# 控制器层典型实现

## 控制器类

依据功能的相关性，将一系列相关的功能，使用一个控制器类来处理，而该控制器的每个方法，就对因某个功能。

注意：控制器是按照功能划分的。（而不是像模型一样，按表来划分）

比赛相关功能控制器类：



## 前端控制器（请求分发器，入口文件）

以上的listAction()操作应该如何被调用呢？

实例化，并掉用方法即可！

在哪里实例化或调用呢？

增加一个可以实例化并调用控制器方法的文件。





逻辑流程：



### 动作action分发参数：a

如何做到一个前端控制器，可以调用一个控制器类不同方法动作呢？

在请求前端控制器index.php时，向其传递a参数，表示当前所需要执行的动作名，例如：

功能：比赛列表：

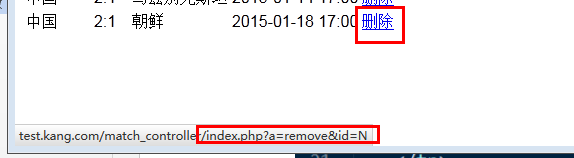
Index.php?a=list

功能：比赛删除：

Index.php?a=remove

Tip：链接地址的形成，应该在HTML代码中就确定好了，再存在一个默认动作即可！

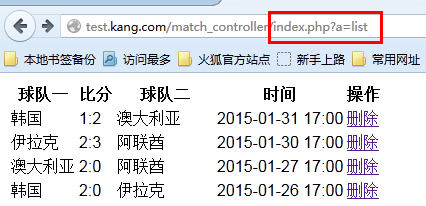


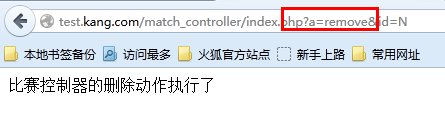


Index.php判断a参数，执行相应的动作即可：



测试：





### 控制器controller分发参数：c

如果需要执行其他控制器的某个动作应该处理处理前端控制器？

在请求前端控制器index.php时，向其传递c参数，表示当前所需要执行的控制器类名，例如：

比赛的列表动作：

Index.php?c=Match&a=list

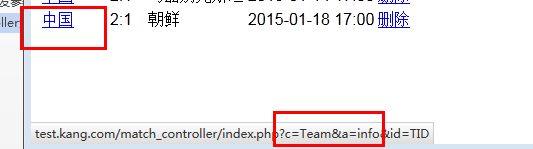
球队的信息动作

Index.php?c=Team&a=info&id=TID

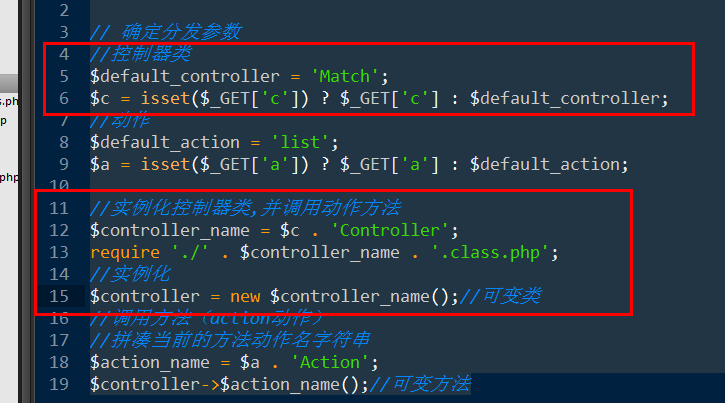
c,a在HTML的链接地址中，自动形成好的







