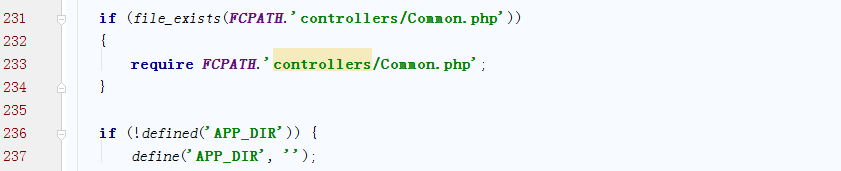
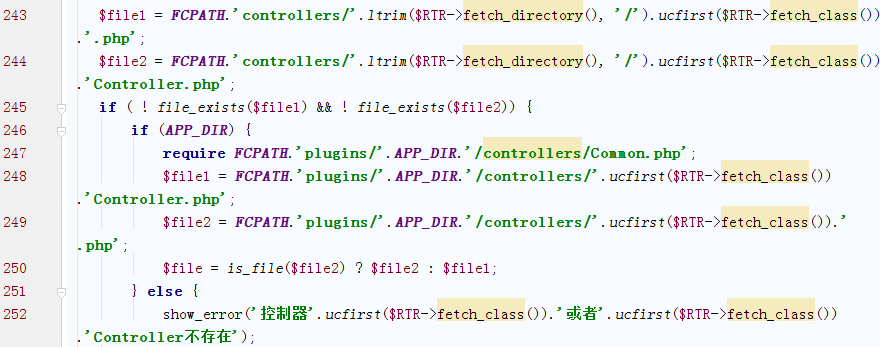
**CodeIgniter.php**

