Prepare your extension for Joomla 4

Allon Moritz
JAB17 2. June 2017
About Me

Allon Moritz
@digitpeak / @laoneo

Founder Digital Peak GmbH

Doing Joomla extensions since 2007

Developer Joomla 4

Team Lead Media Manager

GsoC Mentor
Agenda

● Namespaces in PHP
● Dependency injection
● MVC Layer
● Component
  ○ Namespacing
  ○ Dispatcher
  ○ Form
● Modules
● Plugins
● Backwards compatibility
● Questions
Namespaces in PHP

- Organising classes in a defined structure
- Autoloaded through the classloader, no manual `include` or `require` anymore
- Example `ContentModelArticles` becomes `\Joomla\Component\Content\Administrator\Model\Articles`
- `JLoader` can handle the namespaces automatically
- We can distinguish between front end and back end classes
- Files can be found in `/libraries/src/Joomla/CMS`
Namespaces in PHP

```php
<?php

/** @package Joomla.Administrator */

namespace Joomla\Component\Content\Administrator\Model;

defined('_JEXEC') or die;

use Joomla\CMS\Model\ListModel;
use Joomla\Utilities\ArrayHelper;

/**
 * Methods supporting a list of article records.
 *
 * @since 1.6
 */

class Articles extends ListModel
{
}
```
Dependency Injection

- No woodoo
- Better unit testing
- Application and Input get injected into the dispatcher and the MVC layer
MVC Layer: Introduction

- Basically the legacy layer becomes the default MVC in Joomla 3
- Convert your “New MVC” component to the namespaced MVC, it was done with com_config
- “New MVC” layer will be deprecated
MVC Layer: Libraries

- The old classes still exist but the real files do exist in
- JControllerLegacy is now \Joomla\CMS\Controller\Controller
- JModelLegacy is now \Joomla\CMS\Model\Model
- JViewLegacy is now \Joomla\CMS\View\AbstractView
- JTable stays is now \Joomla\CMS\Table\Table
- JPlugin is now \Joomla\CMS\Plugin\CMSPlugin
Component: Namespacing

- A Joomla 4 component will have the structure as in the picture
- Uppercase folders have classes
- Lowercase folders do hold the rest like forms or template files
- Dispatcher is not namespaced
- Root classes become a \ and front
- Controller and Model need an additional parameter `MvcFactory`
- Front end namespace is `\FooVendor\Component\Bar\Site`
- Back end namespace is `\FooVendor\Component\Bar\Administrator`
Component: Namespacing Todo

- Helpers are need to be done on a case per case basis
  - ContentHelper
  - RouteHelper
  - QueryHelper
  - AssociationsHelper
- JHtml
Component: Dispatcher

- Replacement single entry file content.php in root
- Get the application injected
- Dispatches the controller

```php
class ContentDispatcher extends Dispatcher {
    /**
     * The extension namespace
     *
     * @var string
     *
     * @since __DEPLOY_VERSION__
     */
    protected $namespace = 'Joomla\Component\Content';
}
```
Component: Form

- Forms are in the folder forms under root in the extension on, eg. /forms/article.xml
- Extra fields are registered with addfieldprefix attribute which adds the namespace

```xml
<?xml version="1.0" encoding="utf-8"?>
<form>
    <fieldset
        addfieldprefix="Joomla\Component\Categories\Administrator\Field">
        <field name="id" type="text" class="readonly" label="JGLOBAL_FIELD_ID"
            description ="JGLOBAL_FIELD_ID_DESC" size="10" default="0"
            readonly="true" />
    </fieldset>
</form>
```
Modules

- Module file stays as single entry file
- Namespacing is done on the additional classes
- Modules namespace is `\FooVendor\Module\Bar\Administrator` for front end
- Modules namespace is `\FooVendor\Module\Bar\Site` for front end
Plugins

- Plugin file stays the same not namespaced
- When you have additional classes, they can be namespaced
- Plugin namespace is `\FooVendor\Plugin\Content\Bar`
Backwards compatibility

- Target is that a Joomla 3 extension will run on Joomla 4
- Target is that a Joomla 4 extension will run on Joomla 3.9
- Classmapping for libraries so JForm still exists and maps to `\Joomla\CMS\Form\Form`
- Changes are added not replaced to make sure the old code works
- Support Joomla 3 extensions during full Joomla 4 lifecycle
- Changing to a more service oriented architecture
- No Class overloading anymore
- Event system has changed completely
Questions?

Slides on https://joomla.digital-peak.com/jab17
Thank you and enjoy the day :-}