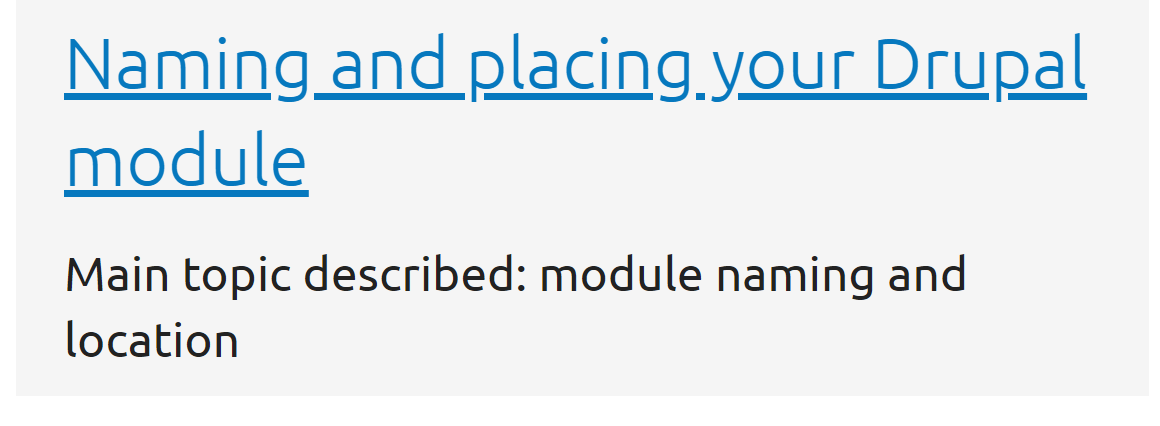
<https://www.drupal.org/docs/creating-custom-modules>

当前学习：



# 起名

不是乱来的，小写，

aa\_bb\_cc

drupal自动识别为：Aa Bb Cc

而且一旦定义好了，跟它相关的方法命名也就定下来了。

# 创建目录

/modules/custom/hello\_world

或者/sites/all/modules/custom/hello\_world

目录名与机器名可以不同，**但是为了保证代码的可读性，我们规定模块目录名必须和模块机器名一样。**

# 文件

## hello\_world.info.yml

* dependencies: 列出你的模块依赖的模块。其格式为{project}:{module}，{project}是出现在Drupal.org项目url中的项目名称，例如drupal.org/project/views，{module}是模块的机器名。
* test\_dependencies: 测试的依赖模块。testbot会对整个模块的依赖关系进行测试。如果通过测试依赖关系是正确的、完整的，应该将其移动dependencies中。在上面的例子中，在测试期间testbot会包含image模块。但站点安装”Hello World Module”模块时不会要求安装image模块。测试依赖的命名规则与依赖的命名规则相同。 testbot测试见 [https://www.drupal.org/docs/8/phpunit/running-phpunit-tests。](https://www.drupal.org/docs/8/phpunit/running-phpunit-tests%E3%80%82)
* configure: 如果你的模块提供了一个配置表单，在这里可以指定配置的路由。当用户展开模块细节时，它将会在Extend页(/admin/modules)显示一个链接。
* hidden:true 在站点Extend页面(/admin/modules)的模块列表中隐藏你的模块。如果模块仅仅包含一个测试或作为一个开发例子，你发现使用这个键是非常有用的。

## hello\_world.routing.yml

包含URL路径和回调函数的映射关系

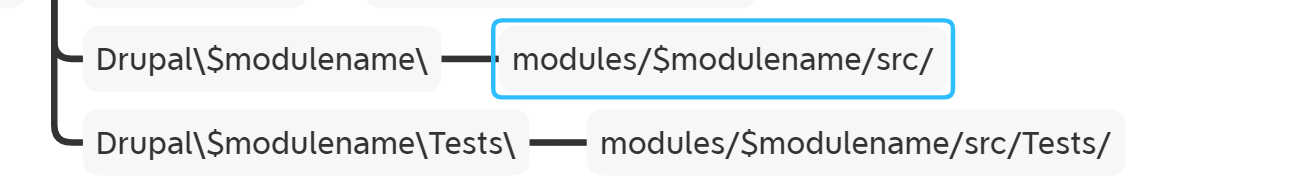
### path: 路由注册的路径，需要以斜线开头

### \_controller: 定义路由的路径HelloController的content方法

如：\_controller: '\Drupal\hello\_world\Controller\HelloController::content'

这个控制器的写在目录：/src/Controller/HelloController.php

遵循PSR-4标准



HelloController 的写法

**<?php**

/\*\*

\* @file

\* Contains \Drupal\hello\_world\Controller\HelloController.

\*/

namespace Drupal\hello\_world\Controller;

use Drupal\Core\Controller\ControllerBase;

class HelloController extends ControllerBase {

public function content() {

return array(

'#type' => 'markup',

'#markup' => $this->t('Hello, World!'),

);

}

}

### 测试路由

在url地址栏输入/hello，你将会看到”Hello,World!”这样的信息。

### requirements: 用户能够访问这个页面所具有的权限

## hello\_world.links.menu.yml

包含菜单项的结构

* route\_name: 在routing.yml中定义的路由名称

parent: 描述的是菜单的父链接菜单。

parent: system.admin\_config\_development

这个菜单链接将会在admin>config>development下创建。

## hello\_world.links.action.yml

等效Drual7的常量MENU\_LOCAL\_ACTION

## hello\_world.links.task.yml

等效Drupal7的常量MENU\_DEFAULT\_LOCAL\_TASK

# 区块开发

ExampleBlock.php>>build()

id: 区块ID

admin\_label: 经过翻译的管理标签

# 表单开发

## 表单元素

* type: 表单类型，如textfield, textarea, html5又包含tel,email,number,date,url,search,range等
* title: 表单标题
* value: 表单值
* default\_value: 表单默认值
* weight: 权重，可控制表单元素的位置
* placeholder: 占位符，如输入密码的文本框中显示"请输入密码"

