



# Drupal 8

Core and API Changes

Shabir Ahmad

MS Software Engg. NUST

Principal Software Engineer – PHP/Drupal

[enr.shabir@yahoo.com](mailto:enr.shabir@yahoo.com)

# Agenda

- What's coming in Drupal 8 for...
  - End users and clients?
  - Site builders?
- Designers and themers?
  - Twig templates
- Developers?
  - Symfony Components
  - Module development
  - Plugin System
- Tools and External libraries (Composer, Drush, Guzzle, DrupalAppConsole)
- API Function changes
- How can I contribute to community

# End Users & Site Builders

- WYSIWYG in Core
- Better Authoring Experience
- In place editing
- Responsive in its Core (Mobile friendly admin panel)
- Responsive admin panel
- Web services and view in its core
- English no more special language. Multilingual out of the box

[http://www.slideshare.net/AcquiaInc/drupal-8-preview-what-to-expect?qid=5f17d509-2207-4521-9994-a093a776a41b&v=qf1&b=&from\\_search=5](http://www.slideshare.net/AcquiaInc/drupal-8-preview-what-to-expect?qid=5f17d509-2207-4521-9994-a093a776a41b&v=qf1&b=&from_search=5)

Designers & Themers

# HTML



Yeyyyy!!! We can use all html 5 benefits without installing any contrib modules

# HTML5 Form Elements



iPod 14:23

Phone:

Website:

Email:

Tickets required:



Previous Next AutoFill Done

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
+ * #	0	⌫

```
$form['telephone'] = array(  
    '#type' => 'tel',  
    '#title' => t('Phone'),  
);
```

```
$form['website'] = array(  
    '#type' => 'url',  
    '#title' => t('Website'),  
);
```

```
$form['email'] = array(  
    '#type' => 'email',  
    '#title' => t('Email'),  
);
```

```
$form['tickets'] = array(  
    '#type' => 'number',  
    '#title' => t('Tickets required'),  
);
```

# HTML5 Form Elements



# TWIG Templates





# What is TWIG??

- A PHP Template Engine to separate Presentation layer from the Controller/Model (PHP)
- Developed by same people behind Symfony; yes, that's in Drupal 8 Core too

But PHP was Just fine,  
Why on earth to use twig!!

# Common Complaints

- Mixed PHP with HTML can be just plain sloppy and hard to read
- PHP still has to scan html documents looking for all those <?php tags amongst the HTML
- Designers have too much power and can open security bugs in the presentation layer with PHP
- Defining a function or filtering for theming was sloppy — no real standard way
- Php is faster but twig is safer As a developer you can' do any php in templates like

```
<?php $user = db_query("SELECT n.nid FROM users WHERE uid = ".$_GET['uid']."); ?>
```

# Benefits

- An extendable template engine
- Secure (can enable variable output escaping globally)
- It's Unit Tested
- Great documentation and online resources
- Not coupled to Symfony so it has its own roadmap and community
- Easy to understand clean syntax
- Once you know it, can use it in other PHP frameworks outside of Drupal
- Syntax support in several IDEs (Sublime text, PHPStorm, Netbeans)

<http://twig.sensiolabs.org/doc/templates.html>

# Lets understand the basics of Twig

Operation	PHP (Drupal 7)	Twig (Drupal 8)
Print output	<code>&lt;?php print \$something['key'] ?&gt;</code>	<code>{{ something.key }}</code>
Comments	<code>&lt;?php // .... Or /* ...*/</code>	<code>{ # ... #}</code>
Filters	<code>&lt;?php t('Welcome, ' . \$user['name']); ?&gt;</code>	<code>{{ 'Welcome, @name'   t({ '@name': user.name }) }}</code>
Combining filters	<code>&lt;?php strtoupper(t('Welcome, ' . \$user['name'])); ?&gt;</code>	<code>{{ 'Welcome, @name'   t({ '@name': user.name })   upper }}</code>
If - else	<code>&lt;?php if (isset(\$user['name'])) {     echo \$user['name'] } else {     echo 'Who are you?'; ?&gt;</code>	<code>{% if user.name is defined %}     {{ user.name }} {% else %}     Who are you? {% endif %}</code>
Loops	<code>&lt;?php foreach ( \$users as \$key =&gt; \$user ) {     print \$user['name']; } ?&gt;</code>	<code>{% for key, user in users %}     {{ user.name }} {% endfor %}</code>

