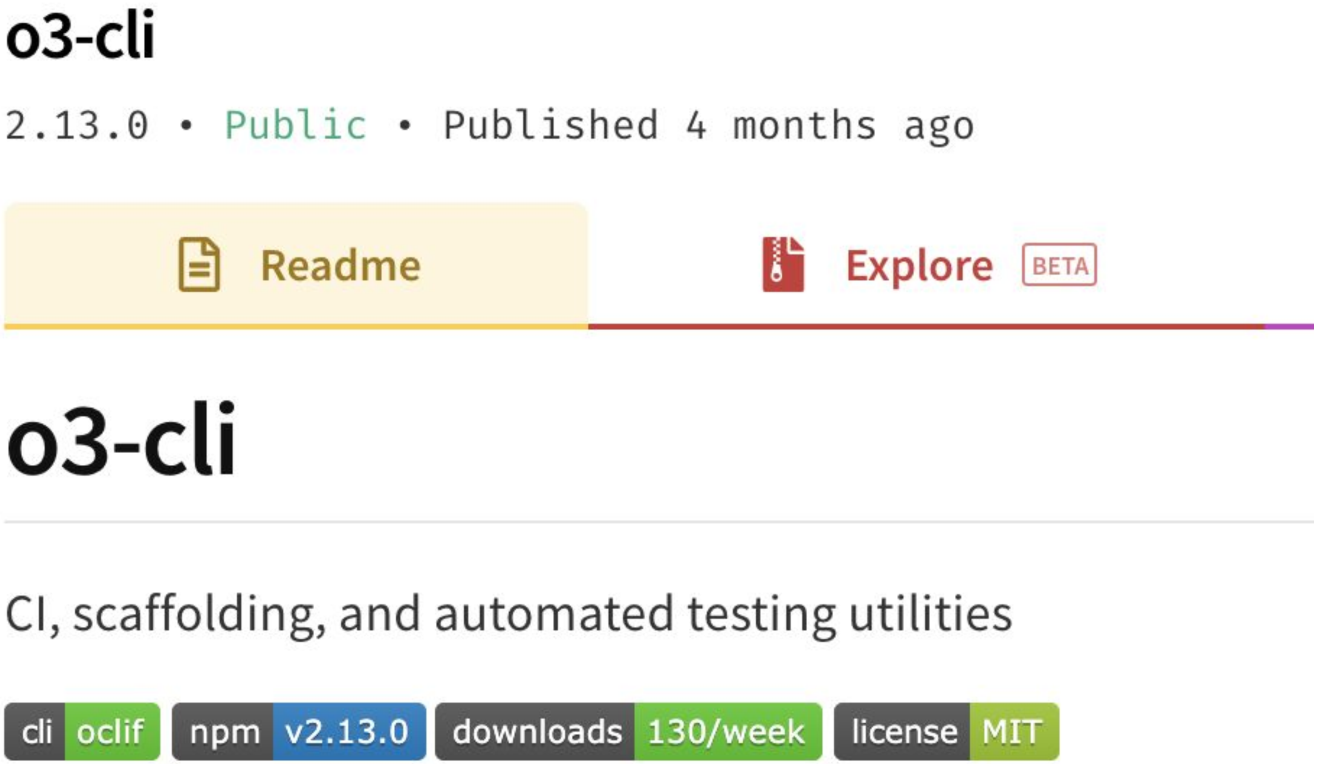
Lost in CMS Translation

Custom Plugin Development across Craft, Drupal, and WordPress

Matt Schaff, O3 World, LLC | CMS Philly 2020

01 / 01 / 2020

We built a CLI tool to interact with all of our clients’ sites.



o3-cli

2.13.0 • Public • Published 4 months ago

o3-cli

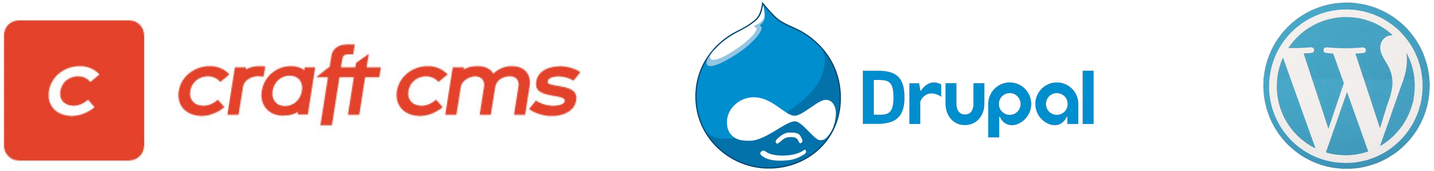
CI, scaffolding, and automated testing utilities

Problem: We needed common interface for the CLI tool regardless of CMS

Problem: We needed common interface for the CLI tool regardless of CMS

Solution: A plugin that does the same thing, for multiple CMSs

BUT…..
How can I translate a single plugin across major CMS frameworks?



BUT…..
How can I translate a single plugin across major CMS frameworks?

craft cms

Drupal

Get ready...

The Basics

THE BASICS | OVERVIEW

	Craft CMS	Drupal	WordPress
License	Proprietary	Open-Source	Open-Source
Language	PHP	PHP	PHP
Framework	Yii	Symfony	N/A
Templating language	Twig	Twig	PHP
Initial Release Date	2013	2000	2003
Plugin pricing model	Free/Paid	Open-Source	Free/Paid
Composer-managed	Yes	Can be	No
Market Share (whatcms.org)	0.23%	1.80%	28.09%
Name for "Unit"	Plugin	Module	Plugin

THE BASICS | CODING GUIDELINES

THE BASICS | CODING GUIDELINES - CRAFT

- Conditions that span multiple lines should have logical operators (`||`, `&&`, etc.) at the end of lines.
- Use type hinting & new-line open-parentheses:

```
public function foo(string $bar = null)
{
}
```

- Chained method calls should each be placed on their own line, with the `->` operator at the beginning of each line.

https://docs.craftcms.com/v3/extend/coding-guidelines.html

THE BASICS | CODING GUIDELINES - DRUPAL

- No need for a `@file` docblock in a class file
- `else` and `elseif` should be on new lines
- Use *same-line* open-parentheses (type-hinting is optional):

```
public function foo($bar = null){
}
```

https://www.drupal.org/docs/develop/standards/coding-standards

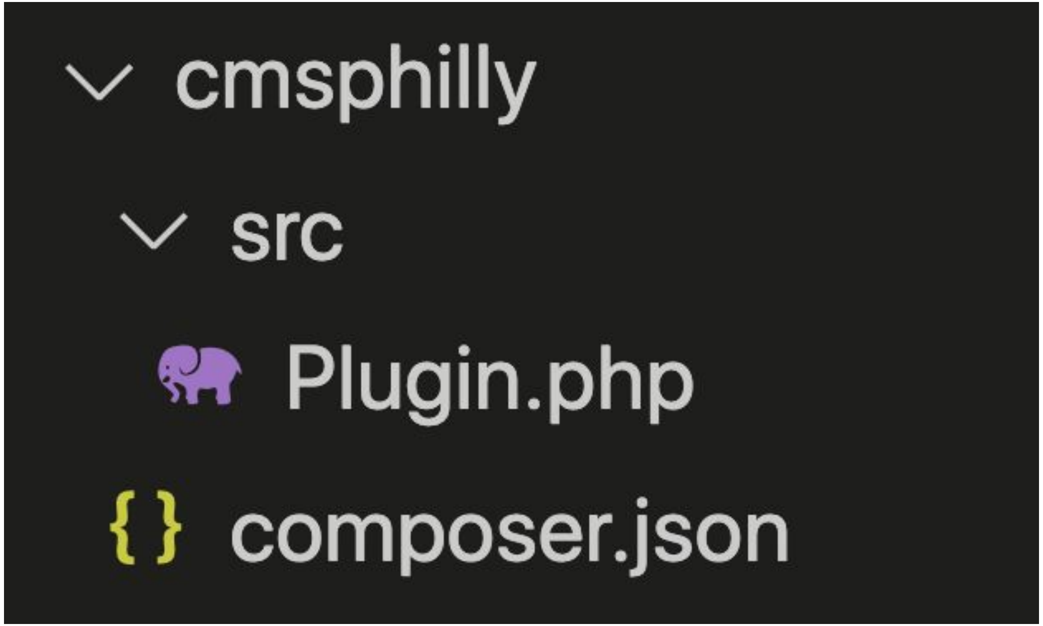
THE BASICS | CODING GUIDELINES - WORDPRESS

- ! Must use *long* `array()` syntax (WP supports PHP 5.3+)
- `else` and `elseif` should be on the *same* line as the previous closing `}`
- Use tabs not spaces

https://make.wordpress.org/core/handbook/best-practices/coding-standards/php/

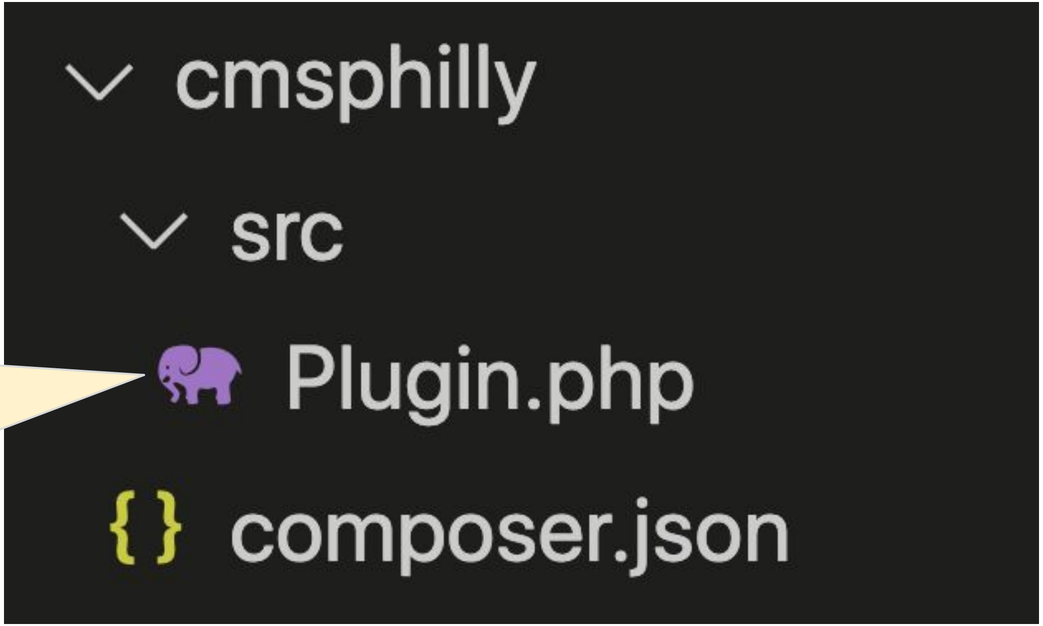
The Plugin Directory

The plugin directory - Craft CMS



```
cmsphilly
  src
    Plugin.php
  composer.json
```

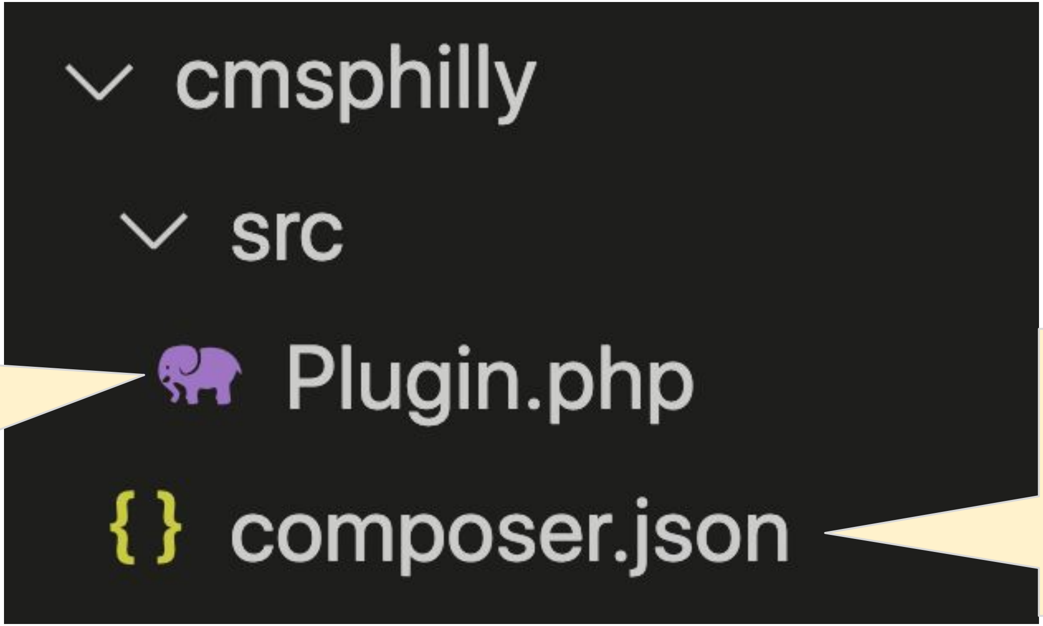
The plugin directory - Craft CMS



cmsphilly
src
Plugin.php
composer.json

Here’s where you hook into Craft events & register your services *at runtime*

The plugin directory - Craft CMS



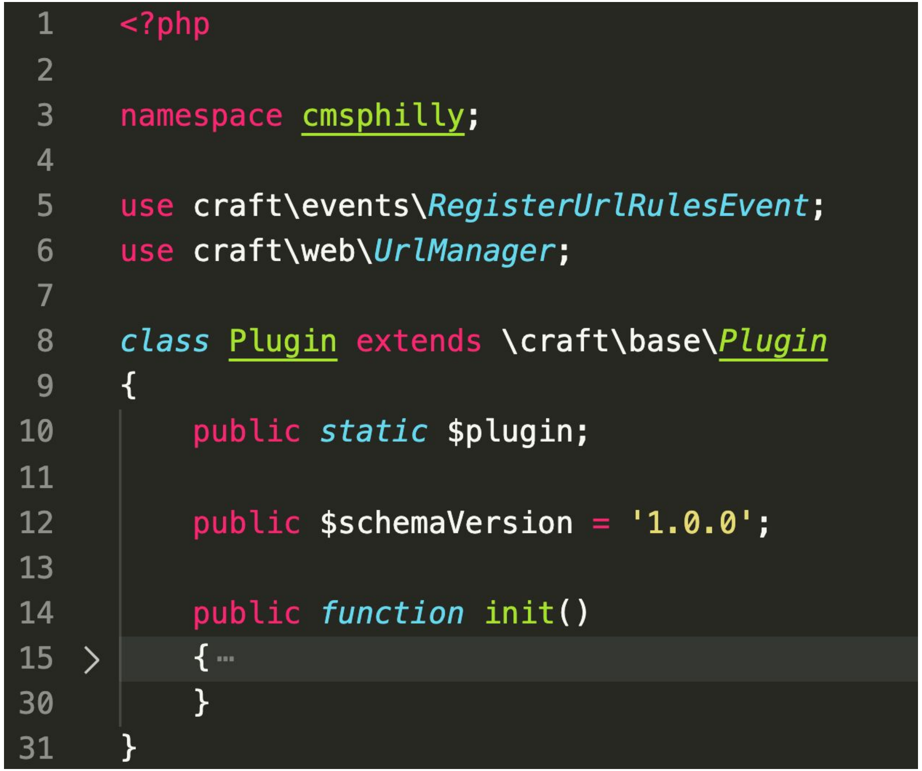
```
cmsphilly
  src
    Plugin.php
  composer.json
```

Here's where you hook into Craft events & register your services *at runtime*

This controls dependencies and autoloading *upon installation*

The plugin directory - Craft CMS

cmsphilly/src/Plugin.php

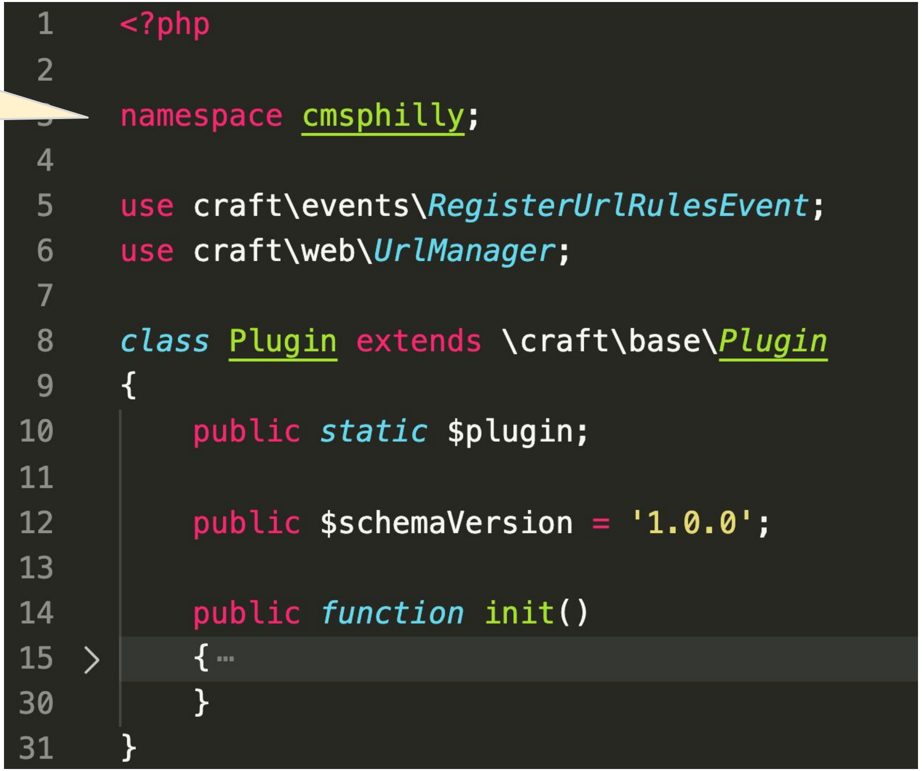


```php
<?php

namespace cmsphilly;

use craft\events\RegisterUrlRulesEvent;
use craft\web\UrlManager;

class Plugin extends \craft\base\Plugin
{
    public static $plugin;

    public $schemaVersion = '1.0.0';

    public function init()
    {⋯
    }
}
```

The plugin directory - Craft CMS

Sometimes the namespace is *“agencyname\pluginname”*

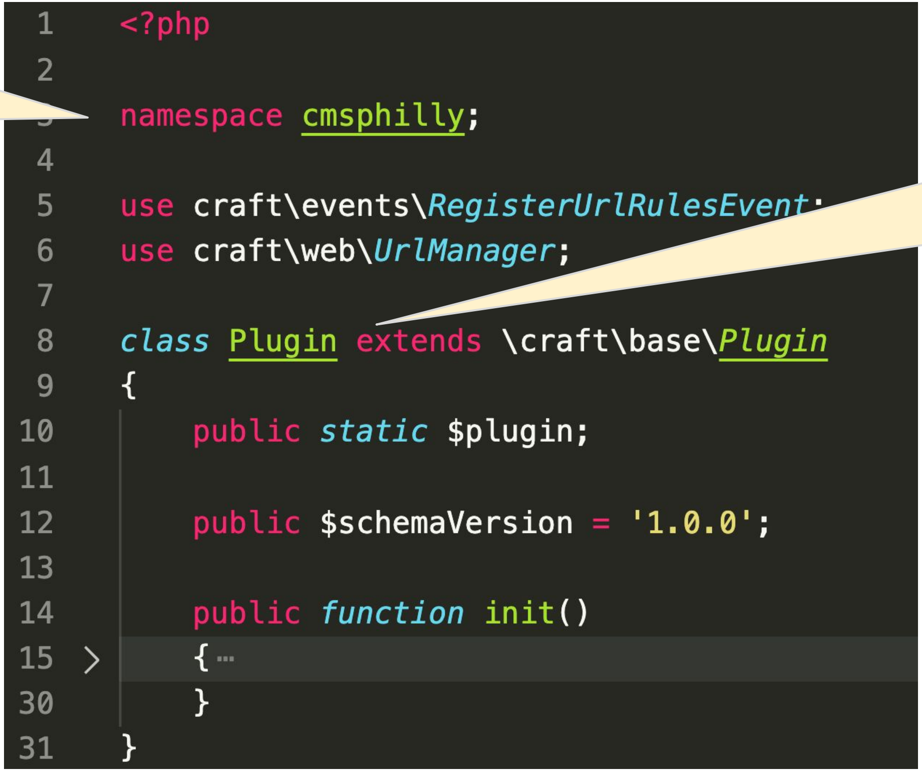
cmsphilly/src/Plugin.php



```php
<?php

namespace cmsphilly;

use craft\events\RegisterUrlRulesEvent;
use craft\web\UrlManager;

class Plugin extends \craft\base\Plugin
{
    public static $plugin;

    public $schemaVersion = '1.0.0';

    public function init()
    {…
    }
}
```

