

系统架构设计文档

系统名称	系统架构设计文档
项目负责人	
作者	王严博
文档提交日期	2010-4-13

版本信息

版本号	更改内容简介	日期
0.1	Draft	2010-4-13
0.9	Updated	2010-4-19
1.0	Release	2010-4-20

1 背景

为了解决 Web 开发过程中的多人并行开发，问题准确快速定位，快速开发 web 程序等问题，设计此软件框架。此框架运用 RESTful 的风格，基于 MVC 模型。

2 名词解释

- MC(memcached) is a high-performance, distributed memory object caching system, generic in nature, but intended for use in speeding up dynamic web applications by alleviating database load.
- REST is a term coined by Roy Fielding in his Ph.D. dissertation to describe an architecture style of networked systems. REST is an acronym standing for Representational State Transfer.

3 系统设计

3.1 系统架构图及说明

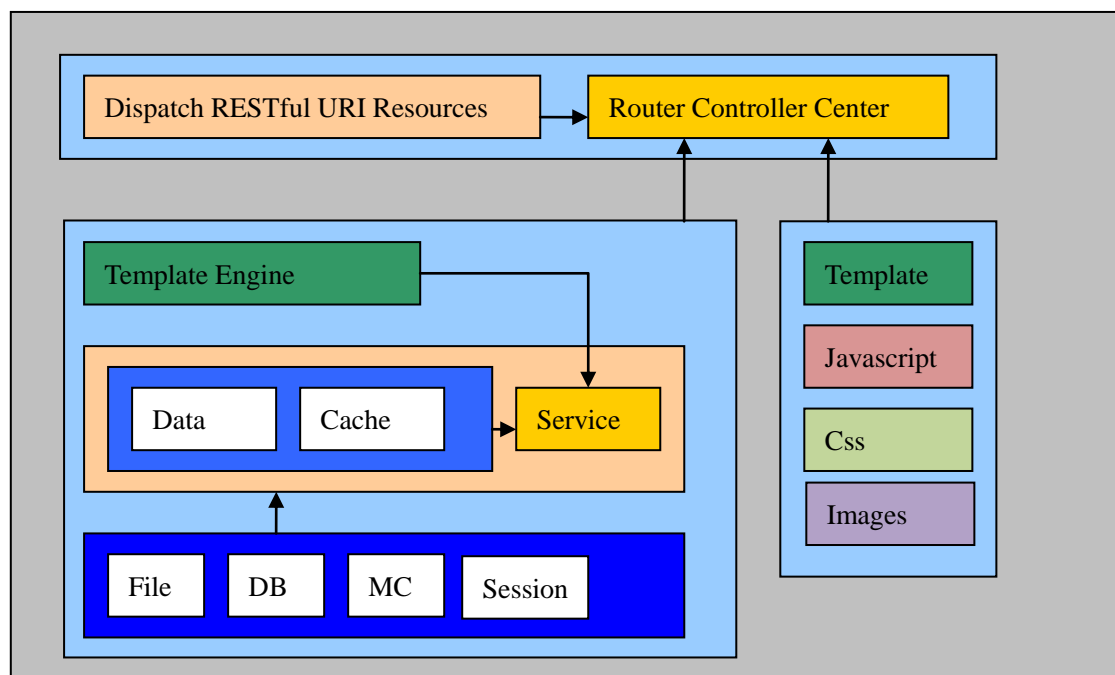


图 3-1 系统架构图

如图 3-1 所示，本系统中包含三大块，控制层（URI 的解析和资源的映射），数据模型层（底层基础模块、逻辑应用模块、模板引擎模块和服务层），视图层（基础模板，前端 javascript 脚本和 Css 等）。

控制层（Controller）包括 URI REST 风格的驱动模式，驱动相应的控制类模块，控制类模块根据业务逻辑进行拼装起 Model 和 View，形成一次用户的交互行为。