Operation	PHP (Drupal 7)	Twig (Drupal 8)
Calculation	<code>&lt;?php print \$user['total'] + 1; ?&gt;</code>	<code>{{ users.total + 1 }}</code>
Concat Strings	<code>&lt;?php print strtolower("Greeting ' . \$user['first']); ?&gt;</code>	<code>{% set greeting = 'Hello ' %} {% set name = user.first %} {{ greeting ~ name lower }} {# Hello Shabir #}</code>
Expressions	Various strings and array function	<code>{% if 1 not in [1, 2, 3] %} {% if 'cd' in 'abcde' %} {% if 'Bob' starts with 'B' %} {% if phone matches '^[\d\.]+\$', %}</code>
Logic with Expressions	<i>PHP: &amp;&amp;    &lt;&gt;, and ,or,not</i>	<code>{% if user.name is not defined or user.name == 'shab' %}</code>

<http://twig.sensiolabs.org/doc/templates.html>

# Extending layout

- Don't duplicate the whole twig file just to change a single piece of output. Extend it.

```
{% extends 'layout.html.twig' %}
```

- Conditional Extends: Mobile variable is set in code and passed to template

```
{% extends mobile ? "mobile.html.twig" : "layout.html.twig" %}
```

- Blocks: Define content in blocks to allow for extending of templates.

```
{% block sidebar %}...content... {% endblock %}
```

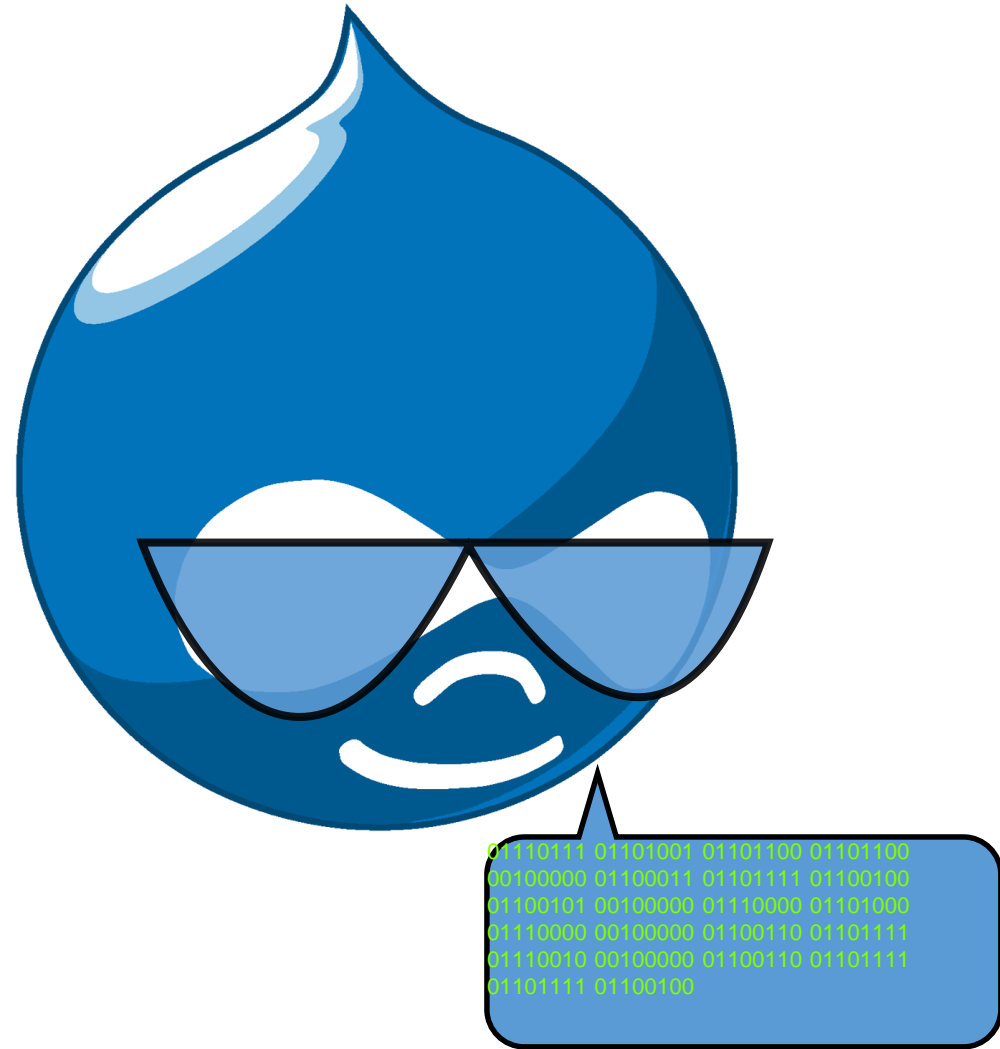
- Debugging code: In setting.php uncomment the following line.

```
# $settings['twig_debug'] = TRUE;  
{{ dump(user) }}
```