The plugin directory - Craft CMS

cmsphilly/src/Plugin.php



```php
<?php

namespace cmsphilly;

use craft\events\RegisterUrlRulesEvent;
use craft\web\UrlManager;

class Plugin extends \craft\base\Plugin
{
    public static $plugin;

    public $schemaVersion = '1.0.0';

    public function init()
    {…
    }
}
```

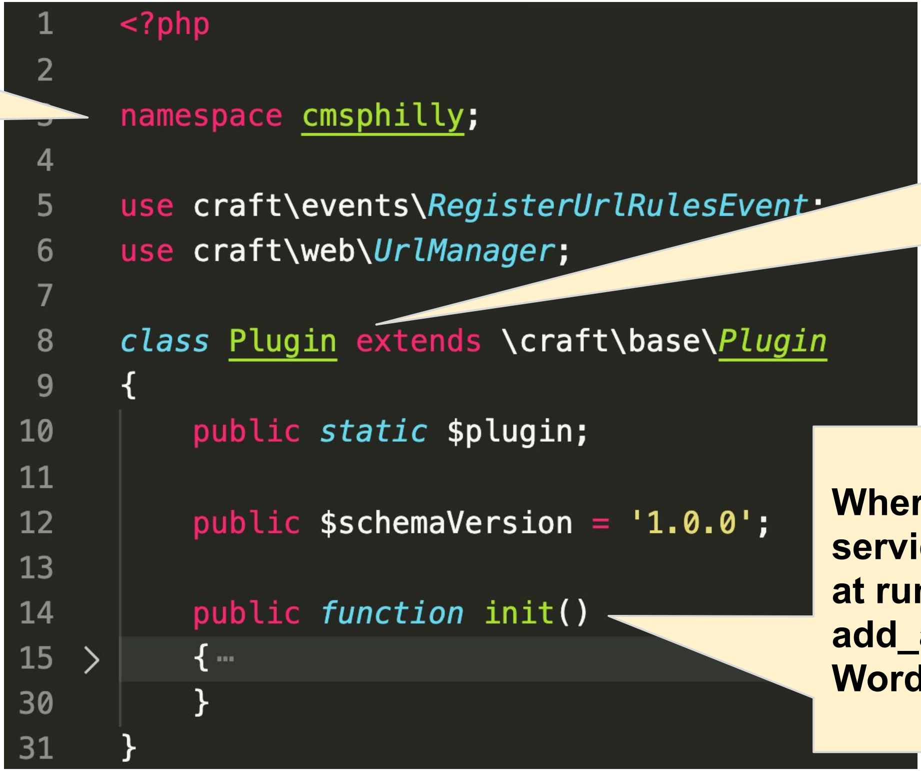
Sometimes the namespace is “*agencyname\pluginname”*

You can actually name your class anything, as long as the name matches the file name.

The plugin directory - Craft CMS

Sometimes the namespace is "*agencyname\pluginname*"

cmsphilly/src/Plugin.php



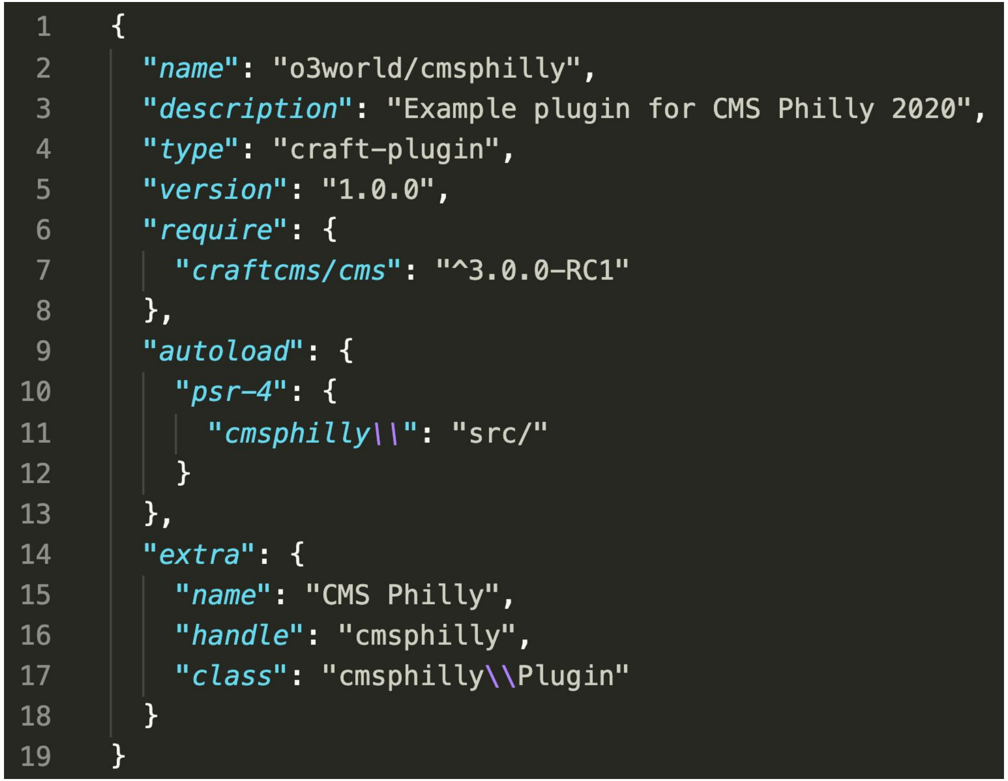
```php
<?php

namespace cmsphilly;

use craft\events\RegisterUrlRulesEvent;
use craft\web\UrlManager;

class Plugin extends \craft\base\Plugin
{
    public static $plugin;

    public $schemaVersion = '1.0.0';

    public function init()
    {⋯
    }
}
```

You can actually name your class anything, as long as the name matches the file name.

Where you register your services/event subscribers at runtime. (like add_action('init') in WordPress

The plugin directory - Craft CMS

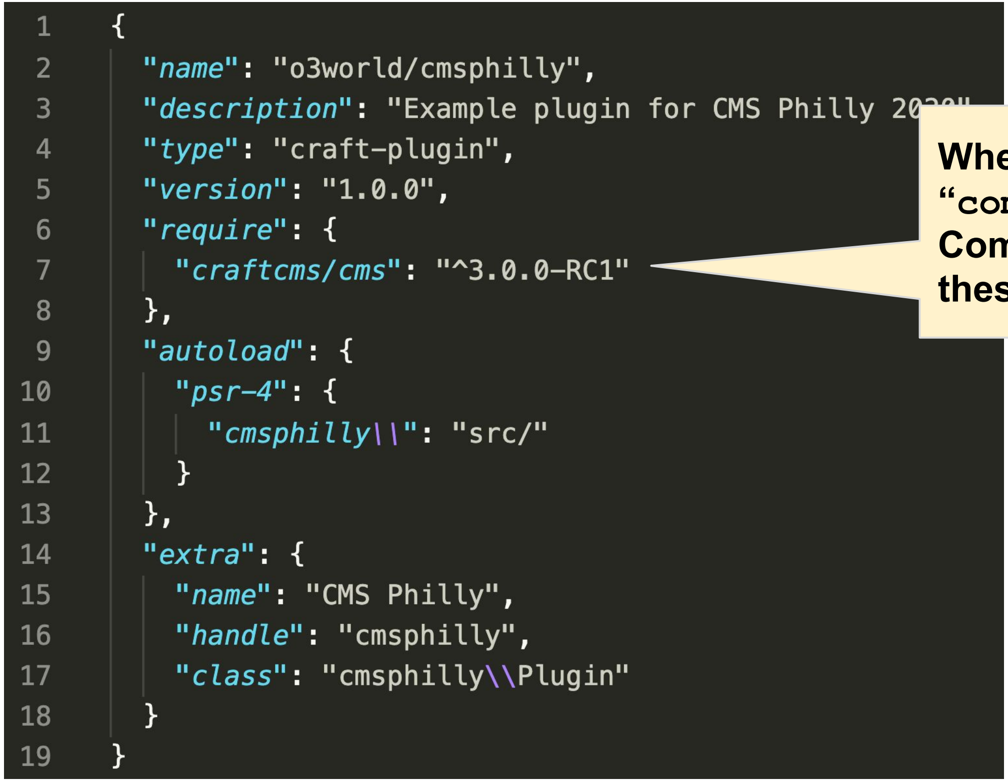
cmsphilly/composer.json



```
{
  "name": "o3world/cmsphilly",
  "description": "Example plugin for CMS Philly 2020",
  "type": "craft-plugin",
  "version": "1.0.0",
  "require": {
    "craftcms/cms": "^3.0.0-RC1"
  },
  "autoload": {
    "psr-4": {
      "cmsphilly\\": "src/"
    }
  },
  "extra": {
    "name": "CMS Philly",
    "handle": "cmsphilly",
    "class": "cmsphilly\\Plugin"
  }
}
```

The plugin directory - Craft CMS

`cmsphilly/composer.json`



```
{
  "name": "o3world/cmsphilly",
  "description": "Example plugin for CMS Philly 2020",
  "type": "craft-plugin",
  "version": "1.0.0",
  "require": {
    "craftcms/cms": "^3.0.0-RC1"
  },
  "autoload": {
    "psr-4": {
      "cmsphilly\\": "src/"
    }
  },
  "extra": {
    "name": "CMS Philly",
    "handle": "cmsphilly",
    "class": "cmsphilly\\Plugin"
  }
}
```

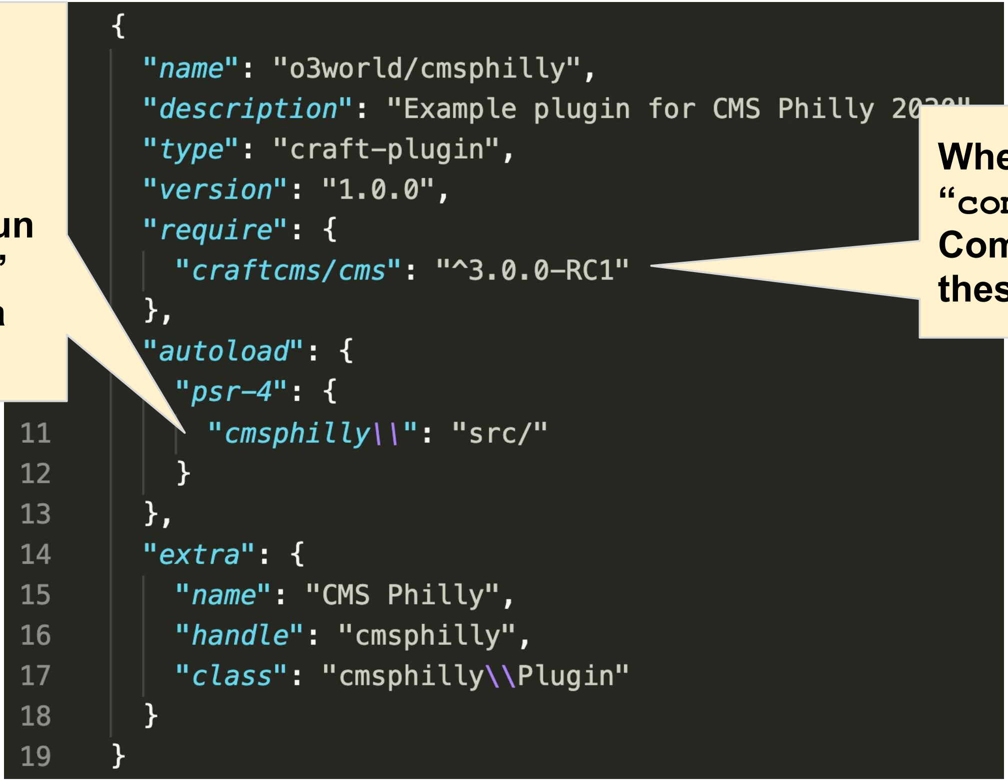
When you run “`composer install`”, Composer downloads these dependencies

The plugin directory - Craft CMS

cmsphilly/composer.json

Maps namespace to directory a la PSR-4 for **autoloading*.

Saves us from having to run “require ./ClassName.php” every time we instantiate a class



```
{
  "name": "o3world/cmsphilly",
  "description": "Example plugin for CMS Philly 2020"
  "type": "craft-plugin",
  "version": "1.0.0",
  "require": {
    "craftcms/cms": "^3.0.0-RC1"
  },
  "autoload": {
    "psr-4": {
      "cmsphilly\\": "src/"
    }
  },
  "extra": {
    "name": "CMS Philly",
    "handle": "cmsphilly",
    "class": "cmsphilly\\Plugin"
  }
}
```

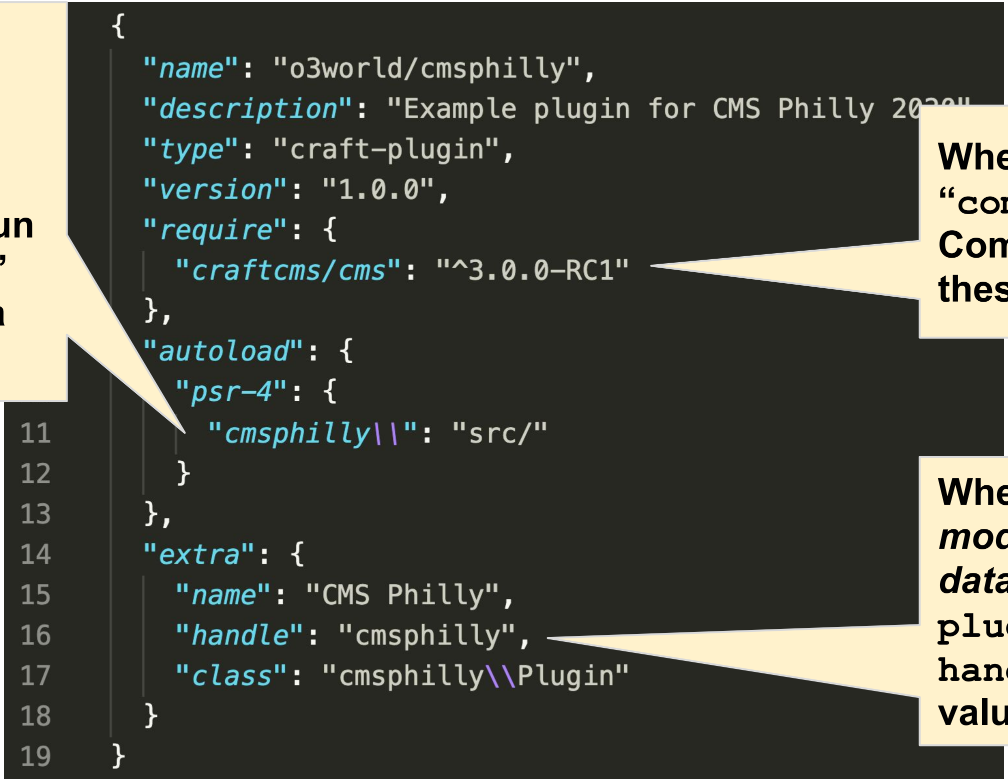
When you run “composer install”, Composer downloads these dependencies

The plugin directory - Craft CMS

`cmsphilly/composer.json`

Maps namespace to directory a la PSR-4 for **autoloading.**

Saves us from having to run "require ./ClassName.php" every time we instantiate a class



```
{
  "name": "o3world/cmsphilly",
  "description": "Example plugin for CMS Philly 2020",
  "type": "craft-plugin",
  "version": "1.0.0",
  "require": {
    "craftcms/cms": "^3.0.0-RC1"
  },
  "autoload": {
    "psr-4": {
      "cmsphilly\\": "src/"
    }
  },
  "extra": {
    "name": "CMS Philly",
    "handle": "cmsphilly",
    "class": "cmsphilly\\Plugin"
  }
}
```

When you run "`composer install`", Composer downloads these dependencies

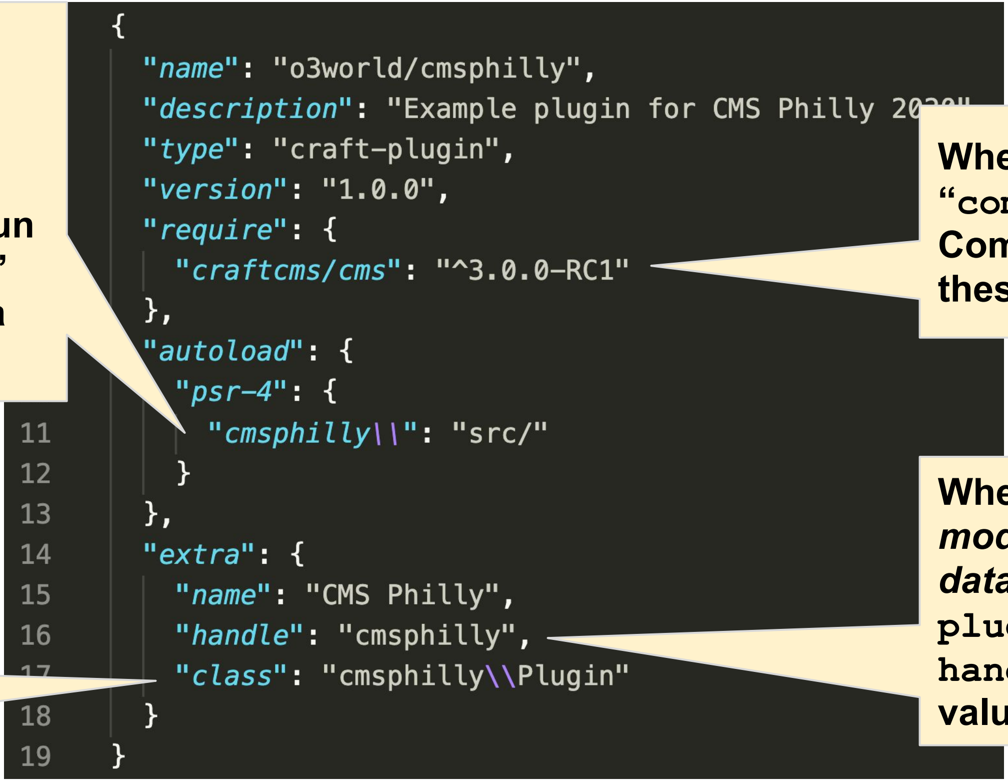
When you *install the module in the Craft database*, "`./craft plugin install handle`" looks for this value

The plugin directory - Craft CMS

`cmsphilly/composer.json`

Maps namespace to directory a la PSR-4 for *autoloading*.

