纵横笑谈laravel

1.laravel为什么受欢迎

在众多的框架里面laravel凭借着优雅、功能多、组件化，使其成为最受欢迎的PHP开发框架。

组件化和可扩展性

Laravel注重代码的组件化和可扩展性。你可以在包含超过5500个程序包的Packalyst目录中找到你想要添加的任何文件。Laravel的目标是让你能够找到任何想要的文件。

优雅的架构设计（中间件、路由、Artisan 命令行、代码迁移、假数据填充），基于接口的形式采用内容开发。

开发模式 : 混编模式-->MVC设计模式-->分层模式-->服务组件化设计

2.分层设计

分层设计根据具体物理逻辑做的分层设计，由每一层负责对应相关业务功能的开发。从而来解决MVC下的臃肿问题。

分层设计大致分为：视图(View)、模型(Model)、控制器(Controller)、服务层(Service)

缺点：

代码多,结构也会开始复杂。代码复用性差、太臃肿,管理问题,扩展问题
开发的时候尽量一个类一个职责, 一个方法一件事,一层只做相同事情。例如：服务层只是做服务的处理。
如果再来一个项目, 项目中工具类,支付方式，日志处理基本相同。那这个时候只能
ctrl+C到ctrl+V 进行代码的移植。

13.服务组件化设计

基于分层设计的问题所以就衍生了服务组件化的内容。而composer 组件复用性本身就是特别高的。凡是包中存在composerjson文件就意味着,把composer组件包转化为了laravel框架的一部分，从而把功能业务转化为了可拆分的组件。

并且这些组件也可以在其他项目中直接下载运用的
设计模式运用=>组件:服务提供者

4.横看laravel的服务

功能业务：控制器模型,试图,路由,中间件，事件,门门面,契约,服务提供者服务容器,Auth,队列

服务容器: I0C容器（ Container）容器就是存储框架各个服务的内容。
容器->服务容器(I0C容器)
管理项目运行的服务

服务提供者:组件ServiceProvider
1.注册一些实例对象到容器中
2.初始化组件->让laravel能够加载这个组件

门面:作用是什么？

Facades实际是服务容器中底层类的「静态代理」, 相对于传统静方法，在使用时能够提供更加灵活、更加易于测试、更加优雅的语法
快速的从容器中解析实例类并使用DB::table

契约:作用是什么。 合同,约束,协议
interface特点是什么?

必须被实现没有方法体,只是声明，不能实例化,可以多个接口。

abstract:
可以单继承,能写方法体

服务组件化:必然laravel会提供服务, 但是组件是不是会升级
组件升级之后,怎么保证方法一致?

契约:给laravel服务提供一套标准, 然后laravel所依赖组件必须实现
那可不可用抽象类代替?
不能，标准制定不是一个人事情，多个人- -起制定