Killed Support for IE6,7 and most of IE8



# Changes for coders



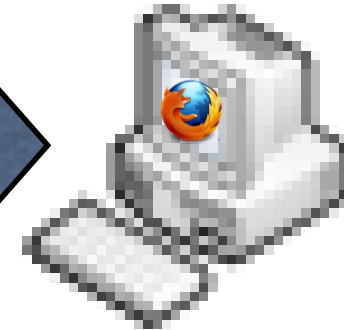
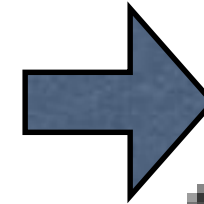
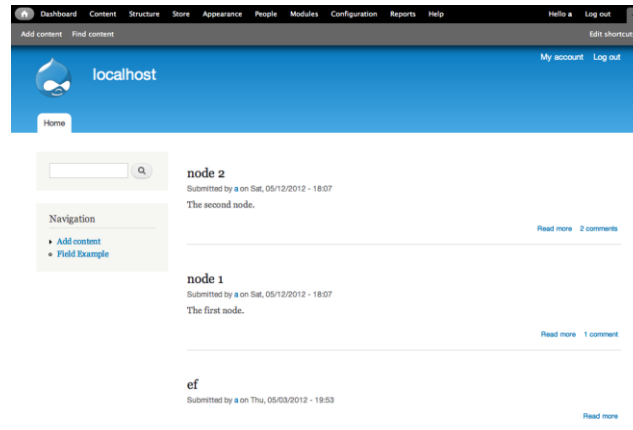
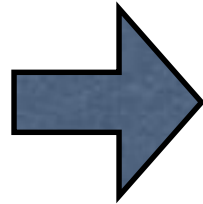


# Web Services



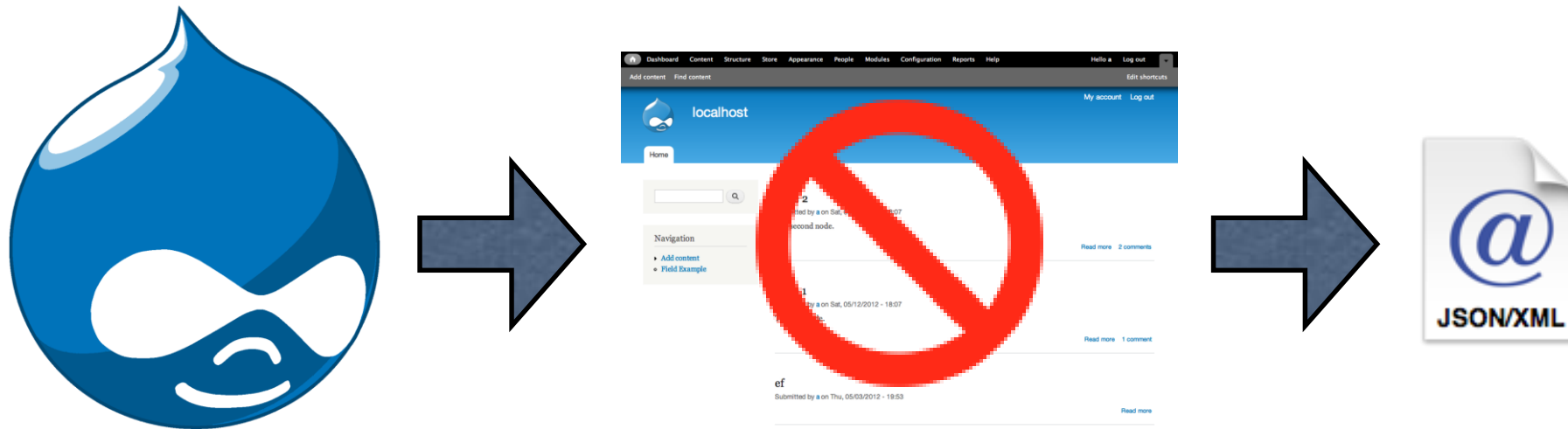
It's like Services module in core, only better.  
Issue tag: WSCCI