Saves us from having to run “require ./ClassName.php” every time we instantiate a class



```
{
  "name": "o3world/cmsphilly",
  "description": "Example plugin for CMS Philly 2020",
  "type": "craft-plugin",
  "version": "1.0.0",
  "require": {
    "craftcms/cms": "^3.0.0-RC1"
  },
  "autoload": {
    "psr-4": {
      "cmsphilly\\": "src/"
    }
  },
  "extra": {
    "name": "CMS Philly",
    "handle": "cmsphilly",
    "class": "cmsphilly\\Plugin"
  }
}
```

When you run “`composer install`”, Composer downloads these dependencies

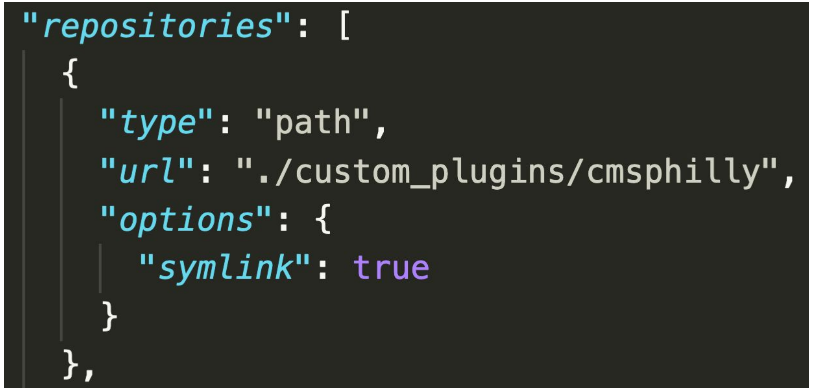
When you *install the module in the Craft database*, “`./craft plugin install handle`” looks for this value

Optional if your plugin file is named Plugin.php

Connecting a route to your controller

Tell Craft about your plugin!

- Add your plugin as a repository in your project root's `composer.json`
- Run `composer install`
- Run `./craft plugin/install plugin-handle`

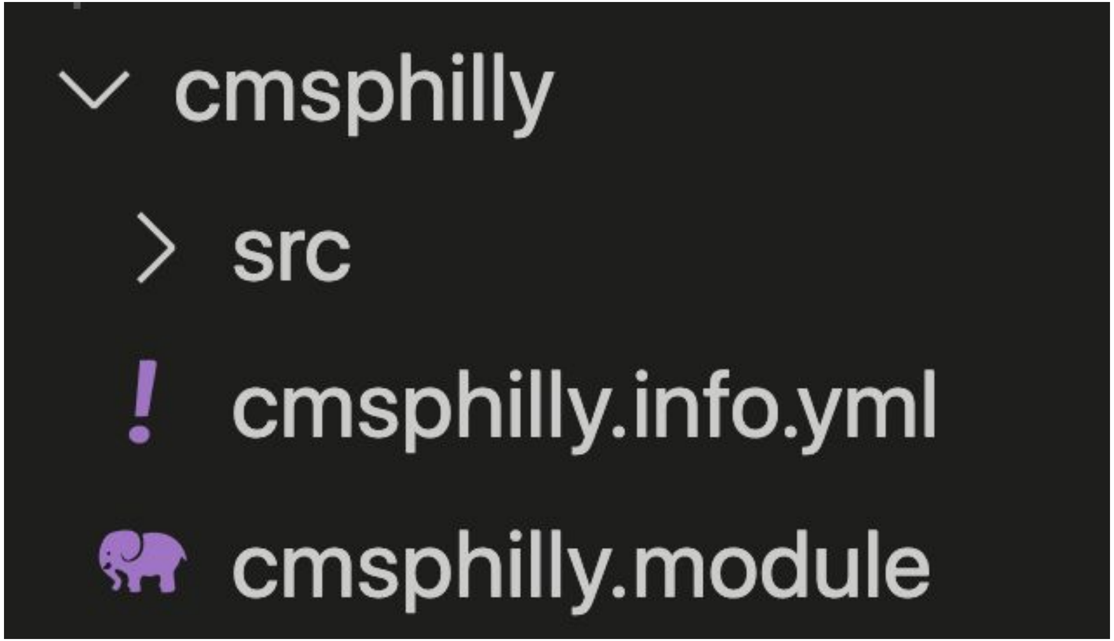


```
"repositories": [
  {
    "type": "path",
    "url": "./custom_plugins/cmsphilly",
    "options": {
      "symlink": true
    }
  },
```

The plugin (*MODULE*) directory - Drupal 8

The plugin (module) directory - Drupal 8

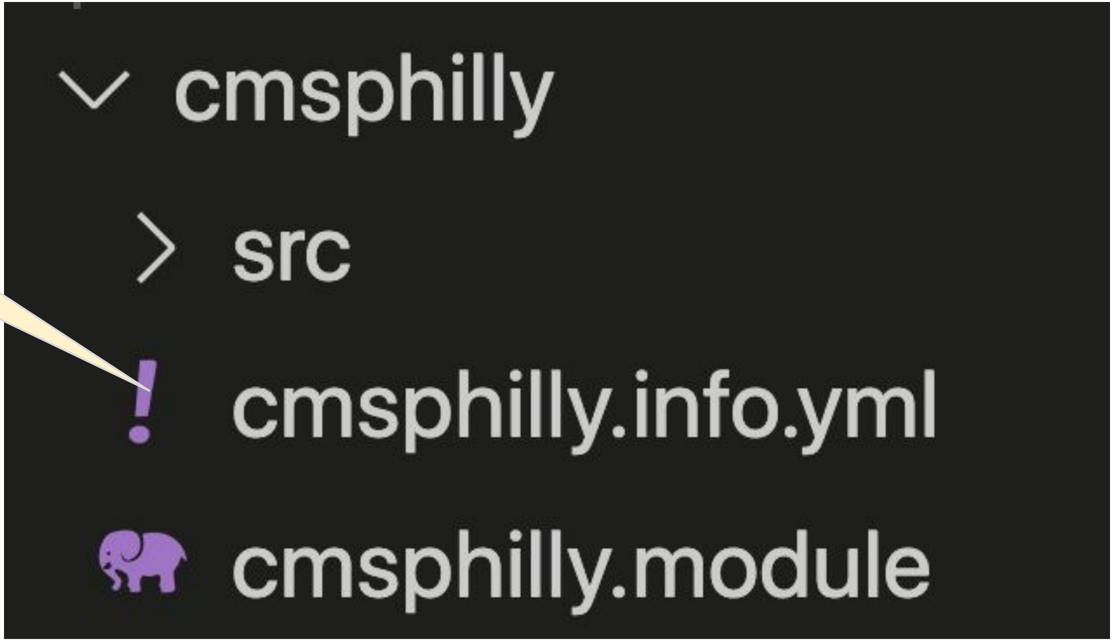
`modules/custom/cmsphilly`



```
cmsphilly
  src
  cmsphilly.info.yml
  cmsphilly.module
```

The plugin (module) directory - Drupal 8

`modules/custom/cmsphilly`



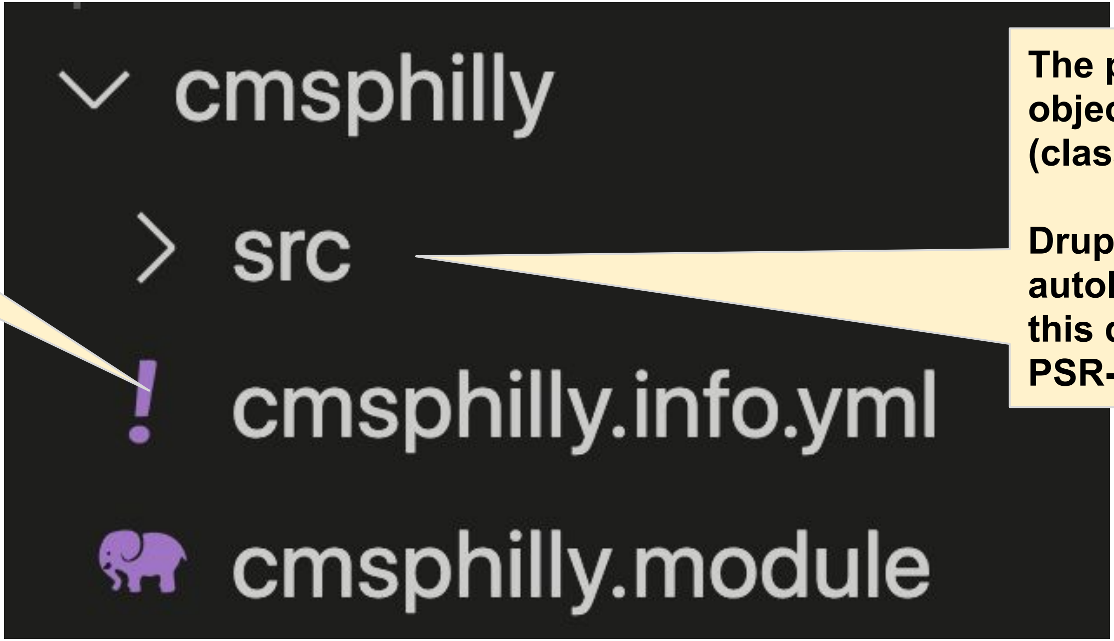
The only required file to install a Drupal 8 module.

```
cmsphilly
  src
  cmsphilly.info.yml
  cmsphilly.module
```

The plugin (module) directory - Drupal 8

`modules/custom/cmsphilly`

The only required file to install a Drupal 8 module.



```
∨ cmsphilly
  > src
  ! cmsphilly.info.yml
    cmsphilly.module
```

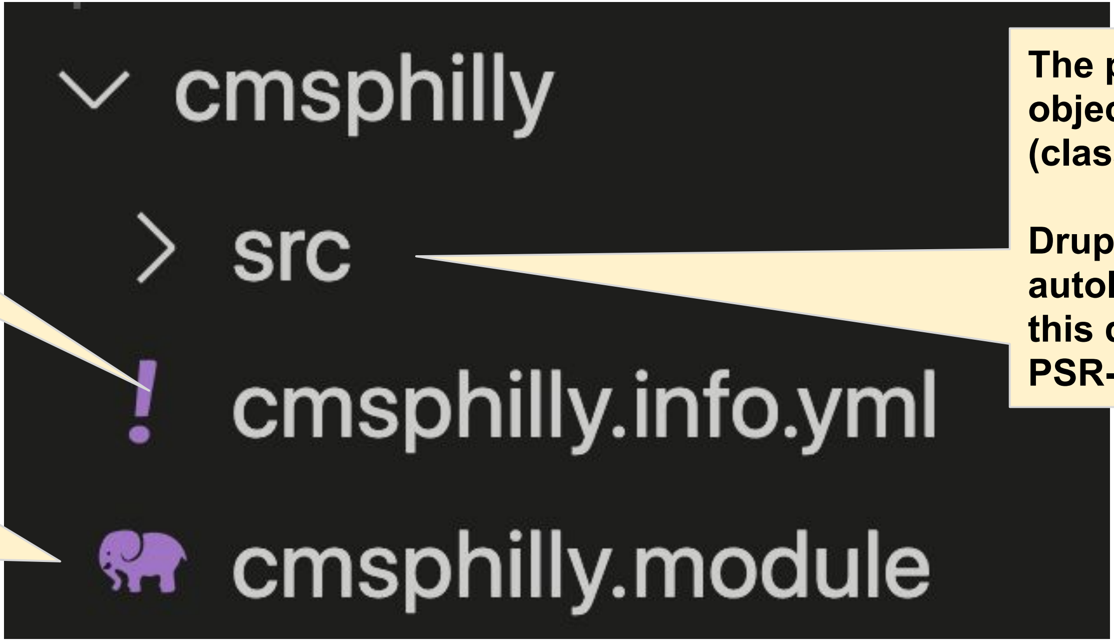
The place for object-oriented code (classes).

Drupal natively autoloads anything in this directory (also PSR-4)

The plugin (module) directory - Drupal 8

`modules/custom/cmsphilly`

The only required file to install a Drupal 8 module.



```
∨ cmsphilly
  > src
  ! cmsphilly.info.yml
    cmsphilly.module
```

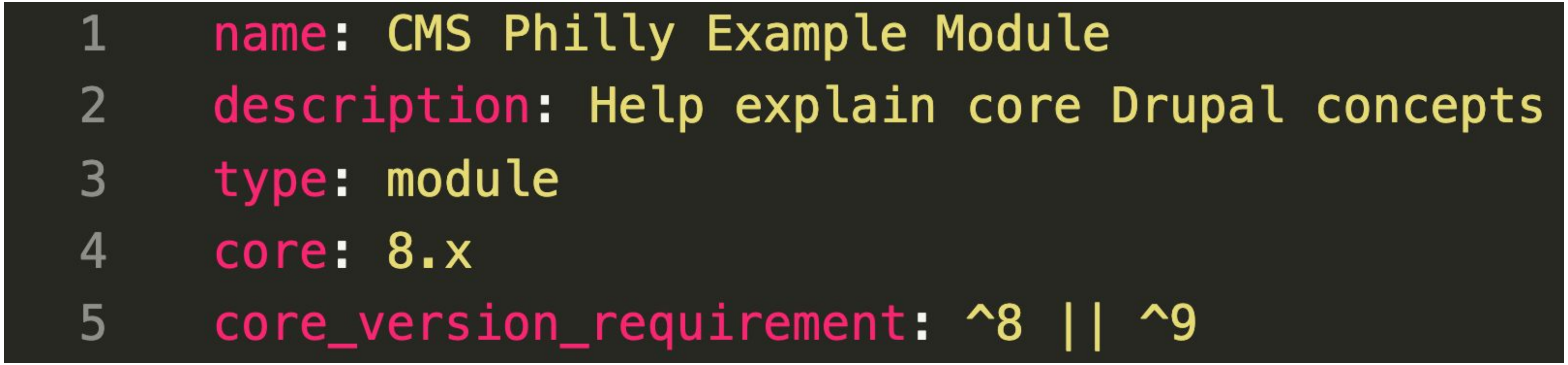
The place for object-oriented code (classes).

Drupal natively autoloads anything in this directory (also PSR-4)

The place for procedural hooks.

The plugin (module) directory - Drupal 8

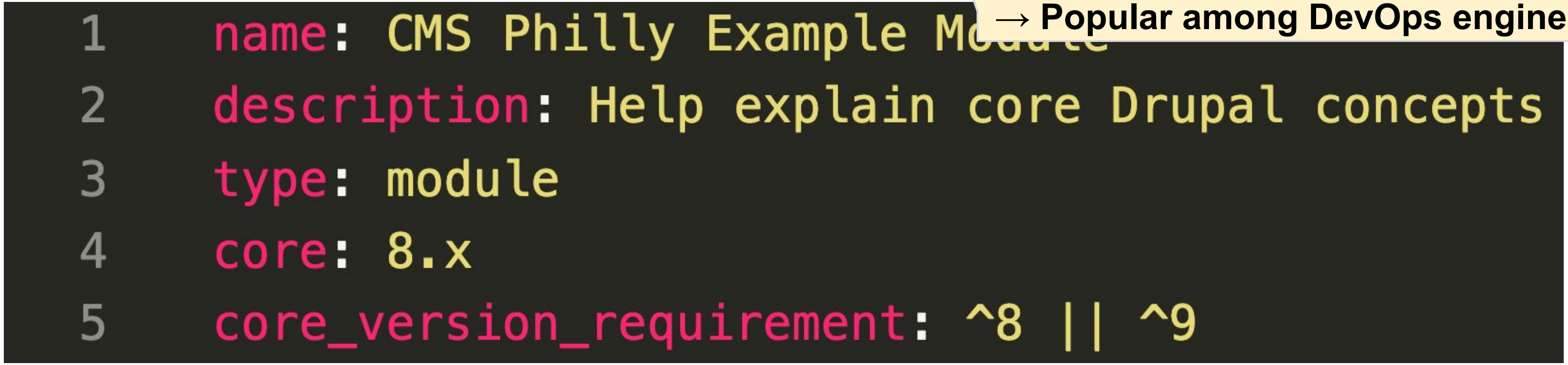
modules/custom/cmsphilly/cmsphilly.info.yml



```yaml
name: CMS Philly Example Module
description: Help explain core Drupal concepts
type: module
core: 8.x
core_version_requirement: ^8 || ^9
```

The plugin (module) directory - Drupal 8

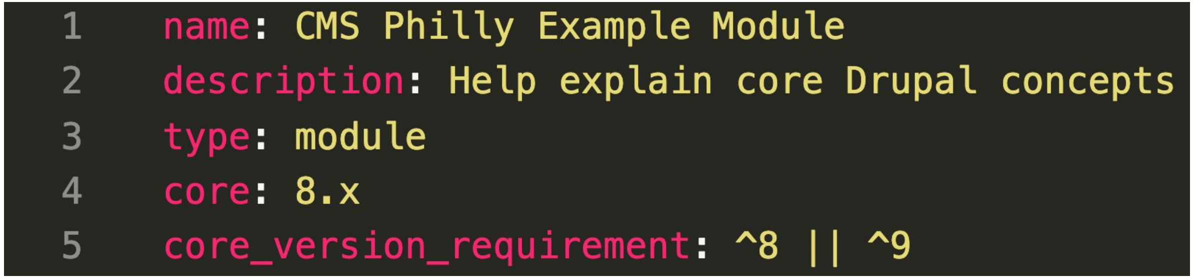
```
modules/custom/cmsphilly/cmsphilly.info.yml
```



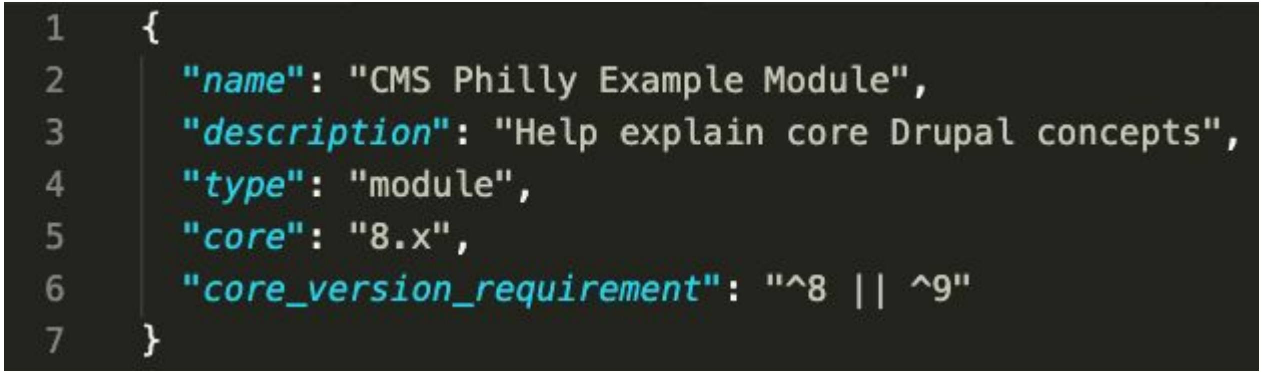
```yaml
name: CMS Philly Example Module
description: Help explain core Drupal concepts
type: module
core: 8.x
core_version_requirement: ^8 || ^9
```

YAML: YAML Ain’t Markup Language
→ A human-readable abstraction on JSON
→ Popular among DevOps engineers

YAML vs. JSON



```yaml
name: CMS Philly Example Module
description: Help explain core Drupal concepts
type: module
core: 8.x
core_version_requirement: ^8 || ^9
```

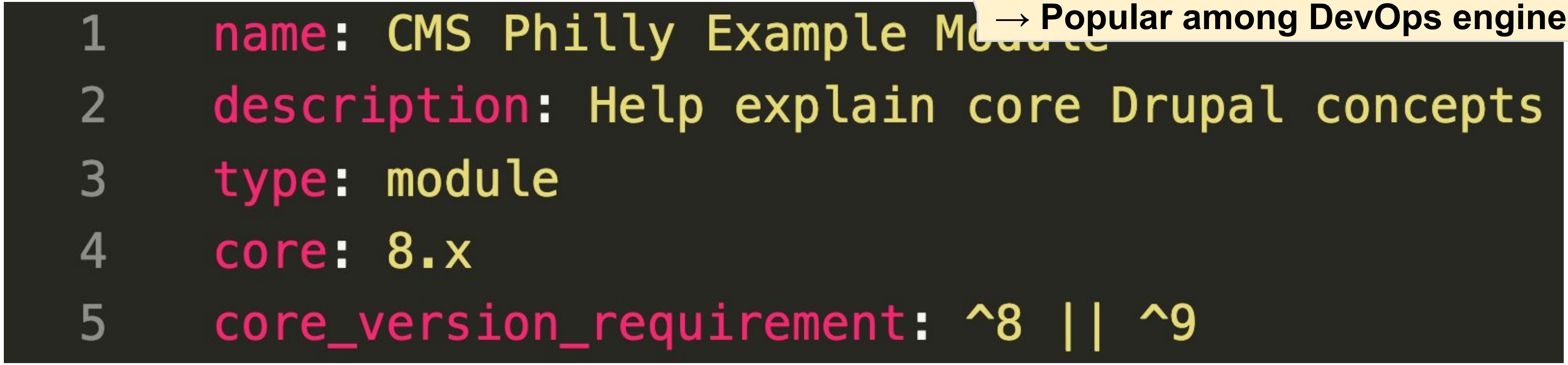


```json
{
  "name": "CMS Philly Example Module",
  "description": "Help explain core Drupal concepts",
  "type": "module",
  "core": "8.x",
  "core_version_requirement": "^8 || ^9"
}
```

