Introduction to Drupal 8 Development

By Joshua Li

joshua@lirujia.com

Who am I

- Drupal developer in Gov
- DrupalACT organizer
- Presenter in different Drupal/PHP conferences
- Drupal id: rli
- https://lirujia.com

The changes in Drupal 8 we know

- Symfony (OOP)
- CMI (workflow)
- Twig (Theme)
- Backbone.js (Inline editor)
- REST (Third party integration)
- ...

For frontend developers

- Preprocess functions
- Twig
- Libraries.yml
- backbone.js

Preprocess functions

Good news:

The preprocess functions from Drupal 7 remain in Drupal 8.

So we can use THEME_preprocess_page, etc.

Use template_preprocess_hook in your module.

Twig

Variables from hook_theme are available in .twig files like the old tpl.php files.

Filters are available in twig files -- logic in twig.

Learn twig: http://twig.sensiolabs.org/documentation

Add css and JS

No more drupal_add_css and drupal_add_js. Always use render array. (You should have been using it in Drupal 7 anyway)

Register css and js in libraries.yml and attach the library in your render array.

Always think about cache when you render something

Add css and JS

```
cuddly-slider:
  version: 1.x
  css:
    theme:
      css/cuddly-slider.css: {}
  js:
    js/cuddly-slider.js: {}
```

```
    return array(
      '#theme' => 'your_module_theme_id',
      '#someVariable' => $some_variable,
      '#attached' => array(
        'library' =>  array(
          'your_module/library_name'
        ),
      ),
    );
```

Reference: https://www.drupal.org/developing/api/8/assets

Caching tags

```
$renderer = \Drupal::service('renderer');

$config = \Drupal::config('system.site');

$build = [
  '#prefix' => '',
  '#markup' => t('Hi, welcome back to @site!', [
    '@site' => $config->get('name'),
  ]),
  '#suffix' => '',
];
$renderer->addCacheableDependency($build, $config);
```

Reference: https://www.drupal.org/developing/api/8/render/arrays/cacheability

Backbone.js

Inline editor

Model - View pattern

Reference: http://backbonejs.org/

The challenge for a Drupal 7 developer

From process oriented programming to object oriented programming

The basic of symfony

From routing to controller

No more hook_menu

```
example.content:
  path: '/example'
  defaults:
    _controller: '\Drupal\example\Controller\ExampleController::content'
    _title: 'Hello World'
  requirements:
    _permission: 'access content'
```

Put the routing info in your module.routing.yml

No more hook_menu

Put your menu callback in your controller

Reference:
https://www.drupal.org/node/2116767

```php
namespace Drupal\example\Controller;

use Drupal\Core\Controller\ControllerBase;

/**
 * An example controller.
 */
class ExampleController extends ControllerBase {

  /**
   * {@inheritdoc}
   */
  public function content() {
    $build = array(
      '#type' => 'markup',
      '#markup' => t('Hello World!'),
    );
    return $build;
  }

}
```

No more hook_menu (for form callback)

```
example.form:
  path: '/example-form'
  defaults:
    _title: 'Example form'
    _form: '\Drupal\example\Form\ExampleForm'
  requirements:
    _permission: 'access content'
```

Put the routing info in your module.routing.yml

No more hook_menu (form callback)

Put your menu callback in your controller

Reference:
https://www.drupal.org/node/2117411

```php
/**
 * @file
 * Contains \Drupal\example\Form\ExampleForm.
 */

namespace Drupal\example\Form;

use Drupal\Core\Form\FormBase;
use Drupal\Core\Form\FormStateInterface;

/**
 * Implements an example form.
 */
class ExampleForm extends FormBase {

  /**
   * {@inheritdoc}
   */
  public function getFormId() {
    return 'example_form';
  }

  /**
   * {@inheritdoc}
   */
  public function buildForm(array $form, FormStateInterface $form_state) {
    $form['phone_number'] = array(
      '#type' => 'tel',
      '#title' => $this->t('Your phone number'),
    );
    $form['actions']['#type'] = 'actions';
    $form['actions']['submit'] = array(
      '#type' => 'submit',
      '#value' => $this->t('Save'),
      '#button_type' => 'primary',
    );
    return $form;
  }
```

No more hook_menu (for form callback)

```
  /**
   * {@inheritdoc}
   */
  public function validateForm(array &$form, FormStateInterface $form_state) {
    if (strlen($form_state->getValue('phone_number')) < 3) {
      $form_state->setErrorByName('phone_number', $this->t('The phone number is too
short. Please enter a full phone number.'));
    }
  }

  /**
   * {@inheritdoc}
   */
  public function submitForm(array &$form, FormStateInterface $form_state) {
    drupal_set_message($this->t('Your phone number is @number', array('@number' =>
$form_state->getValue('phone_number'))));
  }

}
```

Services

To replace the Drupal module APIs

Drupal services

Drupal 8 token example

```
 $token_service = \Drupal::token();
// Replace the token for subject.
$email_auth = $token_service->replace($config['recipient'], array('comment' => $entity));
```

Drupal 7 token example

```
// print node id
print token_replace('[node:nid]', array('node' => $node));
```

Drupal API

Living in the classes now !