# What problems are we trying to solve?

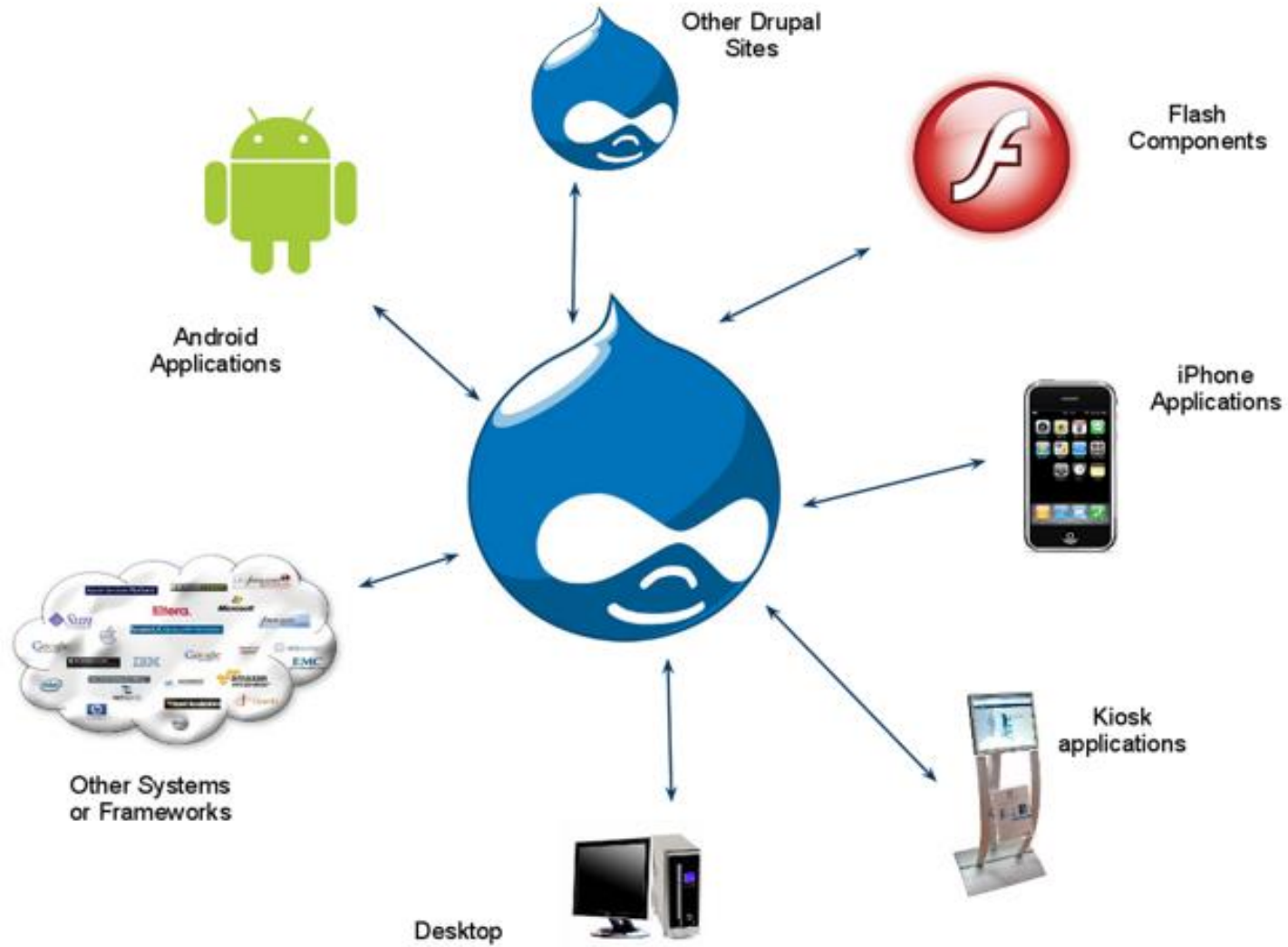


<http://www.youtube.com/watch?v=l98dVUABD4w>

# What problems are we trying to solve?



<http://www.youtube.com/watch?v=I98dVUABD4w>



<http://www.youtube.com/watch?v=l98dVUABD4w>

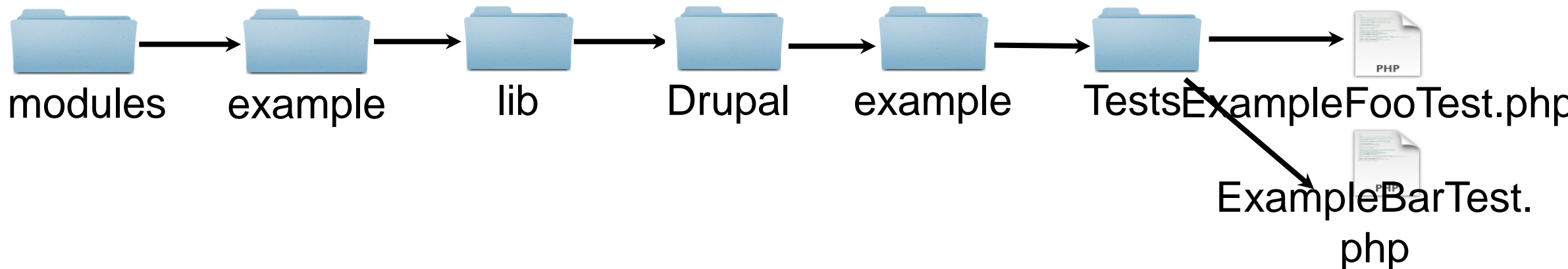
# Module Development!! Welcome Symfony



- Drupal 8 is built on top of the following symphony components
- HttpFoundation and HttpKernel
- Routing
- Eventdispatcher
- Dependency injection
- Class loader
- Yaml, TWIG, Serializer

# Module Anatomy

- The core modules are now in core directory
- The contributed and custom modules are now in modules directory
- .info file is replaced by .info.yml
- .module file are no more mandatory.
- Drupal 8 is built on top of symphony 2 components
- Using PSR-0 for autoloading the classes and controllers



# Lets Develop a Hello World module

- As drupal 8 is now more like a framework, Correct!! So it does have a scaffolding tool! So you people no need to create that PSR-0 imposed nested directories.
- Steps for installation

```
$ cd path/to/drupal8.dev
$ curl -sS https://getcomposer.org/installer | php
$ COMPOSER_BIN_DIR=bin php composer.phar require --dev drupal/console:dev-master
$ ./bin/console --help
```

- Now type the following command and follow the steps

`./bin/console generate:module`

<https://github.com/hechoendrupal/DrupalAppConsole>

# Lets Develop a Hello World module

modules/mymodule/mymodule.info.yml

```
mepco.info x page--front.tpl x views-view--ca x aliases x
1 name: 'My test module'
2 type: module
3 description: 'Drupal camp demo.'
4 core: 8.x
5
```

modules/mymodule/mymodule.local\_task.yml

```
1 mymodule_list_tab:
2   route_name: mymodule.list
3   title: 'List'
4   base_route: mymodule.list
5
6 mymodule_settings_tab:
7   route_name: mymodule.settings
8   title: 'Settings'
9   base_route: mymodule.list
10
11
```



