# 安装yii

**通过 Composer 安装：**

1.在 Linux 和 Mac OS X 中可以运行如下命令：

curl -sS https://getcomposer.org/installer | php

mv composer.phar /usr/local/bin/composer

2.在 Windows 中，你需要下载并运行Composer-Setup.exe

<https://getcomposer.org/Composer-Setup.exe>

composer config -g github-oauth.github.com 4698a93184b0e814f31fa0efa6582f5dc4fb6ec8

composer global require "fxp/composer-asset-plugin:^1.3.1"

composer create-project --prefer-dist yiisoft/yii2-app-basic basic

# 控制器

yii的控制器是继承yii\web\Controller类的对象，负责处理请求和生成响应

## 控制器

### 2.1.2创建控制器

yii的控制器应继承yii\web\Controller ，eg：

namespace app\controllers;

use yii\web\Controller;

class SiteController extends Controller{

}

### 2.1.2控制器命名

通常情况下，控制器用来处理请求有关的资源类型， 因此控制器ID通常为和资源有关的名词。

控制器ID遵循以下规则衍生控制器类名：

**1.将用正斜杠区分的每个单词第一个字母转为大写。注意如果控制器ID包含正斜杠， 只将最后的正斜杠后的部分第一个字母转为大写；**

**2.去掉中横杠，将正斜杠替换为反斜杠;**

**3.增加Controller后缀;**

**4.在前面增加controller namespace控制器命名空间.**

例如使用article作为处理文章的控制器ID：

article 对应 app\controllers\ArticleController;

post-comment 对应 app\controllers\PostCommentController;

controller类必须能被自动加载，因此控制器article 类应在 别名 为app/controllers/ArticleController.php的文件中定义，控制器admin/post-comment应在app/controllers/admin/PostCommentController.php文件中

有时候，我们会遇到使用第三方开发的不可掌控的控制器名，我们可通过配置 yii\base\Application::controllerMap 来强制上述的控制器ID和类名对应

## 2.2操作

### 2.2.1创建操作方法

控制器由操作方法组成，它是执行终端用户请求的最基础的单元，一个控制器可有一个或多个操作。

操作方法必须是以action开头的公有方法，如create操作，就是public function actionCreate(){}，Yii 使用 action 前缀区分普通方法和操作。action 前缀后面的名称被映射为操作的 ID。操作通常是用来执行资源的特定操作，因此，操作ID通常为动词，如view, update等。

**namespace** app\controllers;  
**use** yii\web\Controller;  
**class** SiteController **extends** Controller  
{

**//**定义index操作返回一个index.php的视图  
 **public function** actionIndex()  
 {  
 **return** $this->render(**'index'**);  
 }

**//**定义hello-world操作返回一个HelloWorld字符串  
 **public function** actionHelloWorld()  
 {  
 **return 'Hello World'**;  
 }  
}

### 2.2.2操作ID

操作ID应仅包含英文小写字母、数字、下划线和中横杠，操作ID中的中横杠用来分隔单词，可通过两种方式创建操作ID：内联操作和独立操作.

内联操作是在控制器类中定义以action最为前缀的方法；独立操作是继承yii\base\Action或它的子类的类。 内联操作容易创建，在无需重用的情况下优先使用； 独立操作相反，主要用于多个控制器重用，或重构为扩展

**1.内联操作**

动作方法的名字是根据操作ID遵循如下规则衍生：

**1.将每个单词的第一个字母转为大写;**

**2.去掉中横杠;**

**3.增加action前缀.**

例如index 转成 actionIndex, hello-world 转成 actionHelloWorld。ActionIndex则不会被认为是操作方法，所以请求index操作会返回一个异常

**2.独立操作**

### 2.2.3动作参数

内联动作的操作方法和独立动作的 run() 方法可以带参数，称为**动作参数**，对于Web applications网页应用， 每个动作参数的值从$\_GET中获得，参数名作为键；对于console applications控制台应用, 动作参数对应命令行参数。下例中使用了id和version参数，eg：

**namespace** app\controllers;  
**use** yii\web\Controller;  
**class** PostController **extends** Controller  
{  
 **public function** actionView($id, $version = **null**)  
 {  
 *// ...* }  
}

<http://hostname/index.php?r=post/view&id=123> $id会填入'123'

<http://hostname/index.php?r=post/view&id=123&version=2> $id 和 $version 分别填入 '123' 和 '2'`；

[http://hostname/index.php?r=post/view&id[]=123](http://hostname/index.php?r=post/view&id%5b%5d=123) 会抛出异常 因为$id 参数收到数字值 ['123']而不是字符串.

**如果想让动作参数接收数组值**，需要指定$id为array，如下所示：

public function actionView(array $id, $version = null)

{

// ...

}

现在如果请求为 http://hostname/index.php?r=post/view&id[]=123, 参数 $id 会使用数组值['123']， 如果请求为 http://hostname/index.php?r=post/view&id=123， 参数 $id 会获取相同数组值，因为无类型的'123'会自动转成数组

经过美化的url使用参数一般为：

<http://127.0.0.1/modulesName/controllerName/actionName?id=123&vesion=2>

### 2.2.3默认动作

每一个控制器都有一个由**yii\base\Controller::$defaultAction属性**指定的默认操作，当路由只包含控制器ID(没有指定动作)，会使用所请求的控制器的默认操作。

**namespace** app\controllers;  
**use** yii\web\Controller;  
**class** SiteController **extends** Controller  
{  
 **public $defaultAction** = **'home'**;  
 **public function** actionHome()  
 {  
 **return** $this->render(**'home'**);  
 }  
}

# Twig模板

模板是一种文本文件，它可以生成任何基于文本格式的文件(如HTML,XML等等)，一个模板通常包括变量和表达式，当模板被使用时(转化为目标文本格式如HTML,XML)，变量和表达式会被值所取代

默认情况下，Yii 使用 PHP 作为其默认的模板引擎语言，但是，你可以配置 Yii 以扩展的方式支持其他的渲染引擎，比如 Twig 或 Smarty等。

## 3.1Twig安装配置

Twig需要至少PHP7.0以上才能运行

推荐使用Composer安装，避免处理过多的包依赖

**composer require "twig/twig:^2.0"**

使用Twig的第一步是注册它的autoloader