Index.php对c分发参数进行处理：



测试：



### 使用常量存储分发参数

思考：

在一次请求周期中，所请求的控制器名（当前控制器），和所请求的动作名（当前动作），是否会发生更改？

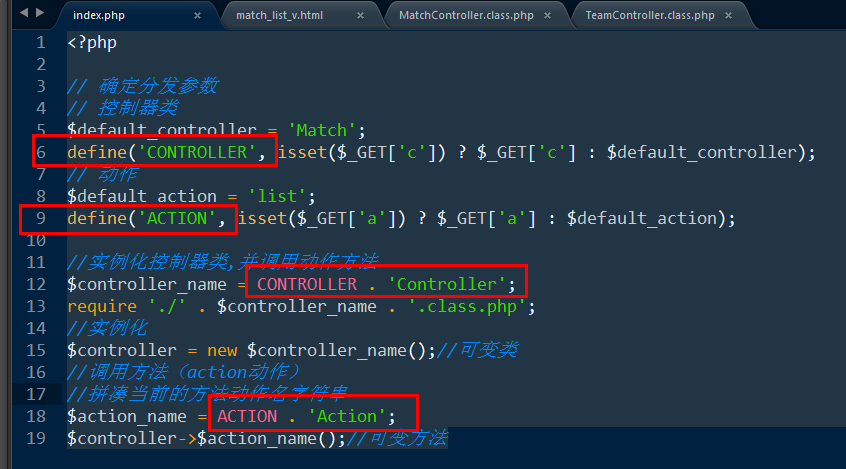
不会发生更改！

如何在语法的层面上，保证在一次请求周期内，当前控制器与当前动作不会发生改变？

存储在变量中，不能保证。

应该使用常量，进行存储当前控制器及其当前动作，保证

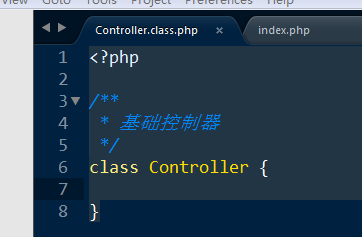
Index.php中实现：



## 基础控制器

增加为 所有的控制器提供基础代码控制器类：

Controller



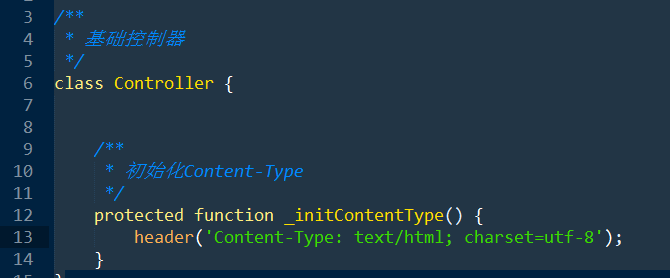
其他控制器，继承自基础控制器：





### 初始化Content-Type的基础操作

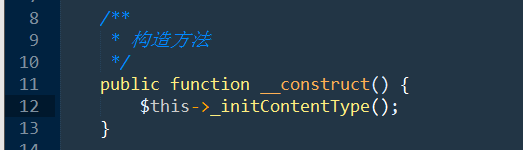
Controller.class.php



调用：

在实例化控制器对象时调用，在构造方法中：

Controller.class.php



## 逻辑图例



注意：

浏览器的请求地址（URL），都是固定的形式：

Index.php?c=Controller&a=action&

称之为单入口模式

项目对外提供的任何功能，都是由某个控制器类的某个方法来实现的。

# 目录布局

存在项目的根目录 shop34

## 框架代码与应用程序代码划分

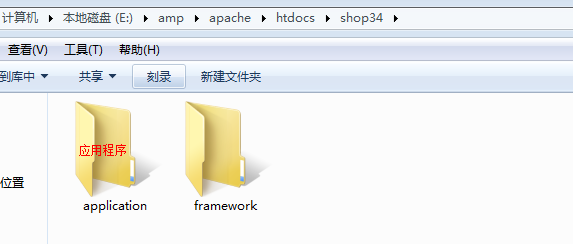
框架代码：

在各个应用程序间可以通用的代码。

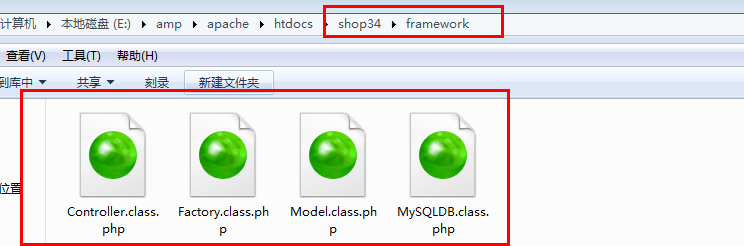
应用程序代码：

当前项目的业务逻辑实现的代码。

分别创建两个子目录存储，框架与应用程序代码：



示例：框架内的代码：



## 入口文件在站点根目录