# Lets Develop a Hello World module

Next Add a main route file that replace hook\_menu  
It should also be in main directory with the name  
modules/mymodule/mymodule.routing.yml

```
mepco.info x page-front.tpi x views-view--ca x aliases x Booktest.php userdir.com x
1 mymodule.list:
2   path: '/admin/config/mymodule/list'
3   defaults:
4     _content: '\Drupal\mymodule\Controller\MyController::dolist'
5     _title: 'Mymodule list'
6   requirements:
7     _access: 'TRUE'
8
9 mymodule.settings:
10  path: '/admin/config/also-mymodule/settings'
11  defaults:
12    _content: '\Drupal\mymodule\Controller\MyController::settings'
13    _title: 'Mymodule settings'
14  requirements:
15    _access: 'TRUE'
16
```

# Controller contains main logic and menu Callback

In lib/mymodule/Drupal/Controller/myController.php paste the following code

```
<?php

namespace Drupal\mymodule\Controller;

use Drupal\Core\Controller\ControllerBase;

class MyController extends ControllerBase {

    public function doList() {
        return __METHOD__;
    }

    public function settings() {
        return __METHOD__;
    }

}
```

# Hook\_info is now a plugin

In lib/mymodule/Drupal/plugin/myblock.php paste the following code

```
/**
 * Provides a block with 'Mymodule' links.
 *
 * @Block(
 *   id = "mymodule_my_block",
 *   admin_label = @Translation("Mymodule block")
 * )
 */
class MyBlock extends BlockBase {

    public function build() {
        return array(
            'first_link' => array(
                '#prefix' => '<p>',
                '#type' => 'link',
                '#title' => t('Mymodule List'), // @todo use translation manager
                '#route_name' => 'mymodule.list',
                '#suffix' => '</p>',
            ),
            'second_link' => array(
                '#prefix' => '<p>',
                '#type' => 'link',
                '#title' => t('Mymodule Settings'),
                '#route_name' => 'mymodule.settings',
                '#suffix' => '</p>',
            ),
        );
    }
}
```

# Drupal 8 Plugin System

- So what is plugin????

*A discreet class that executes an operation within the context of a given scope, as a means to extend Drupal's functionality.*



**Configurable,  
reusable**



**Does one  
thing**



**Limited in  
scope**

Some examples of plugins are

- 1) Block
- 2) Custom fields
- 3) Custom field widget
- 4) Cck filter type
- 5) Custom image effects
- 6) Custom Actions

# Benefits of plugin

- Definition and implementation in one place, in D7 it wasn't necessary
- Plugins are lazy loaded (the code is in loaded to memory until called)
- Code is unified and extensible
- Reusable

<http://www.youtube.com/watch?v=2o5uY-iOoMo>

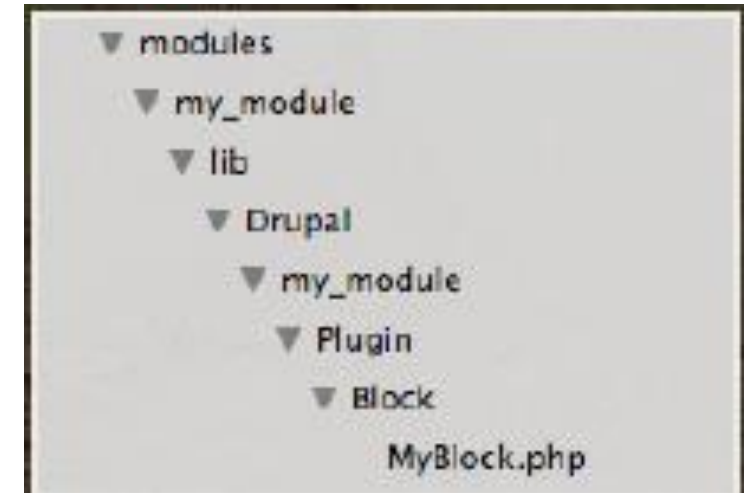
# Core concepts used in Drupal 8

- Dependency Injection
- Service Containers
- PSR-0
- Annotation

# PSR-0 Autoloading

- Fully-qualified namespace should be in the format of *Vendor/Namespace/Classname*
- Directory structure must match namespace for plugin

```
1 <?php
2
3 /**
4  * @file
5  * Contains \Drupal\my_module\Plugin\Block\MyBlock.
6  */
7
8 namespace Drupal\my_module\Plugin\Block;
9
10 class MyBlock {
11     // ...
12 }
```



