehner @da\_wehner

Following

Replying to @vijaycs85

Well, now you just need to reset the environment afterwards ;)

7:21 PM - 1 Nov 2017

1

# References

- <https://www.drupal.org/docs/8/testing>
- <https://www.lullabot.com/articles/an-overview-of-testing-in-drupal-8>
- <https://www.slideshare.net/xsist10/i-put-on-my-mink-and-wizard-behat>

**Thank you!**