模块层（Model）分由逻辑模块和基础库数据模块组成。其中逻辑模块层主要是为控制

层的具体应用提供具体的数据应用接口，密切的和上层应用的逻辑相关；基础库模型层则为逻辑模型层提供接口（例如文件、数据库、Cache、Session、HttpSocket、Socket、Router、File、Folder，View Engine（Smarty）等模块），这样做是为了让整个系统逻辑层和基础模块层开发分开，两者仅用调用接口建立联系，整个底层模块或者上层应用模块的更新升级，透明化。

视图层（View）主要包括基础的模板文件组成，还有操作模板内对象的 JavaScript（根据目前的使用情况，后续全部统一为 jQuery）和 CSS。

3.2 系统目录结构及相关规范

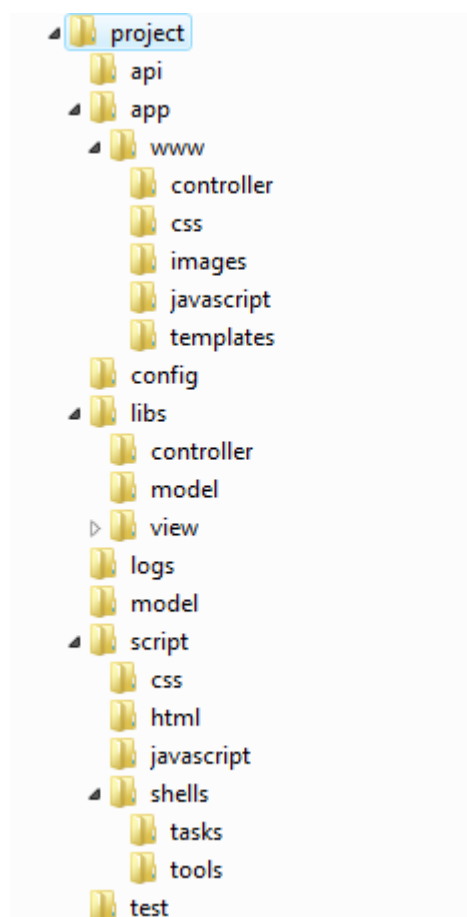


图 3-2 系统目录结构

如上图 3-2，为后续开发系统的目录结构。其中/project/libs 为整个系统的核心库，包含所有基础模块（DB，Cache，View Engine 等）。/project/model 下存放业务逻辑的基础模块类。/project/app/下放置开发项目（比如 www），也可以随便的增加别的新项目，app 目录下由自己项目的 controller,css,images,javascript,templates 组成。/project/index.php 为整个系统的唯一入口。/project/script/下放置公共的 css,javascript 等。

为了增强程序的可读性和可维护性，要求以下几点：

- 整个系统全部基于 OOP 进行开发，所以不要进行过程性的显式代码调用。所有的共有代码尽可能的按照应用进行类封装。
- 整个系统的文件命名采用大驼峰式命名法（控制类名：UserController.php，基础类：User.php，普通文件：DBConfig.php），方法名，函数名采用小驼峰式命名法(getUser)。其中 libs 下的文件必须冠以 Base 前缀和各自子目录名（controller，model，view 等，eg：BaseController.php,BaseModelDB.php）。Model 下的文件命中直接命名即可，但

是不要包含 Base 和 Controller 即可。App 下的 controller 目录下要包含 controller 单词。模板名称也是一样 (Common.js, User.html, Home.css)。采用这样的命名方式之后, 需要注意在程序中名称引用一定要正确, windows 上文件是不区分大小写, 但是服务器 Linux 上是区分的, 所以在使用 windows 做开发机要特别主要文件引用, 免得在服务器上出错。

- 变量命名统一采用小驼峰的方式, 例如 \$userName, \$_keyBuffer
- 缩进排版, 4 个空格为缩进排版的一个单位。
- { 和关键词在一行, } 单独占一行。

```
<?php
    if ($i < 10) {
    }
    else {
    }
?>
```

- 文件夹一律小写完整单词。
- 除了以上说的之外, 其他规范不做强制要求。
- 此系统中对类对象, 直接使用, 不需要 require 或者 include 具体文件。

4 风险评估及对其他系统影响

4.1 在这里加入对已知的或可预知的风险的评估

4.2 在这里加入对其他系统或模块可能的影响。