

目前只修改入口的index.php、引导文件z.php、路由router.class.php，其他的大部分是旧的东西，README.txt之类的了解一下就算了

0、自动加载方法：采用“一级目录映射+二级目录映射+真实类名+后缀名映射”的类名规则来加载类文件。

1、框架强制使用URL重写，所有请求统一重写为index.php?s=...

2、入口文件夹为entry，只有index.php、favicon.ico、robots.txt三个文件。其中index.php需定义默认的应用目录名appDir，并引入框架引导文件

3、框架路由决定了url可以有三种结构模式，并且可以任意选择其中一个访问

4、url参数e、c为基本缺省参数，其中e表示入口或模块名称，c表示控制器名称。

5、入口或模块名e对应app目录下的同名文件夹（index作为缺省值时，指向到入口文件定义的目录。同时，e不能使用public和extend）

6、控制器c对应app/e目录/controller下控制器，控制器必须实现main方法，有需要的话可以实现delay方法，url不接受方法名，以保证一个控制器只做一件事

7、控制器如果作为一个接口，则其不需要依赖模版类，而是依赖框架接口类，并且使用统一的接口规范（之前做法是混合的，并不好使）

8、控制器如果作为一个页面，依赖模版类，而不依赖接口类。并且将解析view下的同名模版文件（组件化后，这个模版相当于页面骨架，这里也只需要解析组件引用的语法即可）

9、组件component目录下只有视图文件，模版类获取到组件后，加载组件类进行解析（没有使用组件的话相当于无组件的单页写法，如果需要支持这种写法的话，需要在模版类增加基本的语法解析，不加载组件类是基于最小化依赖的考虑）

10、组件解析使用一套特别的类名加载方法（不是自动加载方法）。

如类名comUserByDbtestClsForSearchList。

它表示使用component/dbtest/user/searchlist.tpl这个组件模版。

调用model/read/dbtest/user.class.php类的searchlist方法构建组件数据模型。

（即com表示组件，cls表示后缀，以by和for分割成三部分，分别对应文件夹和文件名/方法名。基于集群考虑，这里建议使用tableNameByDbNameForMethodName形式，对应read/write下也使用dbname/tablename.class.php以方便管理，如果是单一库，可以使用其他的两层目录分类以到达便于管理维护的目的）

改为

{component:dbtest/user/searchlist|rDbtestCls}

11、tmpfs用来存放读写频率较高，但又不适合存到数据库的信息文件，结合linux的虚拟内存系统，可以大幅提升读写速度，非linux仅限于方便管理

12、短地址映射文件存放在tmpfs/urlmap目录下，短地址的e、c参数记录在文件中，因此就短地址访问本身而言是不分入口/模块的，这里会出现同一个短地址，不同域名都能访问到同一应用的情形。和同一个域名的子入口/模块部署了一个二级域名或其他域名的入口指向是同样的道理，不能算是一个BUG，但如果没有意识到的话，某些情况下也很致命。基于这种情况，入口index.php可能还需要定义允许访问的应用目录名！

13、模版解析将产生编译文件和静态文件，其中编译文件存放在tmpfs/compiled/对应的e目录下。静态文件保存在tmpfs/cache/对应的e目录下。

旧的写法是把静态文件和动态编译文件的处理写到了一块，这里需要分离出来做一个单独的缓存类。动态编译文件的相关方法将合并到模版类中。静态文件的相关方法将和redis组合构成一个新的缓存类

14静态资源的分离：思路是在app下建立一个静态资源的管理入口/模块，定义静态资源所在空间（同entry的道理一样，可以是框架内外），并且将一个静态资源的域名解析到这个文件夹。考虑到可能需要结合前端来决定静态资源的目录结构，暂时还没实现。