# Annotation

- Metadata inside php docblocks

```
// Single-line comment, not cached by opcode cache.  
/* Multi-line  
   comment... */
```

Normal Comment

```
/**  
 * This is a docblock. Is cacheable by opcode cache.  
 *  
 * @MyPluginName(  
 *   id = "some_plugin",  
 *   label = @Translation("Some Plugin"),  
 *   description = @Translation("Describing some plugin.")  
 * )  
 */
```

Annotation



# Dependency Injection

- It is a symphony design pattern, some time called inversion of control.
- A class is instantiated inside a class, it means the class should know about the class.
- If the dependent class is changed, the other class must be changed.
- We want the code to be ignorant, the more the code knows the less reusable it will be

```
class SomeClass {  
    function __construct() {  
        $this->handler = new Handler();  
    }  
}  
  
$instance = new SomeClass();
```



```
class SomeClass {  
    function __construct($handler) {  
        $this->handler = $handler;  
    }  
}  
  
$handler = new Handler();  
$instance = new SomeClass($handler);
```

# Service Containers

- Auto-instantiating all service oriented classes with all its registered dependencies
- Also called dependency Injection containers

```
# my_module.services.yml
services:
  my_module.some_class:
    class: Drupal\my_module\namespace\SomeClass
    arguments: [@handler]
```

```
$handler = new Handler();
$instance = new SomeClass($handler);
```



```
/* Service Container */
$instance = \Drupal::service('my_module.some_class');
```

# Some of API functions and hooks are changed in drupal 8

Operation	Drupal 7	Drupal 8
String	check_plain(), drupal_placeholder()	String::placeholder(), String::placeholder()
Node/Entity	<pre>\$nodes = Node_load_multiple(\$nids) Node_view_multiple(\$nodes) Node_save(\$node);</pre>	<pre>\$nodes=Entity_load_multiple('node',\$nids) Entity_view_multiple(\$nodes) Entity_create('node',\$values)-&gt;save()</pre>
Menus	hook_menu menu_get_objects()	Module.routing.yml \$request()->attributes->get('node')
Taxonomies	taxonomy_vocabulary_machine_name_load('forums');	entity_load('taxonomy_vocabulary', 'forums');
Fields	Field_info_field() field_info_instance()	<pre>\$fieldInfo = \Drupal\field\Field::fieldInfo(); \$fieldInfo-&gt;getField(\$entity_type, \$field_name); \$fieldInfo-&gt;getInstance('node', 'article', 'field_name');</pre>
Alters	Drupal_alter() // to alter data Hook_form_alter() Hook_form_id_alter	ModuleHandler::alter() Same as d7

# Some of API functions and hooks are changed in drupal 8

Operation	Drupal 7	Drupal 8
hooks	<code>Module_invoke_all()</code> <code>module_implements()</code> //check if the module implements certain hook	<code>moduleHandler::invokeAll()</code> <code>moduleHandler::getImplementations()</code>
URLs	<code>drupal_get_query_parameters()</code>	<code>UrlHelper::filterQueryParameters(\$url);</code>
Paths (Where Am I)	<code>drupal_get_path()</code>	<code>Drupal_get_path()</code>
User (Who Am I)	Global <code>\$user</code>	<code>\$account = \Drupal::currentUser()</code>
Modules loading and Query entity	<code>module_load_include()</code> <code>EntityFieldQuery</code>	<code>\Drupal::moduleHandler-&gt;loadInclude('node', 'inc', 'node.admin');</code> <code>\Drupal::entityQuery()</code>
Javascript/css	<code>Drupal_add_js, drupal_add_css, drupal_add_library</code>	Gone in drupal 8 <code>#attached = array('js' =&gt; 'myjs.js')</code>

# Other tool.

- Composer
- Guzzle and http dev client (Chrome extension)
- Assetic (Asset manager)
- PHPUNIT for testing
- PSR/Log: Consistent logging from components and drupal
- Drupal scaffolding tool aka drupal app console
- Drush 7

# Core mentoring

- Contribute your efforts to this awesome tool and make it the top of the technologies
- Join irc channels at Tuesday morning and Wednesday evening and get an issue to work on
- Or you can go to [www.corementoring.org](http://www.corementoring.org) and get an issue

# Questions????

- Thank you for you patience!!