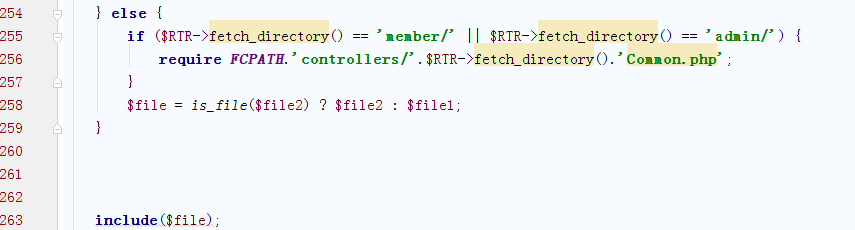


引入controller文件夹下的common.php文件，此处还定义APP\_DIR

|  |
| --- |
| FCPATH为入口文件index.php所在的目录，即，整个项目的根目录 |

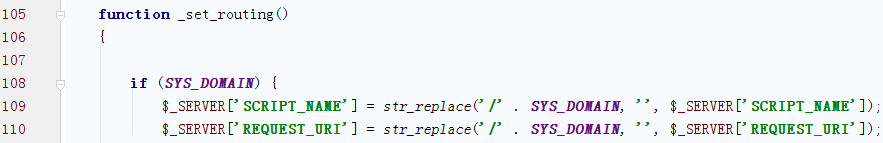


获取控制器（分别获取有和没有Controller后缀的类名），如果不存在，又定义了插件目录，引入插件中的common.php和对应的控制器

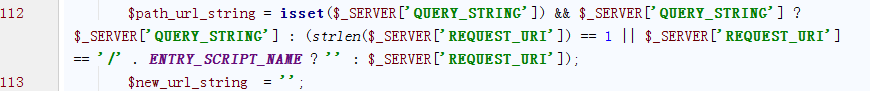


如果控制器存在，访问控制器的目录为admin或member的话，引入其中的common.php，其中，有限引入待Controller后缀的控制器

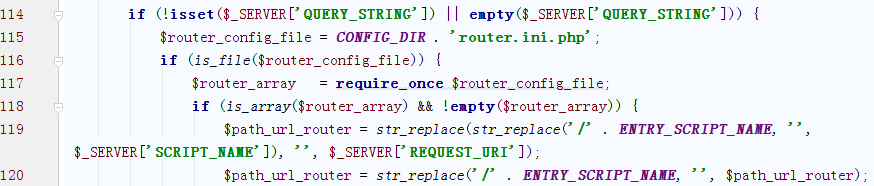
**Router.php**



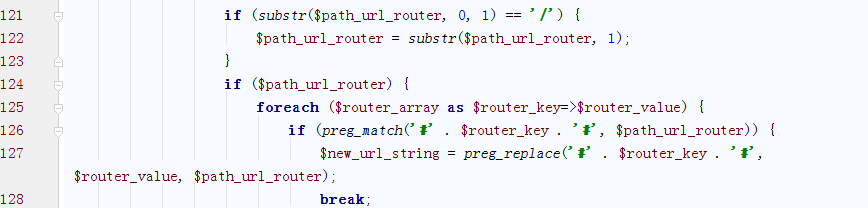
如果存在域名目录（虚拟主机？），去掉



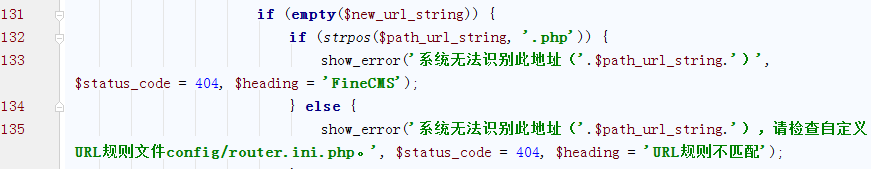
如果有query，设置，否则为按/分割的uri



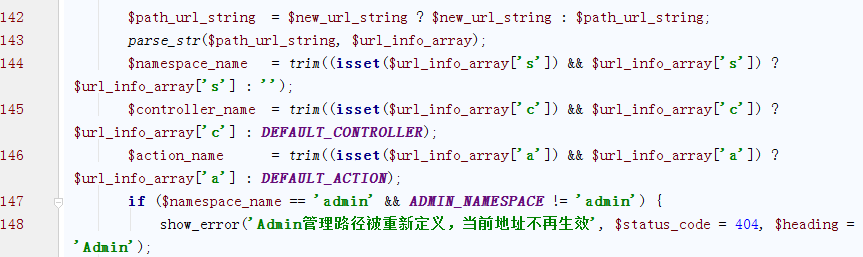
如果没有query，获取路由配置，将uri中的入口文件等去掉



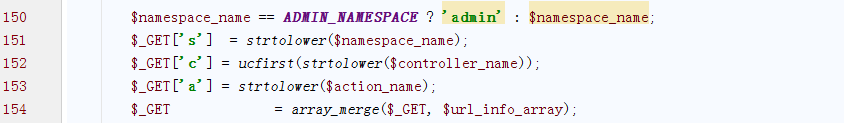
去掉头/，匹配路由



如果没有匹配项，报错



解析uri，分别获取s，c，a等（s为命名空间，看来仅指admin，后面看不是）

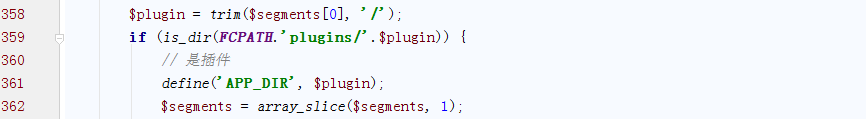


报错到get中

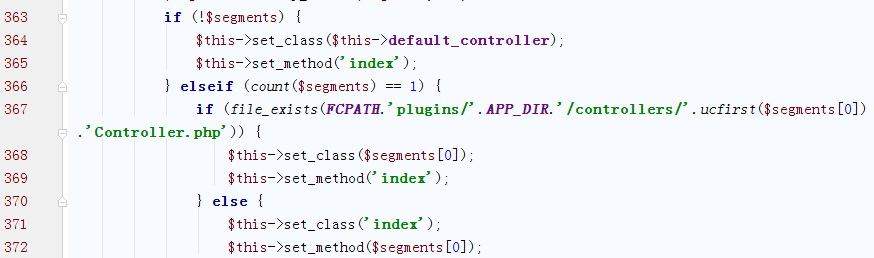


如果有s，如果为admin，设置，否则为原始值

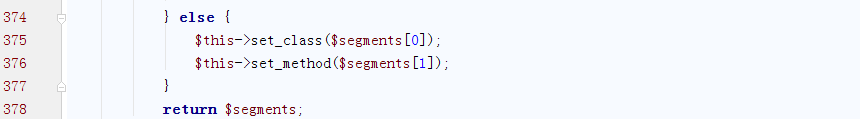
================================================================================



如果插件目录存在，定义APP\_DIR，从segment中取出plugins后面的元素



如果不存在后面的元素（即不再有子目录），获取默认控制器和方法，否则分别获取控制器和方法



如果既有类、方法，设置

**Common.php**

区别在于引入dayrui/libraries/drivers目录下的几个核心文件