The plugin (module) directory - Drupal 8

YAML: YAML Ain’t Markup Language
→ A human-readable abstraction on JSON
→ Popular among DevOps engineers

modules/custom/cmsphilly/cmsphilly.info.yml

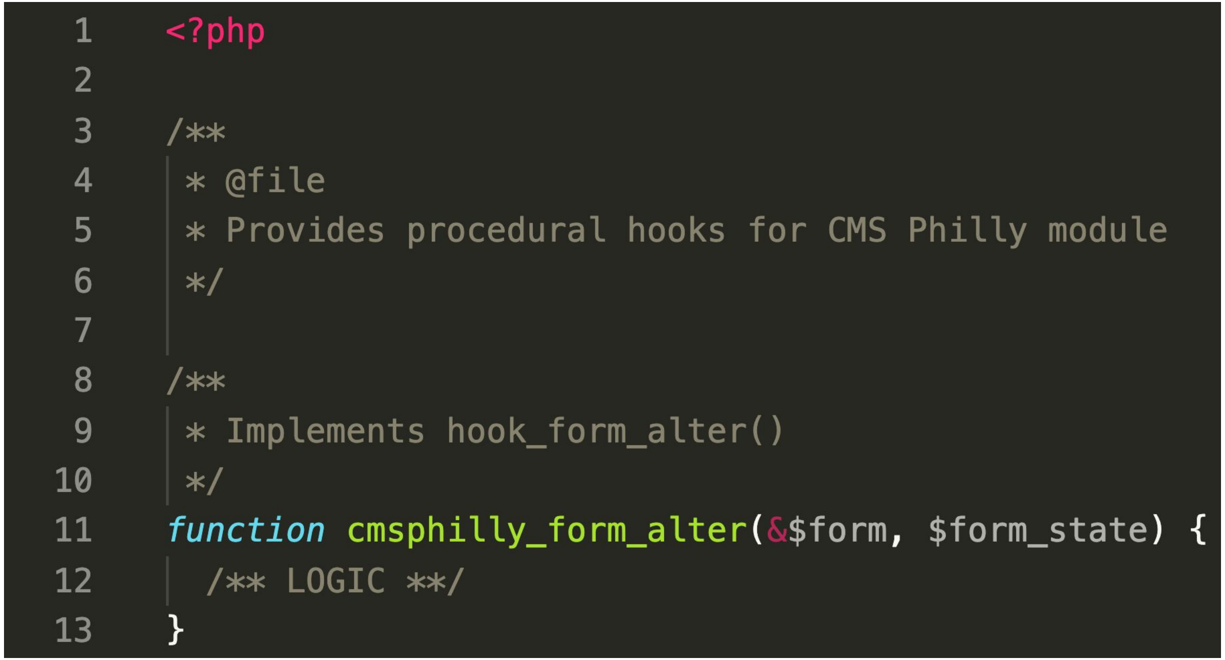


```yaml
name: CMS Philly Example Module
description: Help explain core Drupal concepts
type: module
core: 8.x
core_version_requirement: ^8 || ^9
```

New in Drupal core versions > 8.7.7 → Says that this module is compatible with all versions of Drupal 8 & 9

The plugin (module) directory - Drupal 8

`modules/custom/cmsphilly/cmsphilly.module`

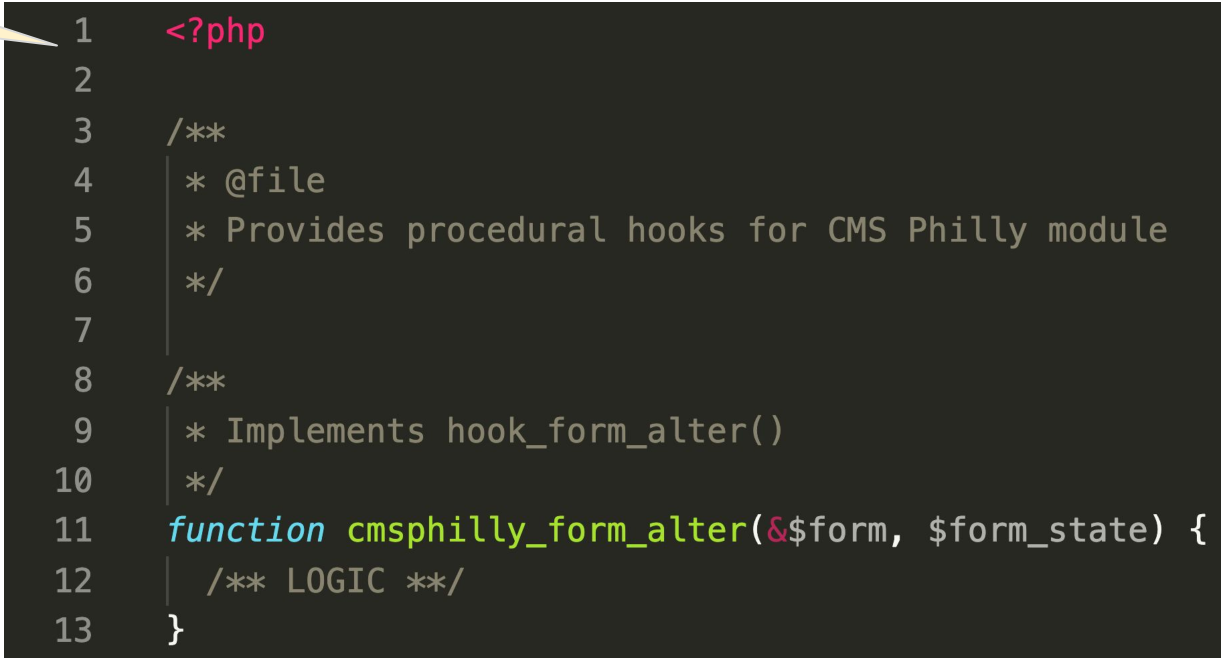


```php
<?php

/**
 * @file
 * Provides procedural hooks for CMS Philly module
 */

/**
 * Implements hook_form_alter()
 */
function cmsphilly_form_alter(&$form, $form_state) {
  /** LOGIC **/
}
```

The plugin (module) directory - Drupal 8

No rules for this file.

`modules/custom/cmsphilly/cmsphilly.module`

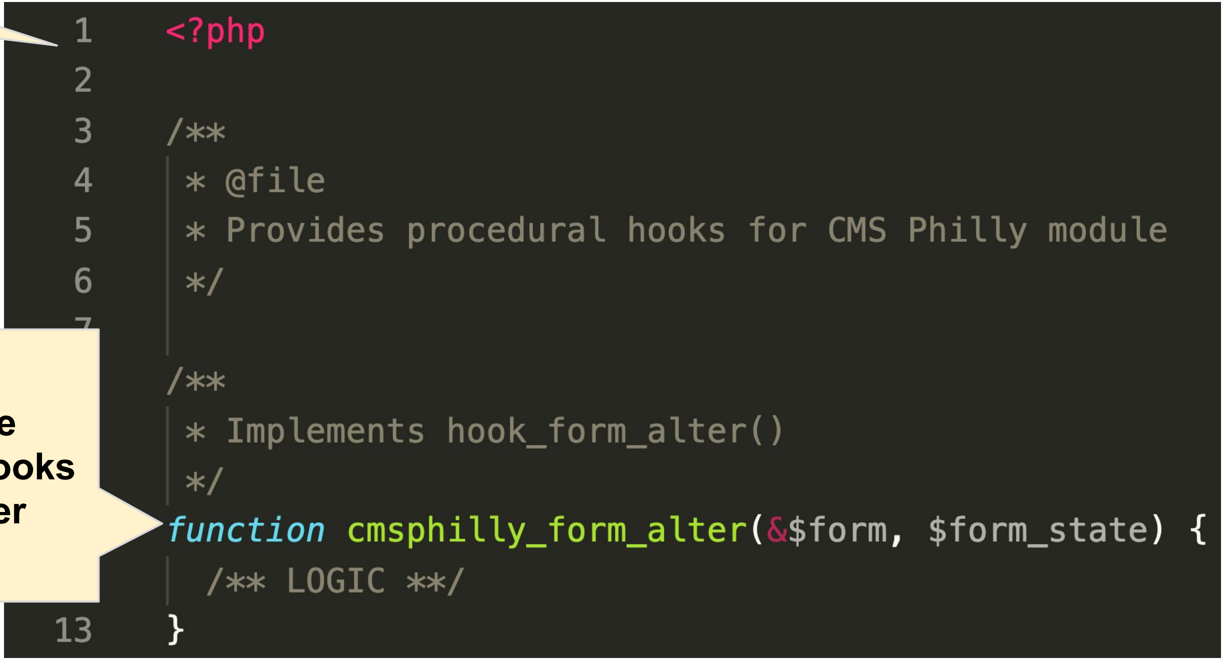


```php
<?php

/**
 * @file
 * Provides procedural hooks for CMS Philly module
 */

/**
 * Implements hook_form_alter()
 */
function cmsphilly_form_alter(&$form, $form_state) {
  /** LOGIC **/
}
```

The plugin (module) directory - Drupal 8

No rules for this file.

`modules/custom/cmsphilly/cmsphilly.module`



```php
<?php

/**
 * @file
 * Provides procedural hooks for CMS Philly module
 */

/**
 * Implements hook_form_alter()
 */
function cmsphilly_form_alter(&$form, $form_state) {
  /** LOGIC **/
}
```

As an example, here’s one of the most popular hooks in Drupal, to alter forms

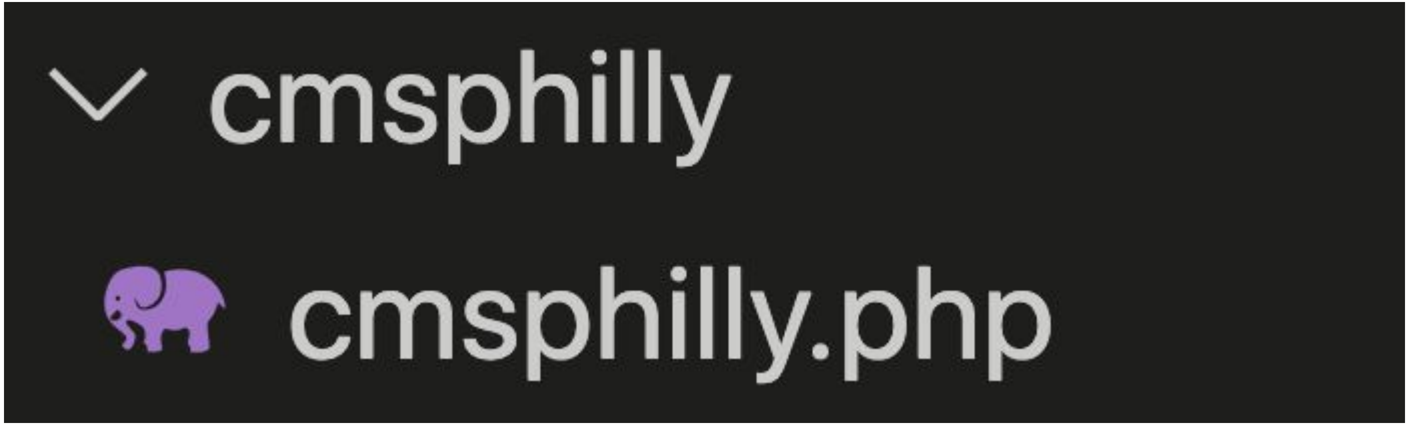
The plugin (module) directory - Drupal 8

Tell Drupal about your module!

- Run `drush en modulename`
  (or install manually in the admin)

The plugin directory - WordPress

`wp-content/plugins/cmsphilly`

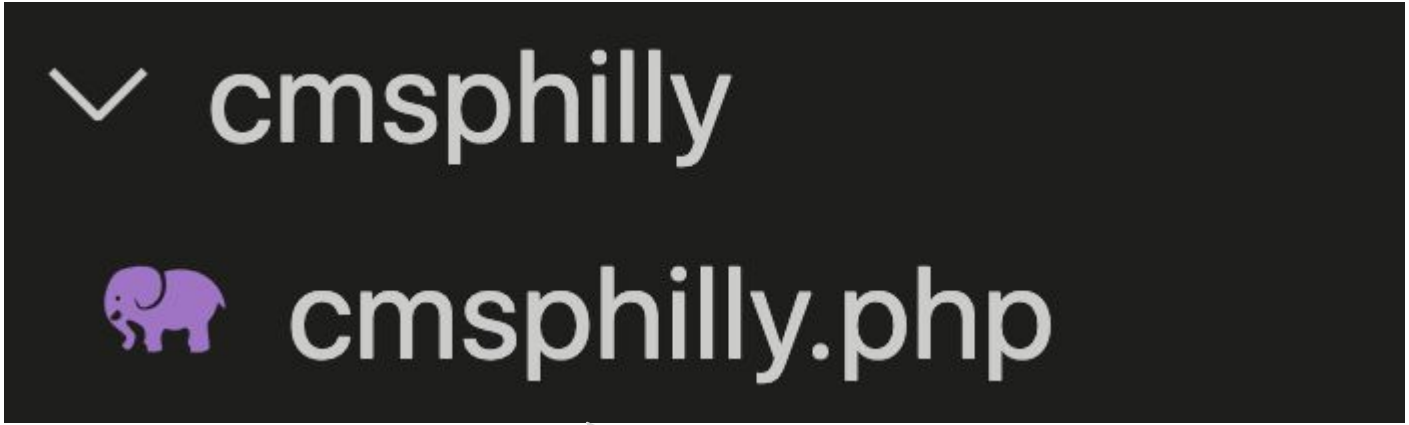


cmsphilly

cmsphilly.php

The plugin directory - WordPress

`wp-content/plugins/cmsphilly`

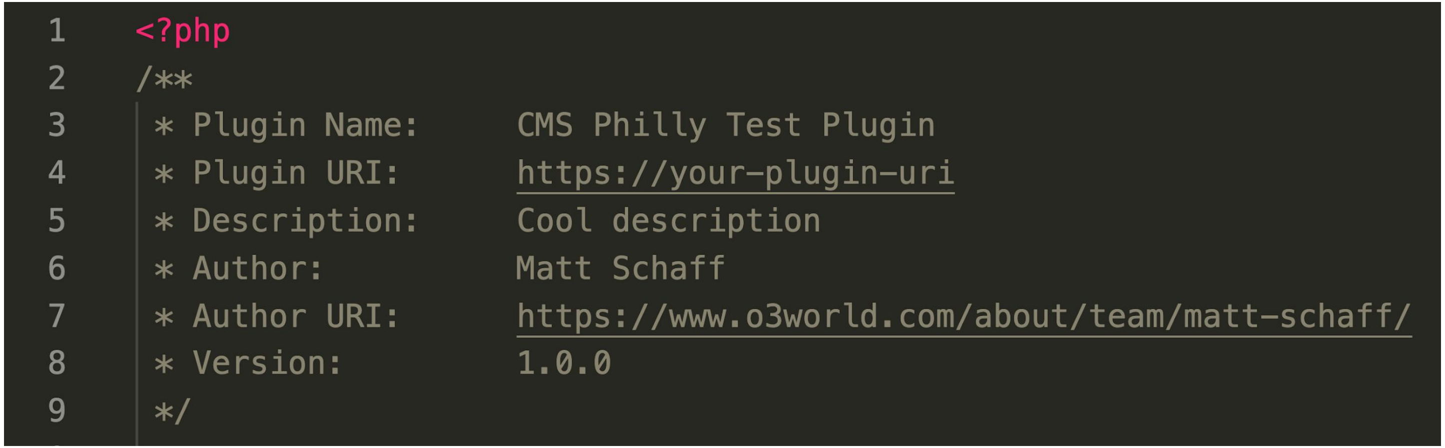


```
cmsphilly
  cmsphilly.php
```

WordPress plugin development is the Wild West

The plugin directory - WordPress

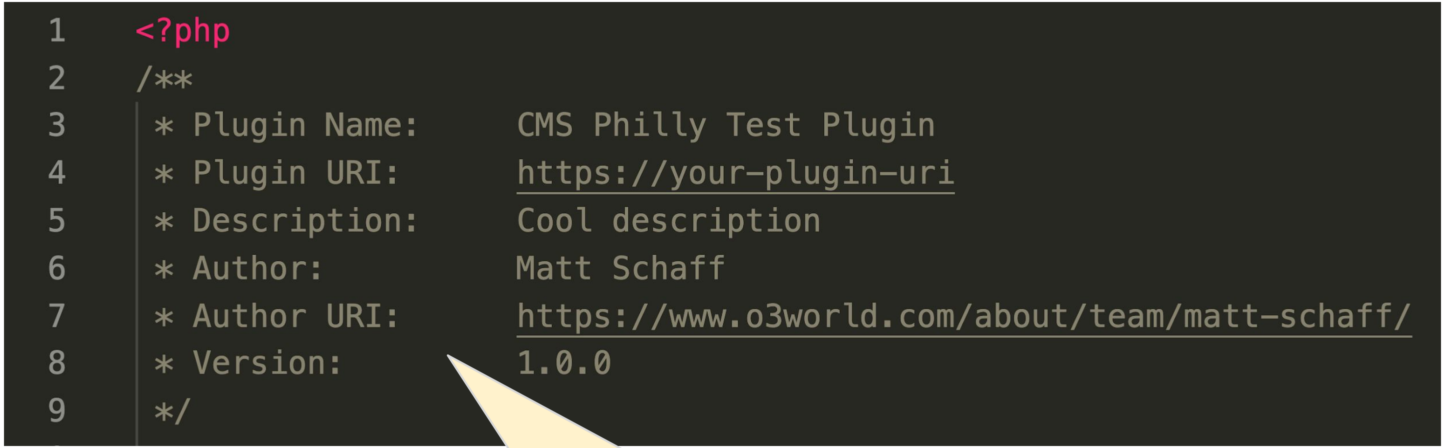
wp-content/plugins/cmsphilly



```
<?php
/**
 * Plugin Name:       CMS Philly Test Plugin
 * Plugin URI:        https://your-plugin-uri
 * Description:       Cool description
 * Author:            Matt Schaff
 * Author URI:        https://www.o3world.com/about/team/matt-schaff/
 * Version:           1.0.0
 */
```

The plugin directory - WordPress

`wp-content/plugins/cmsphilly/cmsphilly.php`



```php
<?php
/**
 * Plugin Name:     CMS Philly Test Plugin
 * Plugin URI:      https://your-plugin-uri
 * Description:     Cool description
 * Author:          Matt Schaff
 * Author URI:      https://www.o3world.com/about/team/matt-schaff/
 * Version:         1.0.0
 */
```

If you’ve filled out the above docblock, you an find & install your plugin in the WP Admin dashboard

PROGRAMMING
BUILDING BLOCKS

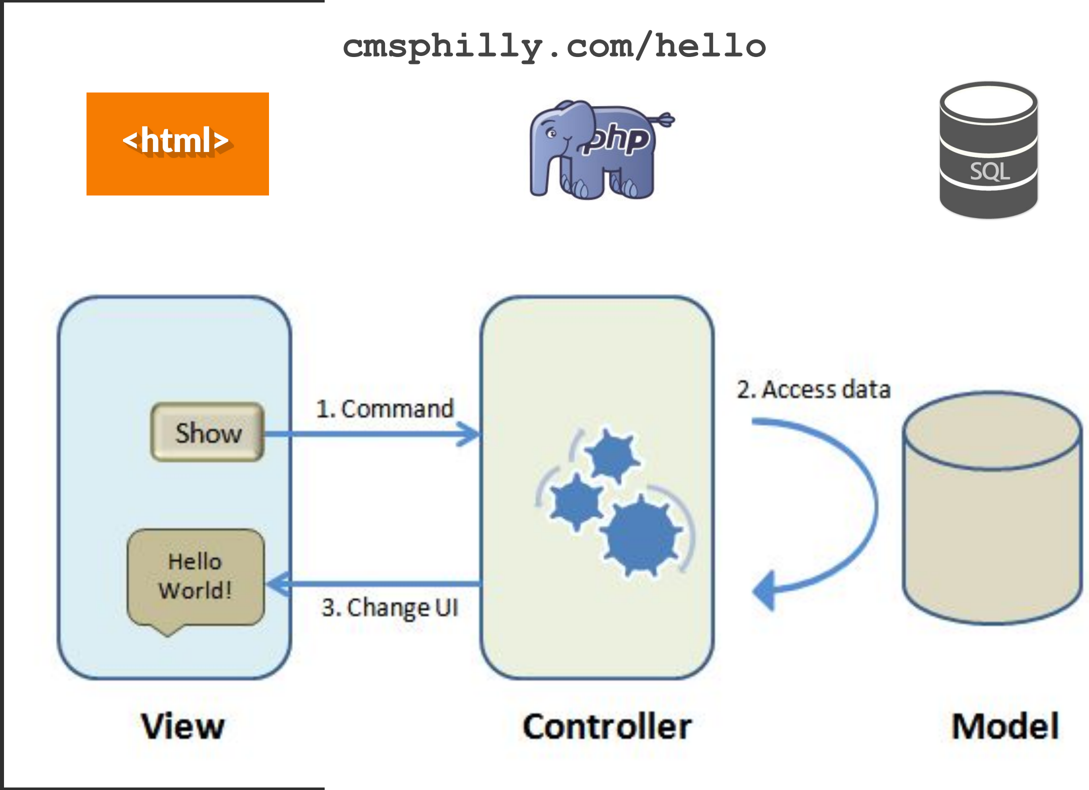
“How can I get to ‘Hello World’ in each CMS?”