Drupal API

l() and url() are removed in favor of a routing based URL generation API

Drupal 7:

```
// Internal path.
$internal_link = l(t('Book admin'), 'admin/structure/book');
```

Drupal 8:

```
// Internal path (defined by a route in Drupal 8).
use Drupal\Core\Url;
$url = Url::fromRoute('book.admin');
$internal_link = \Drupal::l(t('Book admin'), $url);
```

Reference:
https://www.drupal.org/node/2346779

The workflow to port your module to D8

1. Find your API in D7
2. Use the Drupal change record to find the change
3. Use the suggested example to rewrite

Reference:
https://www.drupal.org/list-changes/drupal

Plugins

An interface that you can implement

Block

Annotation

Reference:
https://www.drupal.org/developing/api/8/block_api

```php
namespace Drupal\fax\Plugin\Block;

use Drupal\Core\Block\BlockBase;

/**
 * Provides a 'Fax' block.
 *
 * @Block(
 *   id = "fax_block",
 *   admin_label = @Translation("Fax block"),
 * )
 */
class FaxBlock extends BlockBase {
  // Override BlockPluginInterface methods here.
}
```

Block

Build function from blockbase interface

Reference:
https://www.drupal.org/developing/api/8/block_api

```
  /**
   * {@inheritdoc}
   */
  public function build() {

  $config = $this->getConfiguration();
    $fax_number = isset($config['fax_number']) ? $config['fax_number'] : '';
    return array(
      '#markup' => $this->t('The fax number is @number!', array('@number' =>
$fax_number)),
    );
  }
```

Workflow changes

No more features

Drupal CMI

Before:

1. Make your config change locally
2. Generate features module
3. Commit your code
4. Enable your features module

After:

1. Make your config change locally
2. Drush cex
3. Commit your code
4. Drush cim

Case study

Define a content filter to convert text into links to a taxonomy term page.

For example, we want to turn all ‘Drupal’ words in node body into links to the ‘Drupal’ tag page.

Solution in Drupal 7

Hook_filter_info to define a new filter

- **Settings callback** – settings form if there is any settings I put a link to our config page here
- **Tips callback** – tips text
- **Process callback** – the function that replace the text
- A configuration page to allow user to choose which vocabulary we need to replace with. (**hook_menu and callback**)

```
function hook_filter_info() {
  $filters['filter_html'] = array(
    'title' => t('Limit allowed HTML tags'),
    'description' => t('Allows you to restrict the HT
such as JavaScript events, JavaScript URLs and CSS st
    'process callback' => '_filter_html',
    'settings callback' => '_filter_html_settings',
    'default settings' => array(
      'allowed_html' => '<a> <em> <strong> <cite> <bl
      'filter_html_help' => 1,
      'filter_html_nofollow' => 0,
    ),
    'tips callback' => '_filter_html_tips',
  );
```

Reference:
https://api.drupal.org/api/drupal/modules%21filter%21filter.api.php/function/hook_filter_info/7.x

Our D7 code

```php
/**
 * Implements hook_filter_info().
 */
function termfilter_filter_info() {
  $filters = array();

  $filters['termfilter'] = array(
    'title'            => t('Term filter'),
    'description'      => termfilter_help('admin/modules#description', array()),
    'process callback' => '_filter_termfilter_process',
    'settings callback' => '_filter_termfilter_settings',
    'tips callback'    => '_filter_termfilter_tips',
    'default settings' => array(),
  );

  return $filters;
}
```

Our D7 code

```php
/**
 * Serves as hook_filter_info() process operation callback.
 * Searches for whitelisted terms and replaces them with <a> tags.
 *
 * @param $text
 *   the text to be filtered
 * @param $filter
 *   the filter data
 *
 * @return
 *   returns the text, processed by the filter
 */
function _filter_termfilter_process($text) {
  $list = _termfilter_list();
  return _termfilter_perform_subs($text, $list);
}
```

Our D7 code

```php
function termfilter_menu() {
  $items = array();

  $items['admin/config/content/termfilter'] = array(
    'title' => 'Term filter',
    'description' => 'Replaces terms inside posts with A tags.',
    'page arguments' => array('termfilter_admin_list'),
    'page callback' => 'drupal_get_form',
    'access arguments' => array('administer terms filtered'),
  );

  return $items;
}
function termfilter_admin_list($form, &$form_state) {

  $vocab_list = taxonomy_vocabulary_get_names();
  $checklist_vocab_array = array();
  foreach ($vocab_list as $item) {
    $key = $item->machine_name;
    $value = $item->name;
    $checklist_vocab_array[$key] = $value;
  }

  $form['termfilter_vocablist'] = array(
    '#type'            => 'radios',
    '#title'           => t('Select the vocabulary you want to filter.'),
    '#position'        => 'left' ,
    '#options'         => $checklist_vocab_array ,
    '#default_value'   => variable_get('termfilter_vocablist'),
  );

  return system_settings_form($form);
}
```

Our D8 code

- Check the Drupal change record for hook_filter_info
- Result page https://www.drupal.org/node/2015901
- Create your filter plugin class and copy the code in
- Check the Drupal change record for hook_menu
- Jump to IDE ...
- Jump to D8 site ...

Conclusion

- Drupal 8 is a complete rewrite. For developer with no knowledge of Object Oriented Programming (OOP), it is even harder to understand how it works
- For a D7 develop to onboard quickly, Drupal change record is a bypass that we can use the examples directly
- For JAVA, C++, developers, welcome to Drupal community!

What will I do next?

Go start to port a module.

Let me know if I can help.

The complete module code in this presentation: https://github.com/rujiali/Drupal_term_filter