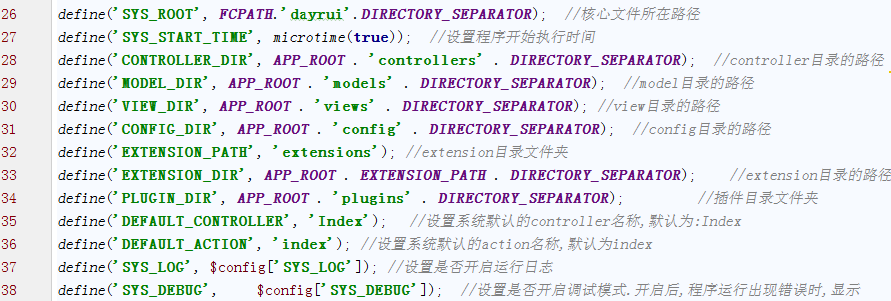
**index.php**

区别在于

（1）APPPATH：原为application/，限制为extensions/

（2）BASEPATH：原味system，现为dayrui

（3）引入config/config.ini.php，并定义大量常量





**1.配置**

没有config数据表，所有配置均保存在文件中，貌似auth.role.ini.php和version.ini.php可以编辑，其他不行

**2.缓存**

使用dayrui/libraries/cache\_file.class.php，构造函数中，设置普通缓存保存地方为data文件夹

而系统使用cache/driver/cache\_file.class.php，区别在于放到了1个单独的data文件夹中（没必要，扔使用系统的）

**3.视图**

思路：首先将模板解析为缓存，在调用系统的编译

有必要继续使用，用于替换模板，生成第1次缓存文件，然后再传给系统的view()方法

**4.模型**

使用dayrui/libraries/driver/Model.php，区别在于里面有缓存设置，也没必要

**问题处理**

1.系统架构

（1）mvc分层，控制器仍在controller中，公开，其他在thirdparty/some文件夹中。some文件夹包括core（core1，core2…），config，services，libraries，templates，views。services包括bm，dao2层，加密。未编译的视图层公开。

2．视图文件

可以有多套视图文件，放在template文件夹下，默认为default文件夹，自定义设置自己命名，在配置中设置。编译后的文件放在views文件夹下（使用add\_package\_path()，第2个参数为true时，系统编译时会自动读取）。

所有控制器最后调用视图文件的方式是$this->view(‘视图名’);，view()方法在父类中定义，先解析文件，在调用$this->load->view();

3.钩子

视图是先从设置的theme中找，没有才从default中找。在控制器中运行对应的钩子（钩子名不用写，与控制器类名一致，就自动运行了）。也可以自设钩子。

4.配置常量

///////////可以在系统文件config/constants.php中定义，在最开始执行的时候就会加载，主要定义如下常量：

既然不该系统，那么自定义常量主要是自己使用，在控制器最开始定义即可。