“How can I get to ‘Hello World’ in each CMS?”

Answer: You need a **route**, and the route’s **controller**

PROGRAMMING
BUILDING BLOCKS

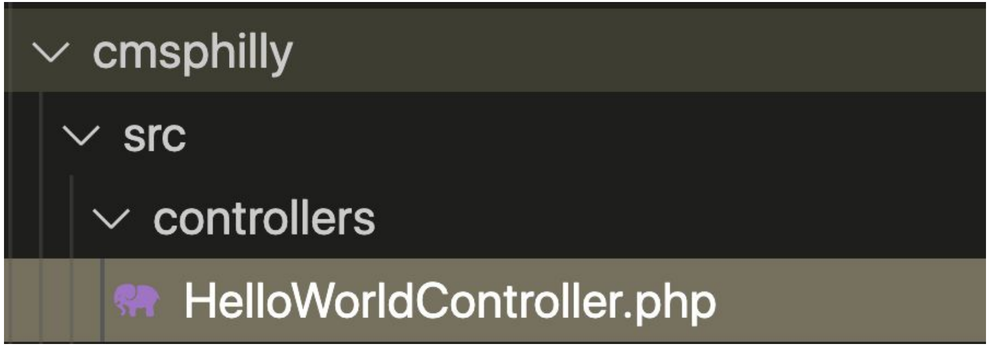
Routing - Controllers



Controllers | Craft CMS

How to create a route with a Controller

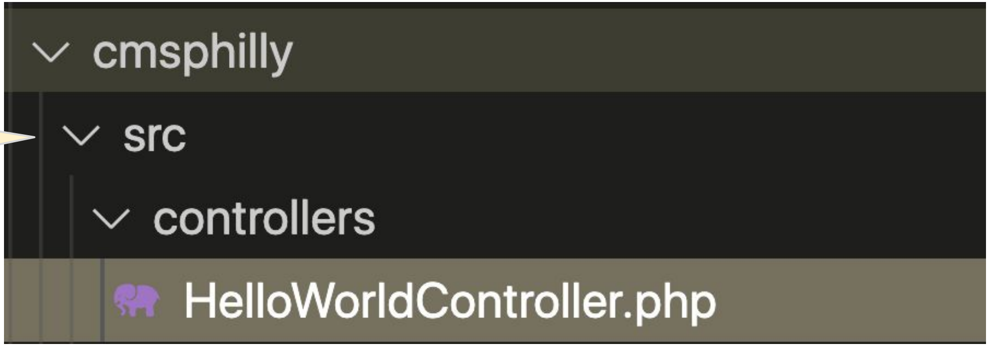
Where to put the Controller file:



```
cmsphilly
  src
    controllers
      HelloWorldController.php
```

Where to put the Controller file:

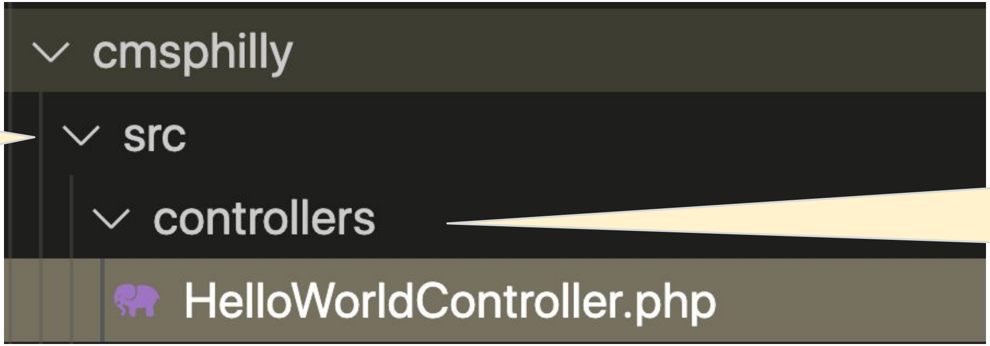
All PHP is in the “src” directory



```
cmsphilly
  src
    controllers
      HelloWorldController.php
```

Where to put the Controller file:

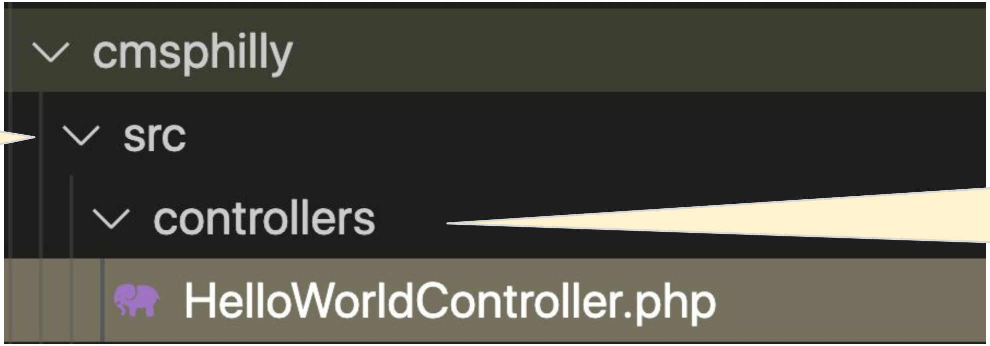
All PHP is in the “src” directory



```
∨ cmsphilly
  ∨ src
    ∨ controllers
        HelloWorldController.php
```

Controllers go in the “controllers” directory

Where to put the Controller file:



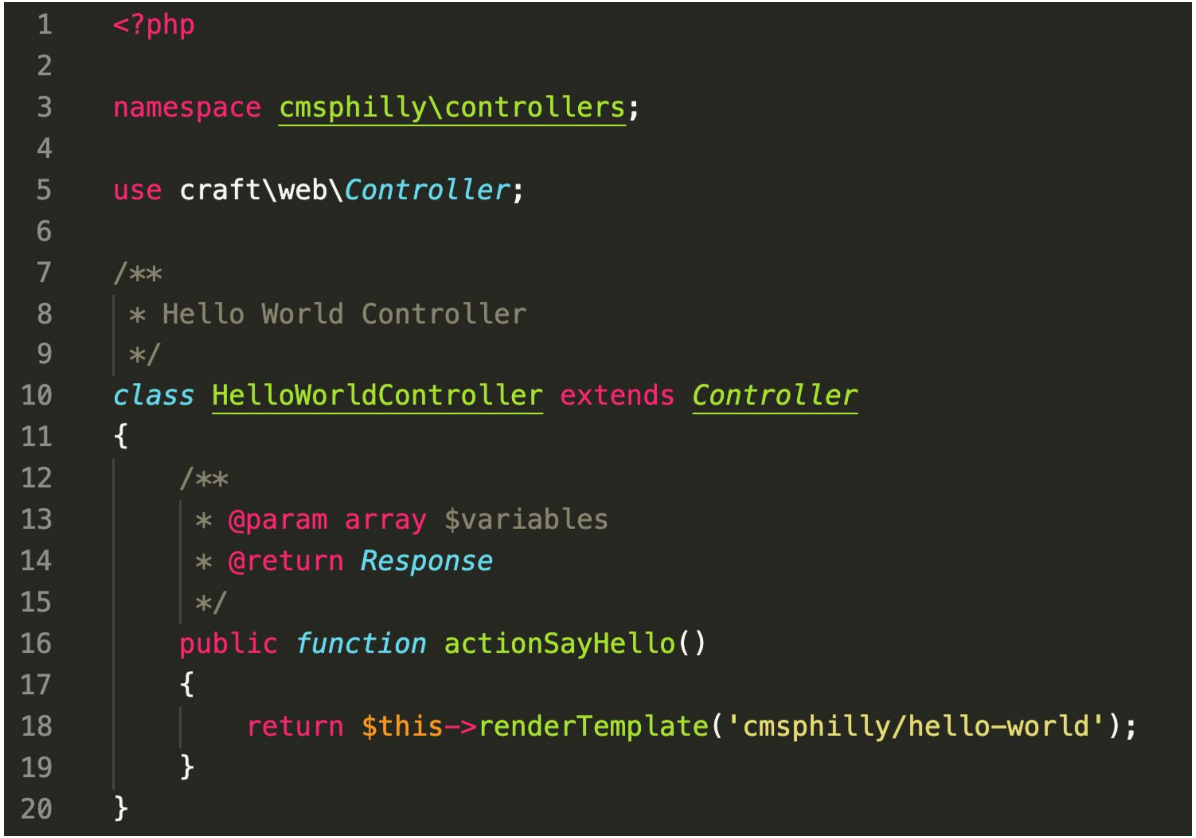
```
cmsphilly
  src
    controllers
      HelloWorldController.php
```

All PHP is in the “src” directory

Controllers go in the “controllers” directory

Must end in “Controller.php”

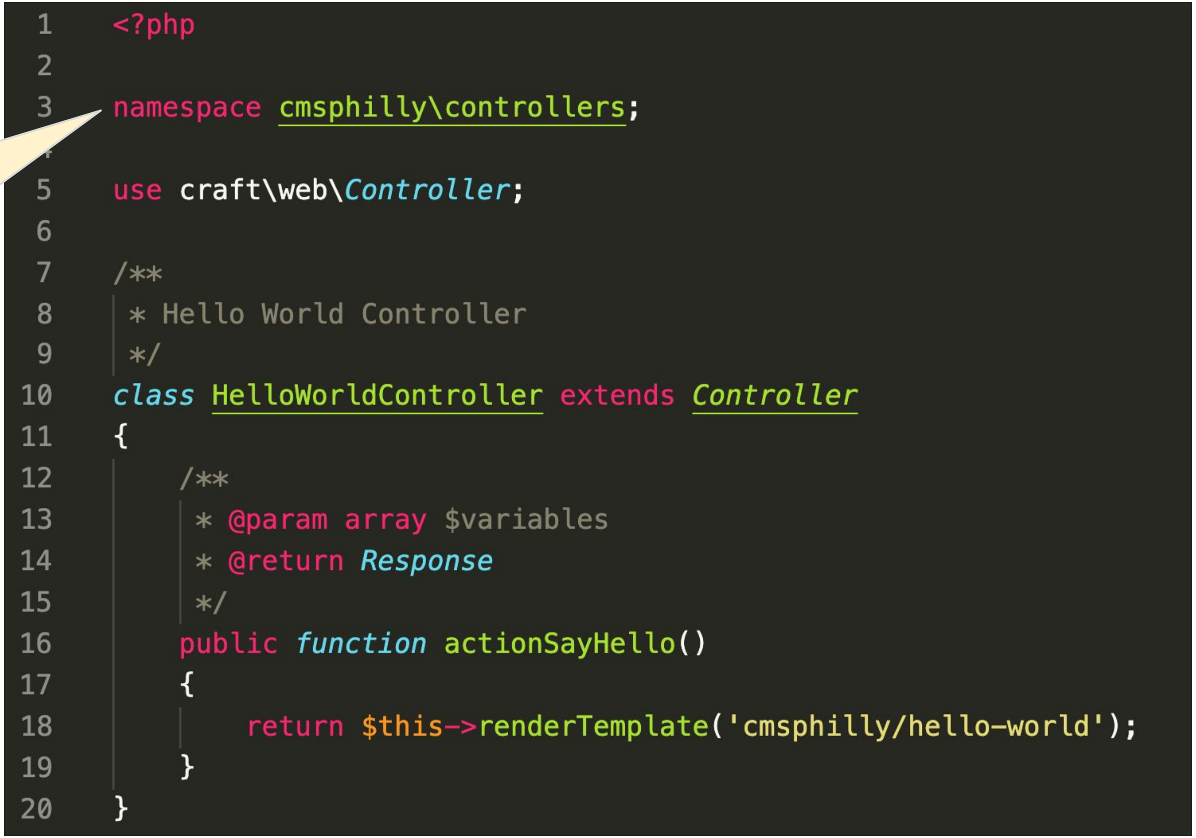
How to create a route with a Controller



```php
<?php

namespace cmsphilly\controllers;

use craft\web\Controller;

/**
 * Hello World Controller
 */
class HelloWorldController extends Controller
{
    /**
     * @param array $variables
     * @return Response
     */
    public function actionSayHello()
    {
        return $this->renderTemplate('cmsphilly/hello-world');
    }
}
```

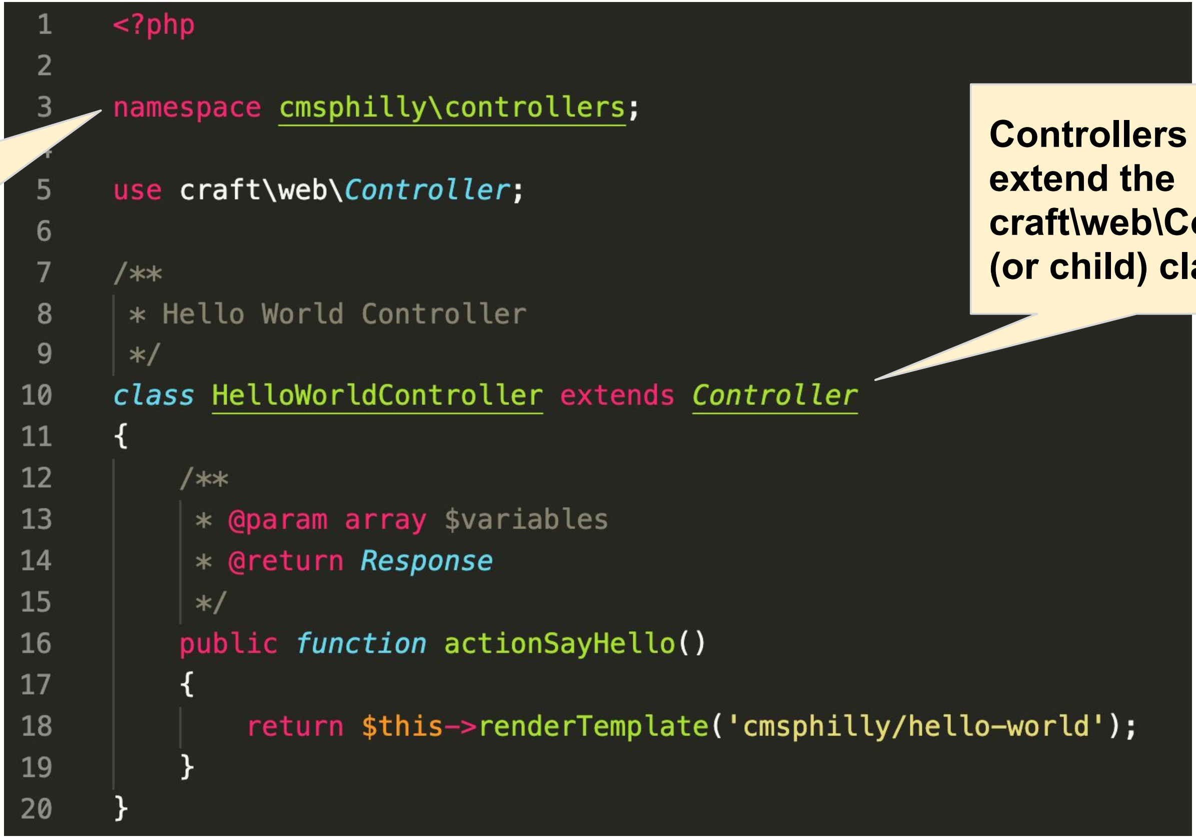
How to create a route with a Controller

Namespaces shorten file names



```php
<?php

namespace cmsphilly\controllers;

use craft\web\Controller;

/**
 * Hello World Controller
 */
class HelloWorldController extends Controller
{
    /**
     * @param array $variables
     * @return Response
     */
    public function actionSayHello()
    {
        return $this->renderTemplate('cmsphilly/hello-world');
    }
}
```

How to create a route with a Controller

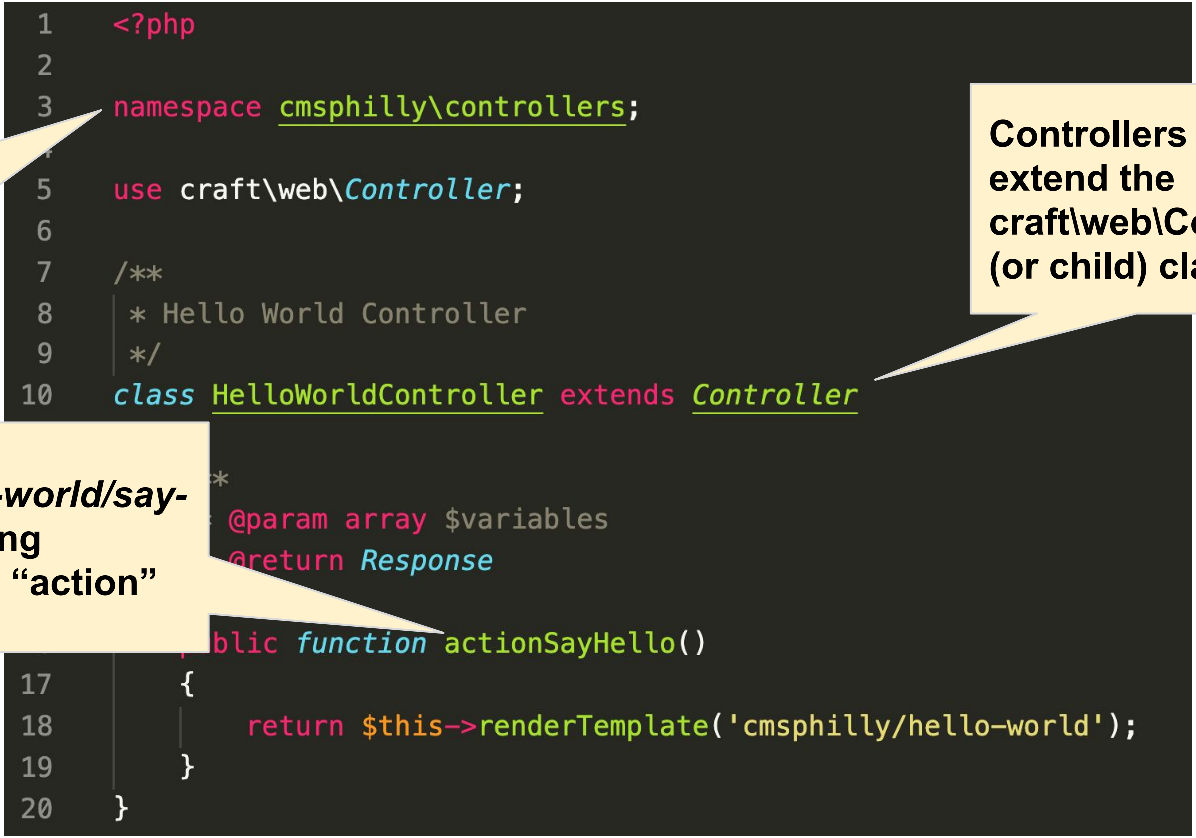


```php
<?php

namespace cmsphilly\controllers;

use craft\web\Controller;

/**
 * Hello World Controller
 */
class HelloWorldController extends Controller
{
    /**
     * @param array $variables
     * @return Response
     */
    public function actionSayHello()
    {
        return $this->renderTemplate('cmsphilly/hello-world');
    }
}
```

