# 控制器

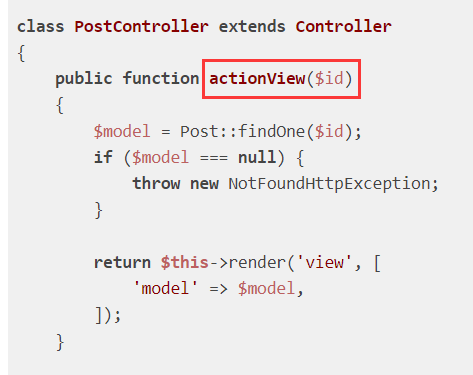
控制器是 [MVC](http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) 模式中的一部分， 是继承[yii\base\Controller](https://www.yiichina.com/doc/api/2.0/yii-base-controller)类的对象，负责处理请求和生成响应。

具体来说，控制器从[应用主体](https://www.yiichina.com/doc/guide/2.0/structure-applications) 接管控制后会分析请求数据并传送到[模型](https://www.yiichina.com/doc/guide/2.0/structure-models)， 传送模型结果到[视图](https://www.yiichina.com/doc/guide/2.0/structure-views)，最后生成输出响应信息。

## 动作

控制器由 动作 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动作。

如下，前缀为action的方法就是动作，一个动作对应一个请求接口。



### 动作结果

操作方法的返回值非常重要， 它表示对应操作结果。

返回值可为 [响应](https://www.yiichina.com/doc/guide/2.0/runtime-responses) 对象，作为响应发送给终端用户。

* 对于[Web applications](https://www.yiichina.com/doc/api/2.0/yii-web-application)网页应用，返回值可为任意数据, 它赋值给[yii\web\Response::$data](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "$data-detail)， 最终转换为字符串来展示响应内容。
* 对于[console applications](https://www.yiichina.com/doc/api/2.0/yii-console-application)控制台应用，返回值可为整数， 表示命令行下执行的 [exit status](https://www.yiichina.com/doc/api/2.0/yii-base-response" \l "$exitStatus-detail) 退出状态。

## 路由

终端用户通过所谓的路由寻找到动作，路由是包含以下部分的字符串：

* 模块ID: 仅存在于控制器属于非应用的[模块](https://www.yiichina.com/doc/guide/2.0/structure-modules);
* 控制器ID: 同应用（或同模块如果为模块下的控制器） 下唯一标识控制器的字符串;
* 操作ID: 同控制器下唯一标识操作的字符串。

路由使用如下格式:

ControllerID/ActionID

如果属于模块下的控制器，使用如下格式：

ModuleID/ControllerID/ActionID

如果用户的请求地址为 http://hostname/index.php?r=site/index, 会执行site 控制器的index 动作。

## 控制器生命周期

处理一个请求时，[应用主体](https://www.yiichina.com/doc/guide/2.0/structure-applications) 会根据请求 [路由](https://www.yiichina.com/doc/guide/2.0/structure-controllers" \l "routes)创建一个控制器， 控制器经过以下生命周期来完成请求：

1. 在控制器创建和配置后，[yii\base\Controller::init()](https://www.yiichina.com/doc/api/2.0/yii-base-baseobject" \l "init()-detail) 方法会被调用。
2. 控制器根据请求操作ID创建一个操作对象:

* 如果操作ID没有指定，会使用[default action ID](https://www.yiichina.com/doc/api/2.0/yii-base-controller" \l "$defaultAction-detail)默认操作ID；
* 如果在[action map](https://www.yiichina.com/doc/api/2.0/yii-base-controller" \l "actions()-detail)找到操作ID， 会创建一个独立操作；
* 如果操作ID对应操作方法，会创建一个内联操作；
* 否则会抛出[yii\base\InvalidRouteException](https://www.yiichina.com/doc/api/2.0/yii-base-invalidrouteexception)异常。

1. 控制器按顺序调用应用主体、模块（如果控制器属于模块）、 控制器的 beforeAction() 方法；

* 如果任意一个调用返回false，后面未调用的beforeAction()会跳过并且操作执行会被取消； action execution will be cancelled.
* 默认情况下每个 beforeAction() 方法会触发一个 beforeAction 事件，在事件中你可以追加事件处理操作；

1. 控制器执行操作:

* 请求数据解析和填入到操作参数；

1. 控制器按顺序调用控制器、模块（如果控制器属于模块）、应用主体的 afterAction() 方法；

* 默认情况下每个 afterAction() 方法会触发一个 afterAction 事件， 在事件中你可以追加事件处理操作；

1. 应用主体获取操作结果并赋值给[响应](https://www.yiichina.com/doc/guide/2.0/runtime-responses).

## 最佳实践

在设计良好的应用中，控制器很精练，包含的操作代码简短； 如果你的控制器很复杂，通常意味着需要重构， 转移一些代码到其他类中。

归纳起来，控制器

* 可访问 [请求](https://www.yiichina.com/doc/guide/2.0/runtime-requests) 数据;
* 可根据请求数据调用 [模型](https://www.yiichina.com/doc/guide/2.0/structure-models) 的方法和其他服务组件;
* 可使用 [视图](https://www.yiichina.com/doc/guide/2.0/structure-views) 构造响应;
* 不应处理应被[模型](https://www.yiichina.com/doc/guide/2.0/structure-models)处理的请求数据;
* 应避免嵌入HTML或其他展示代码，这些代码最好在 [视图](https://www.yiichina.com/doc/guide/2.0/structure-views)中处理.