在应用中创建一个新的 “Hello” 页面：

(主要涉及控制器)

 这将会创建一个[操作](http://www.yiichina.com/doc/guide/2.0/structure-controllers#creating-actions) 和一个[视图](http://www.yiichina.com/doc/guide/2.0/structure-views)：

应用将会分派页面请求给 动作

动作将会依次渲染 视图 呈现 “Hello” 给最终用户

贯穿整个章节，要掌握三件事：

1. 如何创建一个[动作](http://www.yiichina.com/doc/guide/2.0/structure-controllers)去响应请求，
2. 如何创建一个[视图](http://www.yiichina.com/doc/guide/2.0/structure-views)去构造响应内容，
3. 以及一个应用如何分派请求给[动作](http://www.yiichina.com/doc/guide/2.0/structure-controllers" \l "creating-actions)。

一、创建动作（在控制器中） ：

为了 “Hello”，需要创建一个 say [操作](http://www.yiichina.com/doc/guide/2.0/structure-controllers#creating-actions)， 从请求中接收 message 参数并显示给最终用户。 如果请求没有提供 message 参数，操作将显示默认参数 “Hello”。

[操作](http://www.yiichina.com/doc/guide/2.0/structure-controllers#creating-actions)是最终用户可以直接访问并执行的对象。 操作被组织在[控制器](http://www.yiichina.com/doc/guide/2.0/structure-controllers)中。 一个操作的执行结果就是最终用户收到的响应内容。

操作必须声明在[控制器](http://www.yiichina.com/doc/guide/2.0/structure-controllers)中。

 比如这个控制器由文件 controllers/TestController.php 定义。以下是一个操作的声明：

<?php

**namespace** **app**\**controllers**;

**use** **yii**\**web**\**Controller**;

**class** **TestController** **extends** **Controller**

{

// ...现存的代码...

**public** **function** **actionSay**($message = 'Hello')

{

**return** $this->render('say', ['message' => $message]);

}

}

上述 TestController 代码中，say 操作被定义为 actionSay 方法。

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 ID 总是被以小写处理，如果一个操作 ID 由多个单词组成， 单词之间将由连字符连接（如 create-comment）。 操作 ID 映射为方法名时移除了连字符，将每个单词首字母大写，并加上 action 前缀。 例子：操作 ID create-comment 相当于方法名 actionCreateComment。

上述的操作方法接受一个参数 $message， 它的默认值是 “Hello”.

在操作方法中，yii\web\Controller::render() 被用来渲染一个 名为 say 的[视图](http://www.yiichina.com/doc/guide/2.0/structure-views)文件。 message 参数也被传入视图，这样就可以在里面使用。操作方法会返回渲染结果。 结果会被应用接收并显示给最终用户的浏览器（作为整页 HTML 的一部分）。

二、创建视图：

[视图](http://www.yiichina.com/doc/guide/2.0/structure-views)是你用来生成响应内容的脚本。为了说 “Hello”， 你需要创建一个 say 视图，以便显示从操作方法中传来的 message 参数。

say 视图应该存为 views/test/say.php 文件:

<?php **use** **yii**\**helpers**\**Html**; ?> <?= Html::encode($message) ?>

当一个操作中调用了 yii\web\Controller::render() 方法时， 它将会按 views/控制器ID/视图名.php 路径加载 PHP 文件。

以上代码中，message 参数在输出之前被 yii\helpers\Html::encode() 方法处理过。 这很有必要，当参数来自于最终用户时，参数中可能隐含的恶意 JavaScript 代码会导致 [跨站脚本（XSS）攻击](http://en.wikipedia.org/wiki/Cross-site_scripting)。

在 say 视图里可以放入更多内容。内容可以由 HTML 标签，纯文本， 甚至 PHP 语句组成。实际上 say 视图就是一个由 yii\web\Controller::render() 执行的 PHP 脚本。

视图脚本输出的内容将会作为响应结果返回给应用。

应用将依次输出结果给最终用户。

## 试运行

创建完动作和视图后，你就可以通过下面的 URL 访问新页面了：

**http**://hostname/index.php?r=test/say&message=Hello+World