Namespaces shorten file names

Controllers must extend the craft\web\Controller (or child) class

How to create a route with a Controller



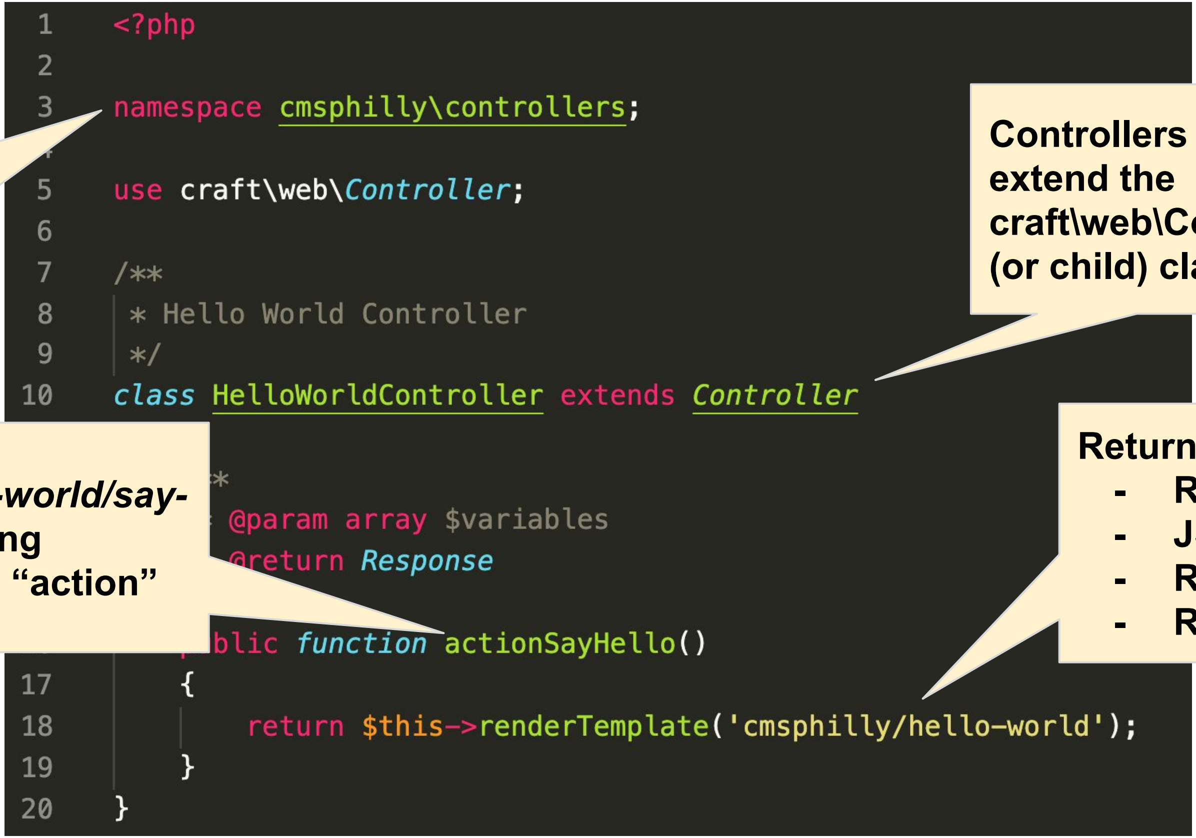
```
<?php

namespace cmsphilly\controllers;

use craft\web\Controller;

/**
 * Hello World Controller
 */
class HelloWorldController extends Controller

     **
     * @param array $variables
     @return Response

     blic function actionSayHello()
    {
        return $this->renderTemplate('cmsphilly/hello-world');
    }
}
```

Namespaces shorten file names

Controllers must extend the craft\web\Controller (or child) class

The “*cmsphilly/hello-world/say-hello*” action string translates to this “action” function

How to create a route with a Controller



```
<?php

namespace cmsphilly\controllers;

use craft\web\Controller;

/**
 * Hello World Controller
 */
class HelloWorldController extends Controller

         *
       @param array $variables
       @return Response

     blic function actionSayHello()
    {
        return $this->renderTemplate('cmsphilly/hello-world');
    }
}
```

Namespaces shorten file names

Controllers must extend the craft\web\Controller (or child) class

The “*cmsphilly/hello-world/say-hello*” action string translates to this “action” function

Return a view:

- Rendered template
- JSON
- Raw text
- Redirection

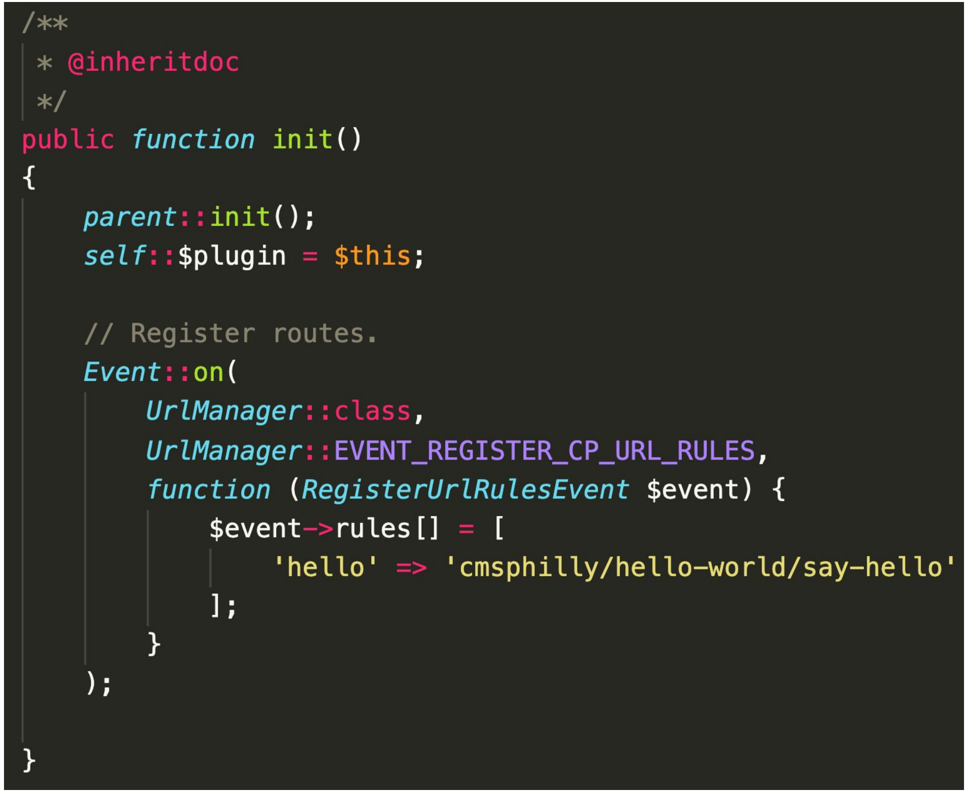
Connecting a route to your controller

Now, tell Craft about your controller!

Connecting a route to your controller

Now, tell Craft about your controller!

cmsphilly/src/Plugin.php



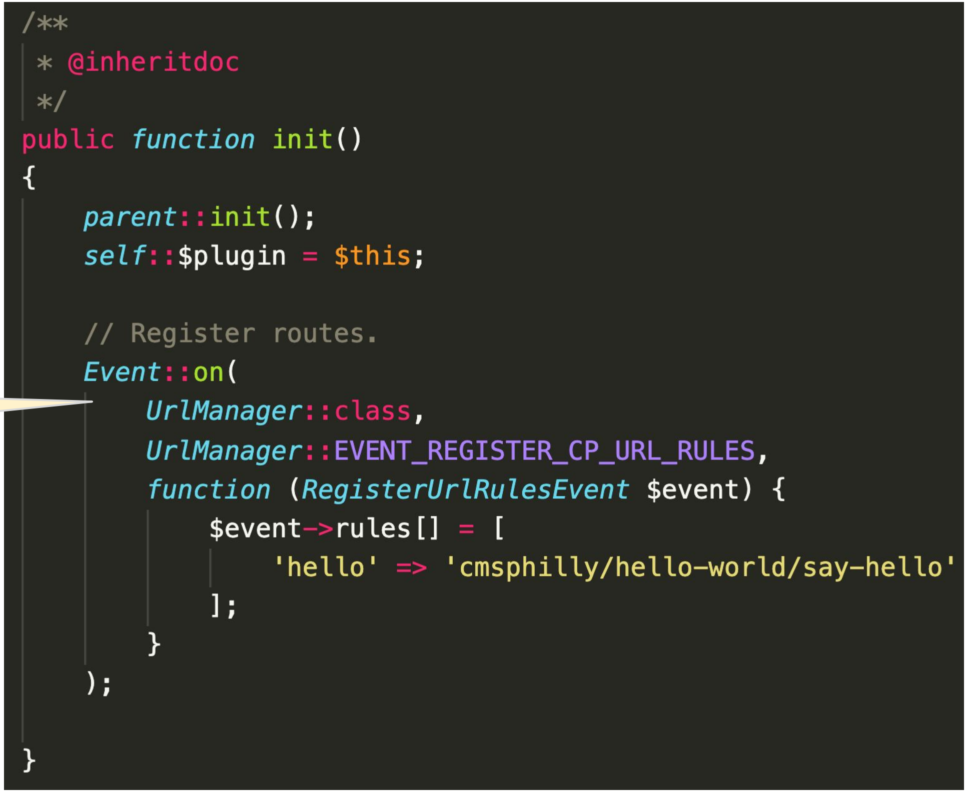
```php
/**
 * @inheritdoc
 */
public function init()
{
    parent::init();
    self::$plugin = $this;

    // Register routes.
    Event::on(
        UrlManager::class,
        UrlManager::EVENT_REGISTER_CP_URL_RULES,
        function (RegisterUrlRulesEvent $event) {
            $event->rules[] = [
                'hello' => 'cmsphilly/hello-world/say-hello'
            ];
        }
    );

}
```

Connecting a route to your controller

Now, tell Craft about your controller!

Subscribe to the route subscription event!

`cmsphilly/src/Plugin.php`



```php
/**
 * @inheritdoc
 */
public function init()
{
    parent::init();
    self::$plugin = $this;

    // Register routes.
    Event::on(
        UrlManager::class,
        UrlManager::EVENT_REGISTER_CP_URL_RULES,
        function (RegisterUrlRulesEvent $event) {
            $event->rules[] = [
                'hello' => 'cmsphilly/hello-world/say-hello'
            ];
        }
    );

}
```

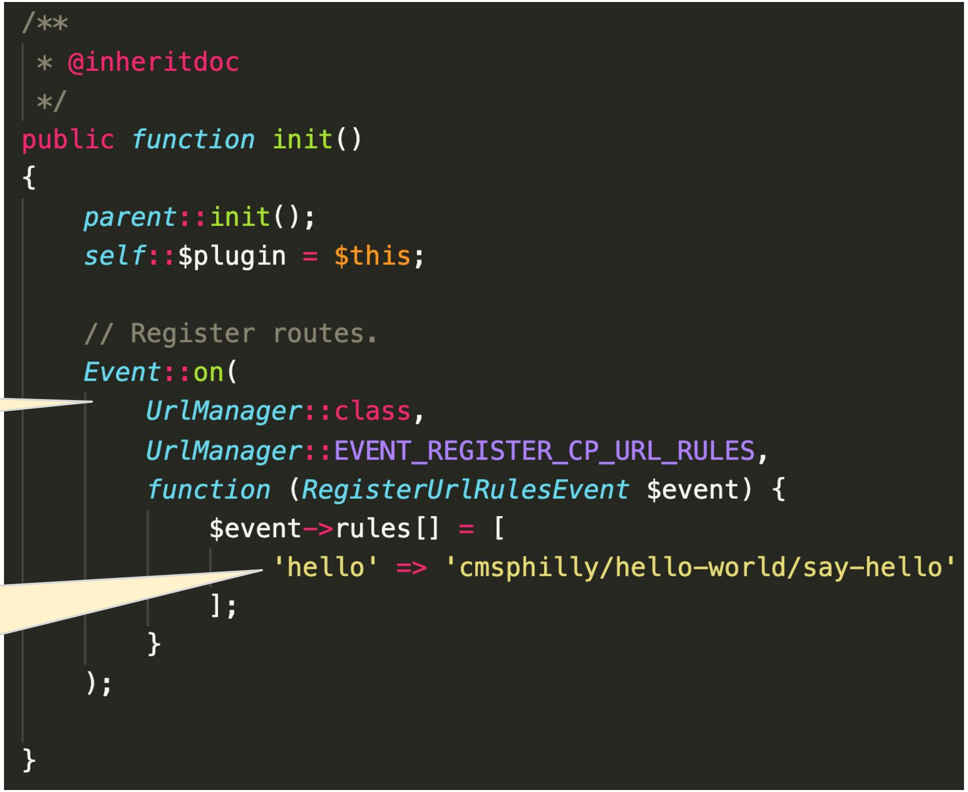
Connecting a route to your controller

Now, tell Craft about your controller!

Subscribe to the route subscription event!

The route “hello” is now *controlled* by your Controller!

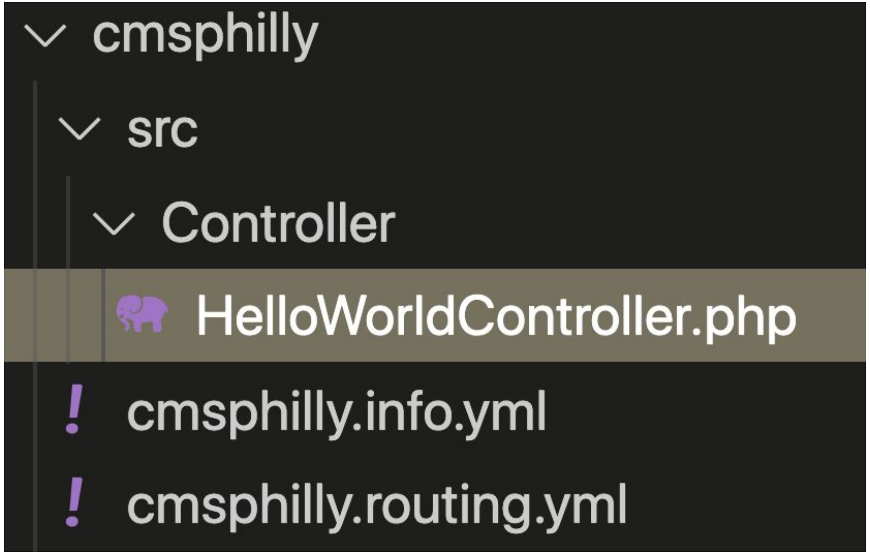
cmsphilly/src/Plugin.php



```
/**
 * @inheritdoc
 */
public function init()
{
    parent::init();
    self::$plugin = $this;

    // Register routes.
    Event::on(
        UrlManager::class,
        UrlManager::EVENT_REGISTER_CP_URL_RULES,
        function (RegisterUrlRulesEvent $event) {
            $event->rules[] = [
                'hello' => 'cmsphilly/hello-world/say-hello'
            ];
        }
    );

}
```

Controllers | Drupal

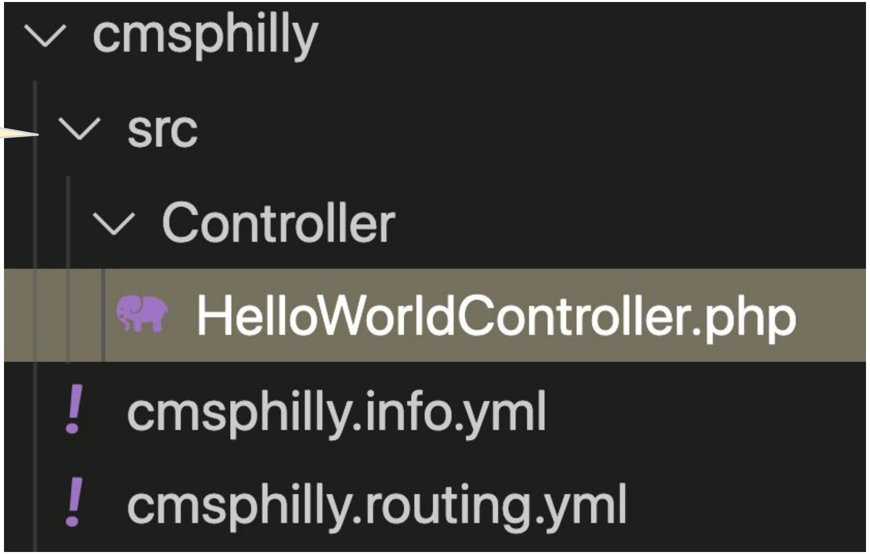
Where to put the Controller file:



```
cmsphilly
  src
    Controller
      HelloWorldController.php
  cmsphilly.info.yml
  cmsphilly.routing.yml
```

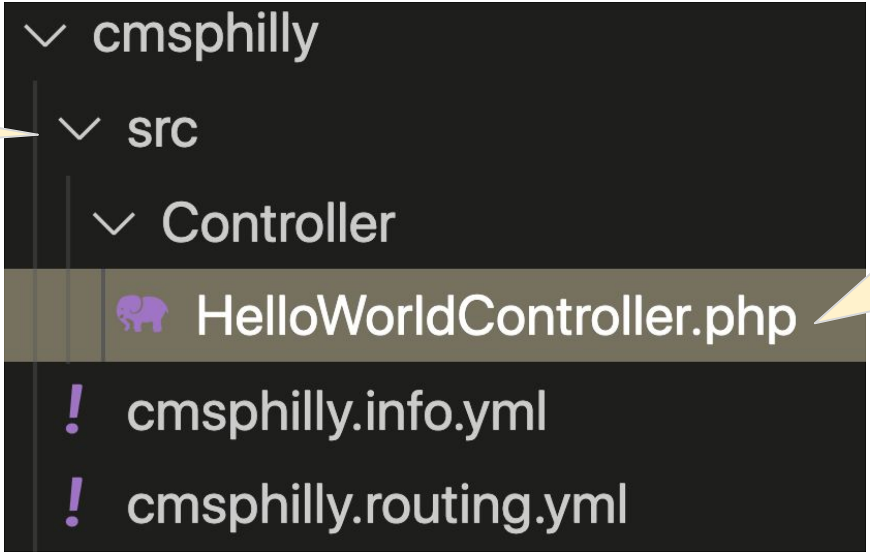
Where to put the Controller file:

Only autoloaded PHP classes are in the "src" directory



```
cmsphilly
  src
    Controller
      HelloWorldController.php
  cmsphilly.info.yml
  cmsphilly.routing.yml
```

Where to put the Controller file:

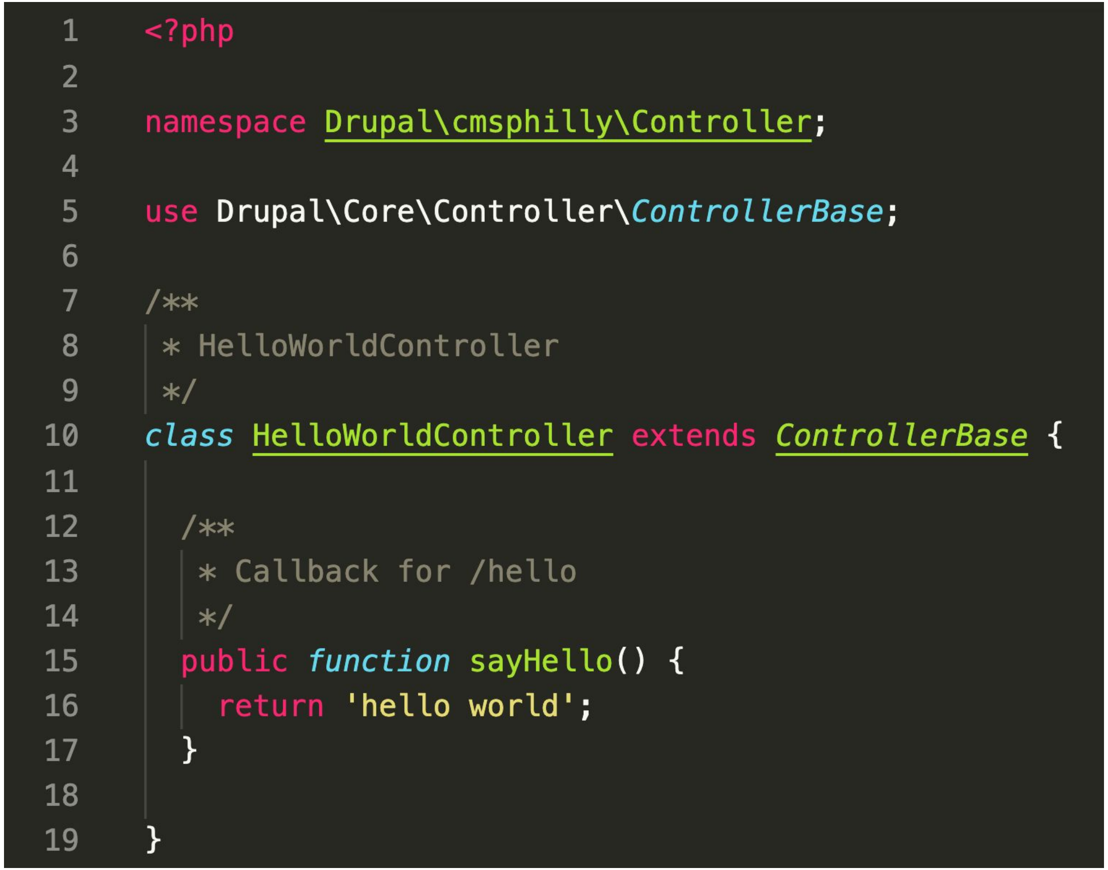


```
cmsphilly
  src
    Controller
      HelloWorldController.php
  cmsphilly.info.yml
  cmsphilly.routing.yml
```

Only autoloaded PHP classes are in the "src" directory

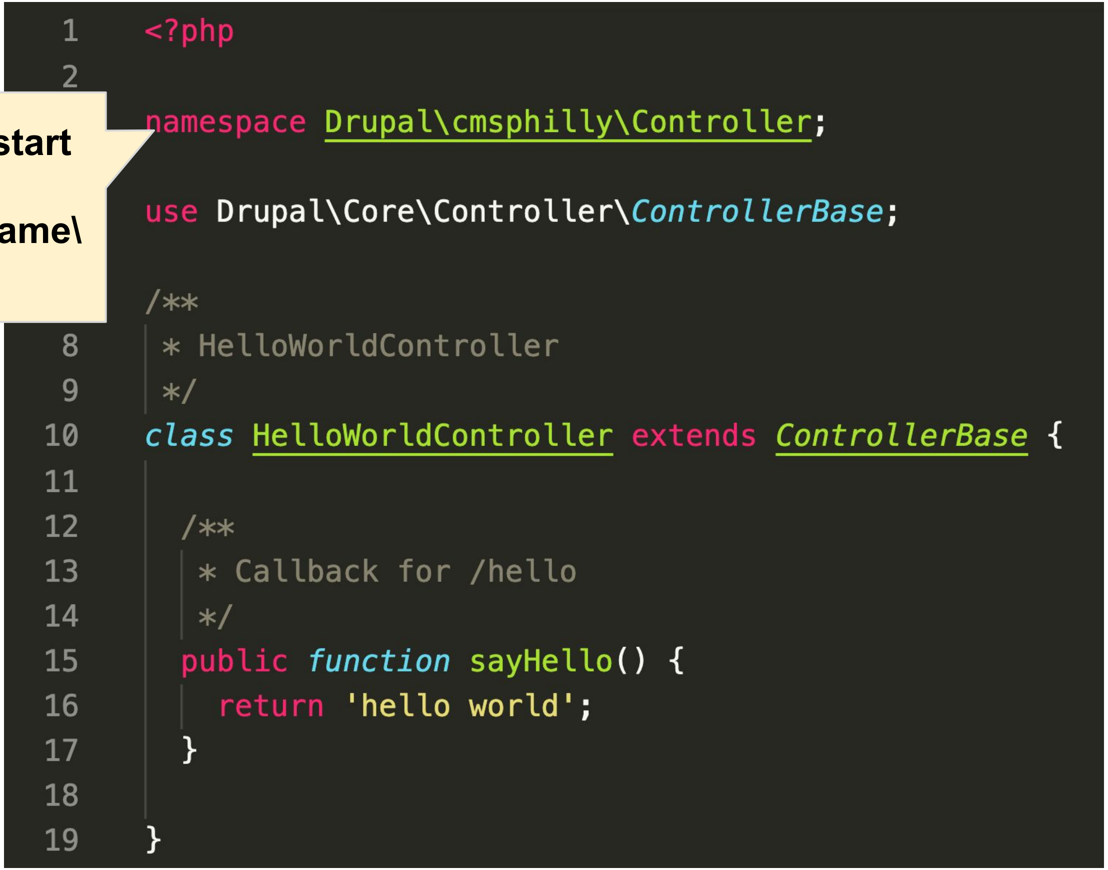
You can name your controller anything! I just appended "Controller" to the name out of habit

How to create a route with a Controller



```php
<?php

namespace Drupal\cmsphilly\Controller;

use Drupal\Core\Controller\ControllerBase;

/**
 * HelloWorldController
 */
class HelloWorldController extends ControllerBase {

  /**
   * Callback for /hello
   */
  public function sayHello() {
    return 'hello world';
  }

}
```

How to create a route with a Controller

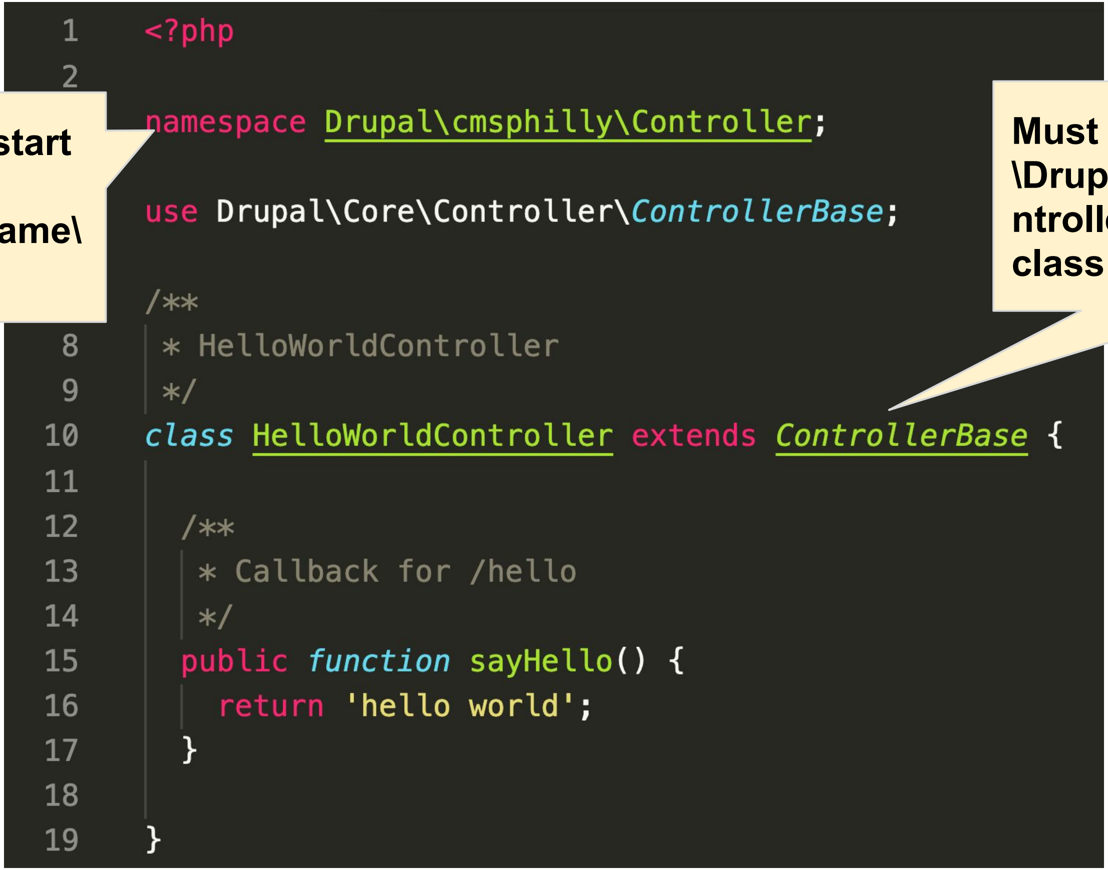


```php
<?php

namespace Drupal\cmsphilly\Controller;

use Drupal\Core\Controller\ControllerBase;

/**
 * HelloWorldController
 */
class HelloWorldController extends ControllerBase {

  /**
   * Callback for /hello
   */
  public function sayHello() {
    return 'hello world';
  }

}
```

Namespace must start with “Drupal\module_name\YadaYada”

How to create a route with a Controller

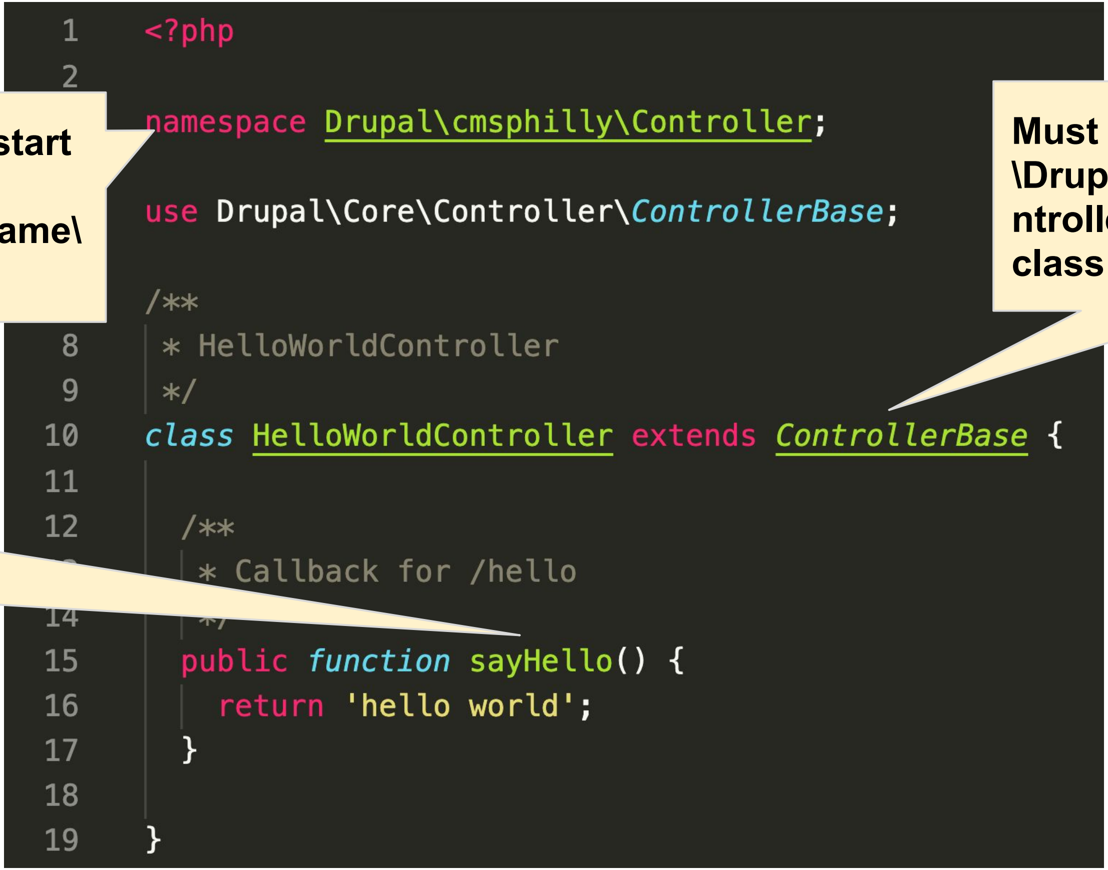
Namespace must start with "Drupal\module_name\YadaYada"



```php
<?php

namespace Drupal\cmsphilly\Controller;

use Drupal\Core\Controller\ControllerBase;

/**
 * HelloWorldController
 */
class HelloWorldController extends ControllerBase {

  /**
   * Callback for /hello
   */
  public function sayHello() {
    return 'hello world';
  }

}
```

Must extend the \Drupal\Core\Controller\ControllerBase (or child) class

How to create a route with a Controller



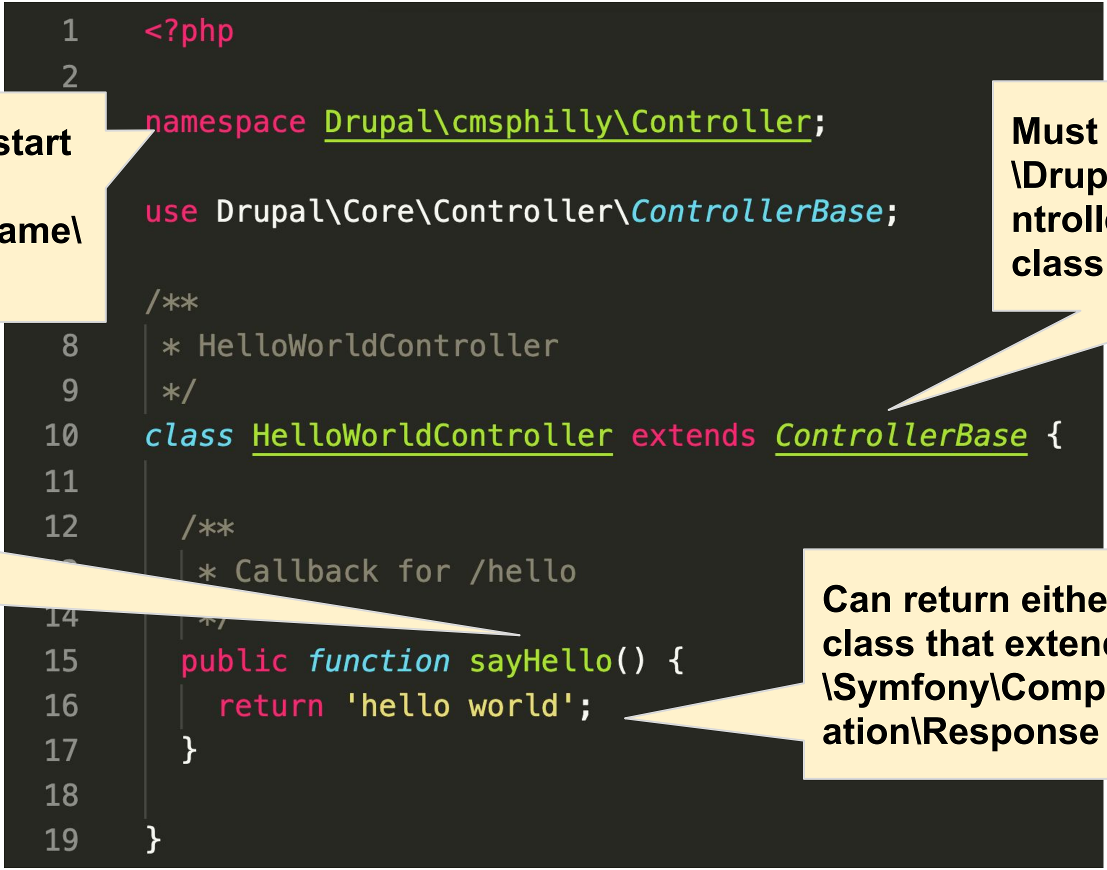
```php
<?php

namespace Drupal\cmsphilly\Controller;

use Drupal\Core\Controller\ControllerBase;

/**
 * HelloWorldController
 */
class HelloWorldController extends ControllerBase {

  /**
   * Callback for /hello
   */
  public function sayHello() {
    return 'hello world';
  }

}
```

Namespace must start with “Drupal\module_name\YadaYada”

Must extend the \Drupal\Core\Controller\ControllerBase (or child) class

Callback function can be named anything

How to create a route with a Controller



```php
<?php

namespace Drupal\cmsphilly\Controller;

use Drupal\Core\Controller\ControllerBase;

/**
 * HelloWorldController
 */
class HelloWorldController extends ControllerBase {

  /**
   * Callback for /hello
   */
  public function sayHello() {
    return 'hello world';
  }

}
```

Namespace must start with “Drupal\module_name\YadaYada”

Must extend the \Drupal\Core\Controller\ControllerBase (or child) class

Callback function can be named anything

Can return either plain text or any class that extends \Symfony\Component\HttpFoundation\Response (e.g. Json)

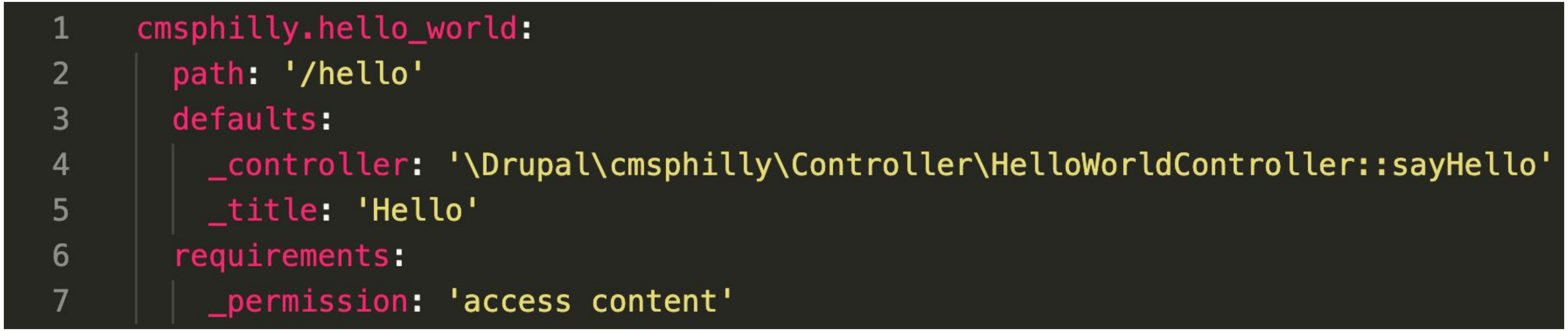
Connecting a route to your controller

Now, tell Drupal about your controller!