## 创建表单

### getFormId:

* 必须定义的方法，返回表单机器名。

### buildForm:

* 必须定义的，它返回一个渲染数组，类似 hook\_form。

支持的表单：

<http://nowicode.com/bookpage/212>

### 1、电话号码(tel)

$form['phone'] = array(

'#type' => 'tel',

'#title' => t('Phone'),

);

copy

### 2、邮箱(email)

$form['email'] = array(

'#type' => 'email',

'#title' => t('Email'),

);

copy

### 3、数字(integer)

$form['integer'] = array(

'#type' => 'number',

'#title' => t('Some integer'),

// The increment or decrement amount

'#step' => 1,

// Miminum allowed value

'#min' => 0,

// Maxmimum allowed value

'#max' => 100,

);

copy

### 4、日期(date)

$form['date'] = array(

'#type' => 'date',

'#title' => t('Date'),

'#date\_date\_format' => 'Y-m-d',

);

copy

### 5、输入URL(website)

$form['website'] = array(

'#type' => 'url',

'#title' => t('Website'),

);

copy

### 6、搜索关键字(search)

$form['search'] = array(

'#type' => 'search',

'#title' => t('Search'),

'#autocomplete\_route\_name' => FALSE,

);

copy

注意：填入正确的route\_name，会启用自动完成搜索。

### 7、范围(range)

$form['range'] = array(

'#type' => 'range',

'#title' => t('Range'),

'#min' => 0,

'#max' => 100,

'#step' => 1,

);

### validateForm:

* 是可选的，进行校验，类似 hook\_form\_validate。

#### $form\_state

* isValueEmpty: 用来验证值是否为空
* getValue: 获取元素的值
* setErrorByName: 抛出异常错误，第一个参数是元素名称，第二个参数是错误信息。
* setValidateHandlers 一个表单可以支持多个验证处理器

$form\_state->setErrorByName('email'**,** $this->t('This is not a .com email address.'))**;**



### submitForm:

* 执行提交处理，类似 hook\_form\_submit。

一个表单可以有多个提交处理器

如一个类提供了method1和method2，要想执行它们。可以在buildForm方法中添加下面的代码:

$form\_state->setSubmitHandlers([

['::submitForm'],

['::method1'],

[$this, 'method2']

]);

执行顺序为从上自下。

## 在controller中调用form

$form = \Drupal::formBuilder()->getForm('Drupal\hello\_world\Form\HelloWorldForm');

## 修改已存在的表单

# 字段开发

## 注解

* id: 字段类型的机器名，唯一ID。
* label: 用于用户界面的标签。
* description: 字段描述

class EntityUserAccessField extends FieldItemBase {

modules/custom/hello\_world/src/Plugin/Field/FieldType/EntityUserAccessField.php

## propertyDefinitions:

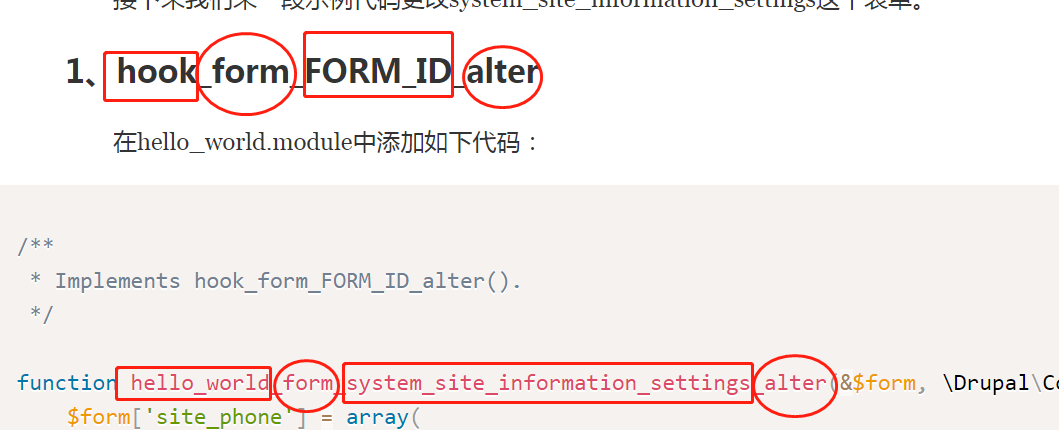
和baseFieldDefinition类似，定义字段的属性。

## schema:

告诉Drupal如何创建字段。

# 模块HOOK调用

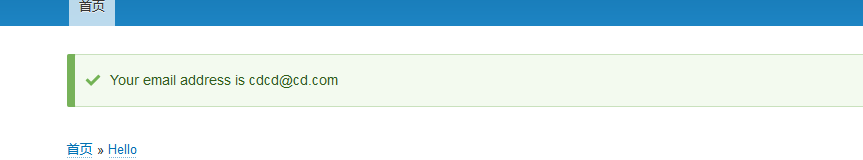
## HOOK对应法则



# API收集

## drupal\_set\_message

drupal\_set\_message($this->t('Your email address is @email'**,** ['@email' => $form\_state->getValue('email')]))**;**



## $form\_state

$form\_state->setErrorByName('email'**,** $this->t('This is not a .com email address.'))**;**