Connecting a route to your controller

Now, tell Drupal about your controller!

`modules/custom/cmsphilly/cmsphilly.routing.yml`



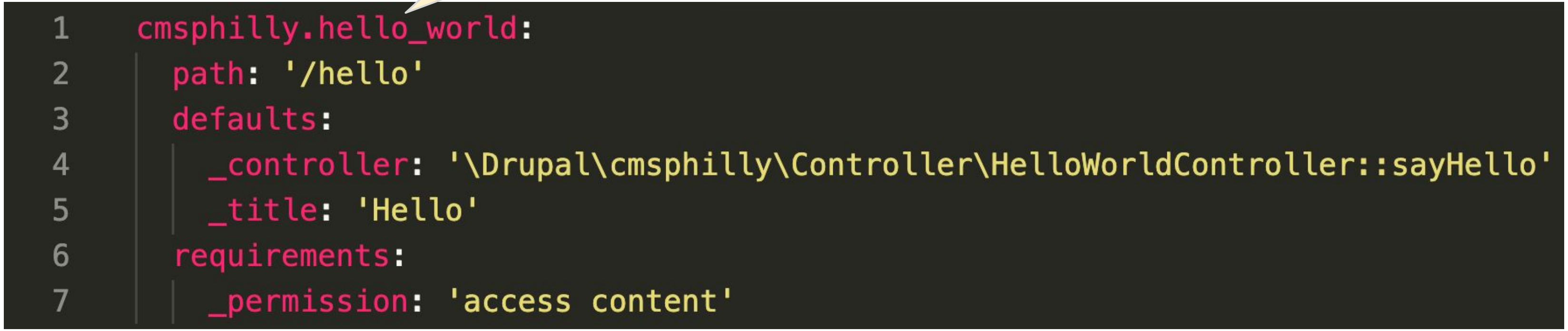
```yaml
cmsphilly.hello_world:
  path: '/hello'
  defaults:
    _controller: '\Drupal\cmsphilly\Controller\HelloWorldController::sayHello'
    _title: 'Hello'
  requirements:
    _permission: 'access content'
```

Connecting a route to your controller

Now, tell Drupal about your controller!

Symfony route name

modules/custom/cmsphilly/cmsphilly.routing.yml



```yaml
cmsphilly.hello_world:
  path: '/hello'
  defaults:
    _controller: '\Drupal\cmsphilly\Controller\HelloWorldController::sayHello'
    _title: 'Hello'
  requirements:
    _permission: 'access content'
```

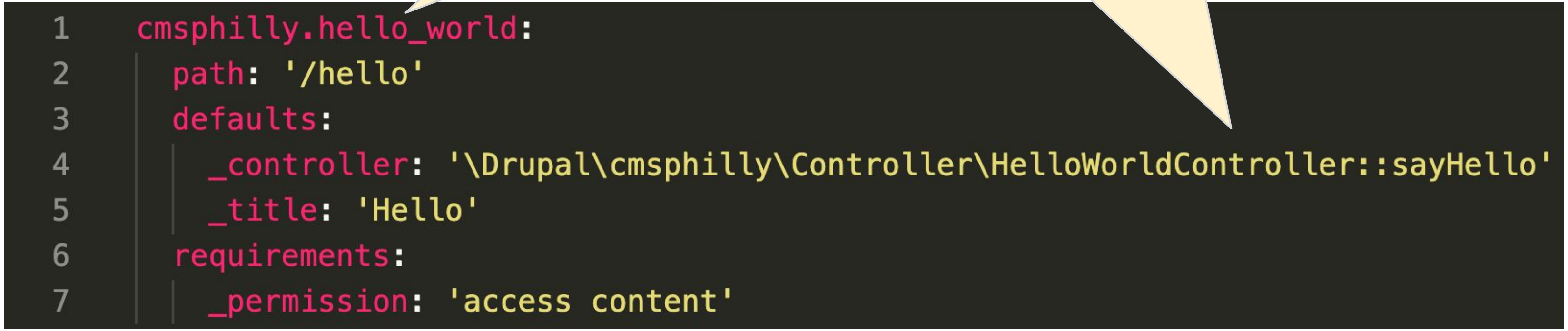
Connecting a route to your controller

Now, tell Drupal about your controller!

Symfony route name

Full namespaced class name, with method separated by “::” (scope resolution operator)

modules/custom/cmsphilly/cmsphilly.routing.yml



```yaml
cmsphilly.hello_world:
  path: '/hello'
  defaults:
    _controller: '\Drupal\cmsphilly\Controller\HelloWorldController::sayHello'
    _title: 'Hello'
  requirements:
    _permission: 'access content'
```

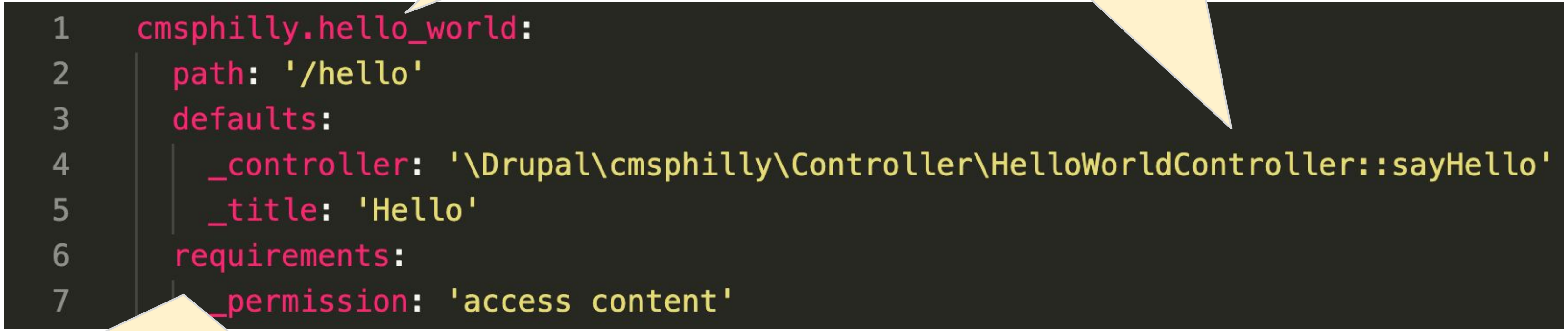
Connecting a route to your controller

Now, tell Drupal about your controller!

Symfony route name

Full namespaced class name, with method separated by “::” (scope resolution operator)

`modules/custom/cmsphilly/cmsphilly.routing.yml`



```
cmsphilly.hello_world:
  path: '/hello'
  defaults:
    _controller: '\Drupal\cmsphilly\Controller\HelloWorldController::sayHello'
    _title: 'Hello'
  requirements:
    _permission: 'access content'
```

Can be either Drupal permissions, role, custom callback

Controllers | WordPress

How to create a route with a Controller

Where to put the Controller file:

How to create a route with a Controller

Where to put the Controller file:

TRICK QUESTION!

There are no controllers in WordPress.

How to create a route with a Controller

Where to put the Controller file:

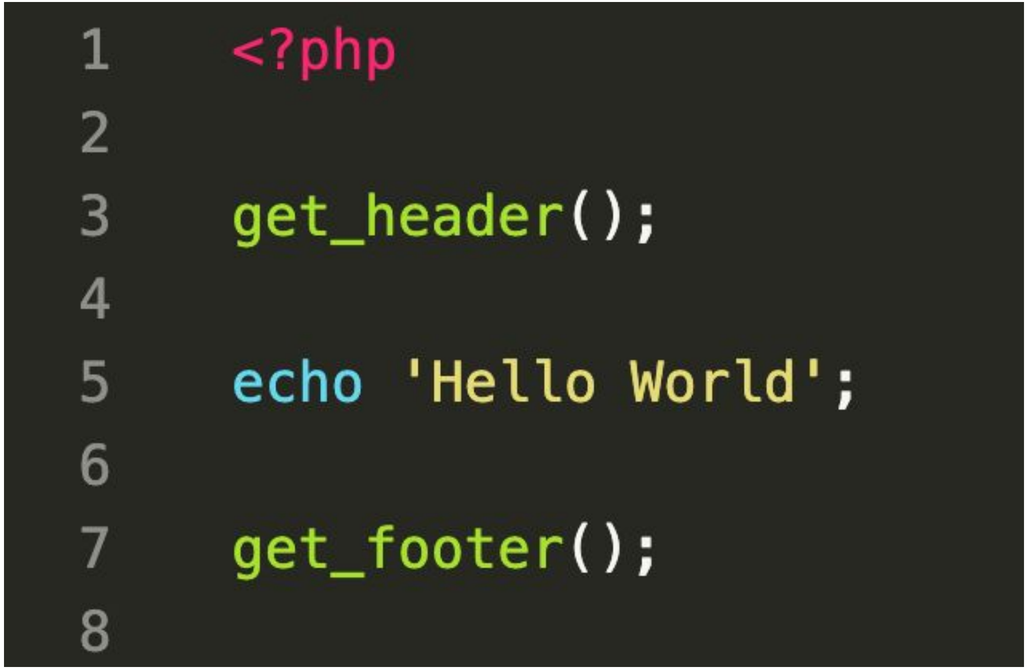
TRICK QUESTION!

There are no controllers in WordPress.

*(Unless you build your own MVC framework **within** your plugin…)*

How to create a “controller”
(really a plugin template)

`wp-content/plugins/cmsphilly/hello-template.php`



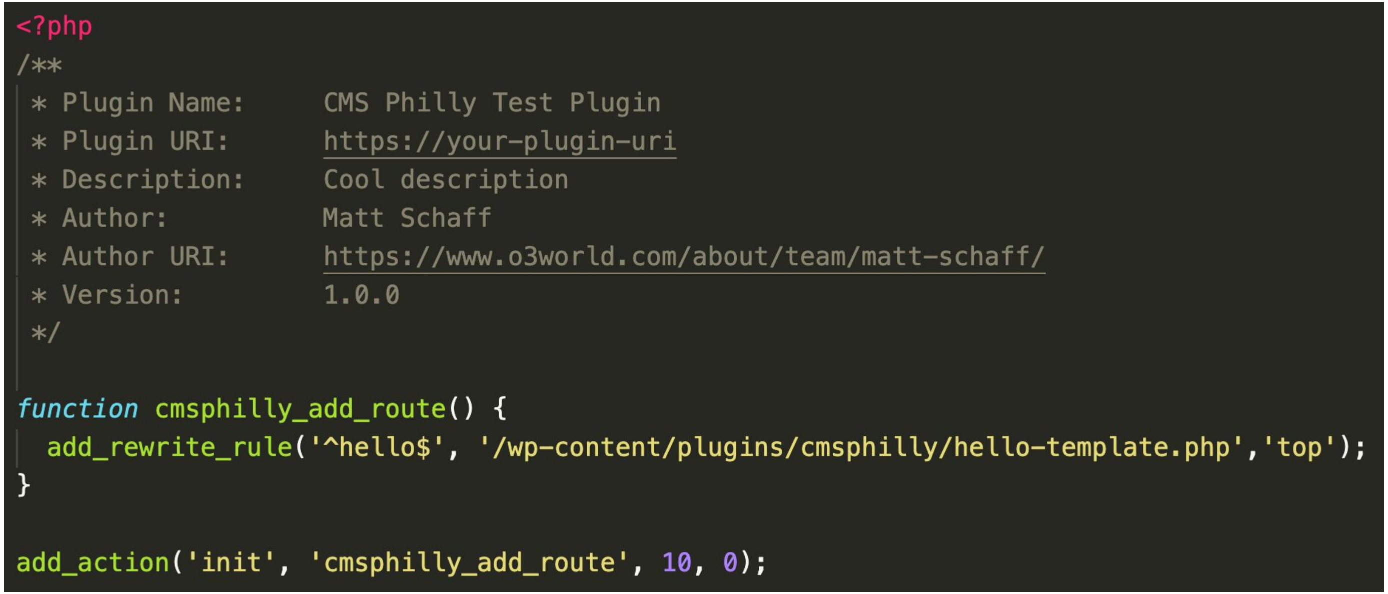
```php
<?php

get_header();

echo 'Hello World';

get_footer();

```

How to connect a route with a template

Tell WordPress about your template!

How to connect a route with a template

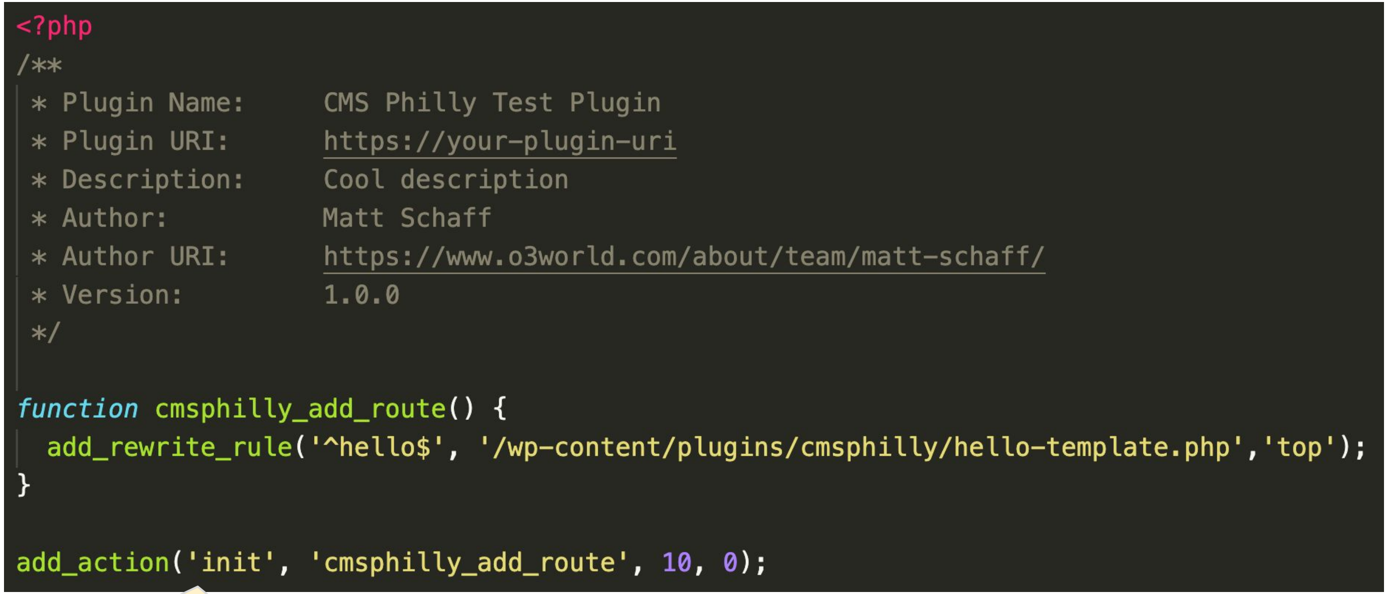
wp-content/plugins/cmsphilly/cmsphilly.php



```php
<?php
/**
 * Plugin Name:     CMS Philly Test Plugin
 * Plugin URI:      https://your-plugin-uri
 * Description:     Cool description
 * Author:          Matt Schaff
 * Author URI:      https://www.o3world.com/about/team/matt-schaff/
 * Version:         1.0.0
 */

function cmsphilly_add_route() {
  add_rewrite_rule('^hello$', '/wp-content/plugins/cmsphilly/hello-template.php','top');
}

add_action('init', 'cmsphilly_add_route', 10, 0);
```

How to connect a route with a template

wp-content/plugins/cmsphilly/cmsphilly.php

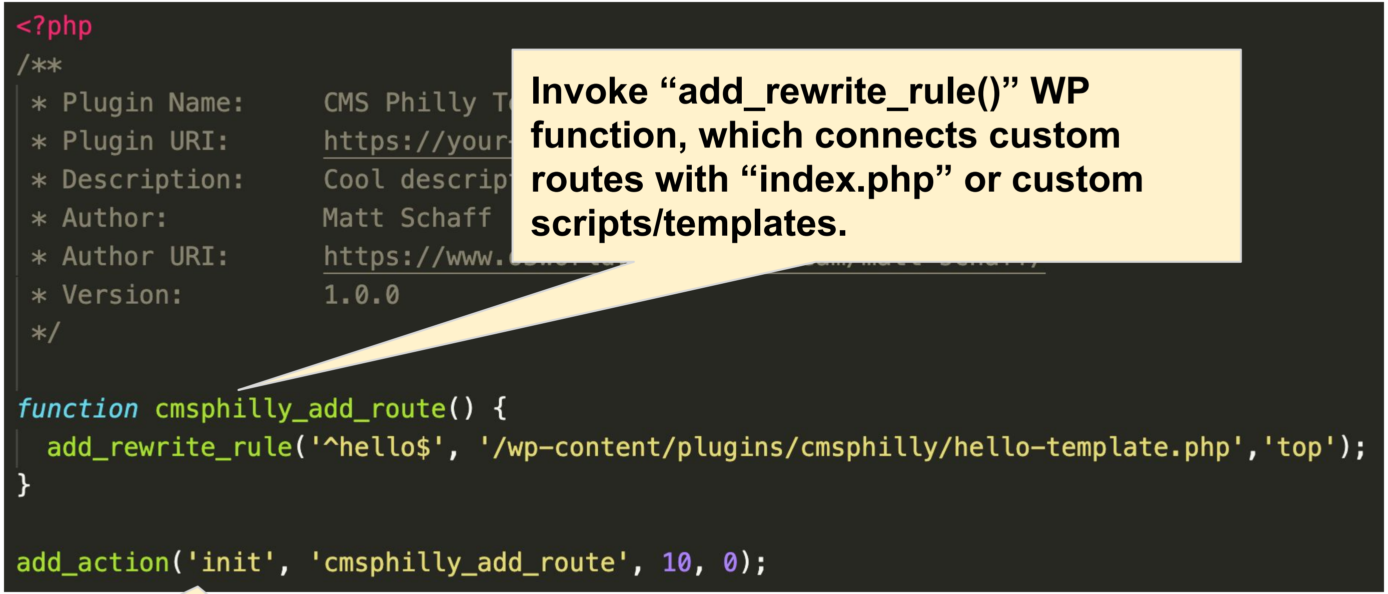


```php
<?php
/**
 * Plugin Name:     CMS Philly Test Plugin
 * Plugin URI:      https://your-plugin-uri
 * Description:     Cool description
 * Author:          Matt Schaff
 * Author URI:      https://www.o3world.com/about/team/matt-schaff/
 * Version:         1.0.0
 */

function cmsphilly_add_route() {
  add_rewrite_rule('^hello$', '/wp-content/plugins/cmsphilly/hello-template.php','top');
}

add_action('init', 'cmsphilly_add_route', 10, 0);
```

Action hook: Run a function when WordPress “initializes”

How to connect a route with a template

wp-content/plugins/cmsphilly/cmsphilly.php



```
<?php
/**
 * Plugin Name:     CMS Philly T
 * Plugin URI:      https://your-
 * Description:     Cool descrip
 * Author:          Matt Schaff
 * Author URI:      https://www.
 * Version:         1.0.0
 */

function cmsphilly_add_route() {
  add_rewrite_rule('^hello$', '/wp-content/plugins/cmsphilly/hello-template.php','top');
}

add_action('init', 'cmsphilly_add_route', 10, 0);
```

Invoke “add_rewrite_rule()” WP function, which connects custom routes with “index.php” or custom scripts/templates.

Action hook: Run a function when WordPress “initializes”

Contribute your plugin

craft cms https://docs.craftcms.com/v3/extend/plugin-store.html#registering-your-plugin

Drupal https://www.drupal.org/contribute-projects

https://wordpress.org/plugins/developers/add/

Matt Schaff
mattschaff@o3world.com

Invisible Hands

HELP YOUR NEIGHBORS MAKE A DIFFERENCE

About Us

We're Invisible Hands, a group of young volunteers offering free delivery of groceries, supplies, or whatever else you may need.

Know Drupal?
Wanna help?
@ me

Matt Schaff
engineering@invisiblehandsdeliver.org
https://invisiblehandsdeliver.org