项目代码结构讲解

一．项目代码讲解

1. 项目代码结构

代码基于MVC的核心概念（答辩常问）：

代码主要为MVC的架构 M：代表实体层，是负责和数据库交互的实体层

V：视图层，主要为用户展示页面的功能

以及向后端传递前端数据和参数，展示前端页面的视图

C：控制器层，主要功能为控制前端向后端传参和后端向前端传递数据，代码中为web层，内部为controller的代码，目的就是控制前后端的数据交互。

2.代码层级分析

后端：

src：存放后端代码的文件夹

annotation:自定义注解的层

bean:自定义传给前端返回值的返回类型的层，比如我们给前端传参的时候需要定义一些code来确认是否是返回成功，在这里我们定义此类，并且此类里还有分页等

entity：实体类层（也可以叫model），实体类主要作用为存放我们代码中所需要的各种数据库字段的变量，然后通过java进行和后台数据库的交互

因为java语言本身就是面向对象的，所以在实体层建立对象去和数据库内的字段信息进行交互。

举例：在实体类中需要拿到数据库的用户密码字段，那么在entity的代码里写上

1611980623

另外在实体层中，如果我们后续的代码想要调用实体类的数据，如何获取。这里要在实体类中加入get/set方法，通过英文含义我们可以理解为，如果取实体类的值，就用get方法，如果想设置为一个新的值，用set（通俗理解，具体get/set方法如何使用可以百度）

此项目我们使用JPA，所以在注解上会有很多JPA的注解来确认表/库等

dao：数据访问层（也可以及叫repository或者dao层），主要作用如名，进行数据访问时对数据库进行操作的层，可以这么理解，当我们需要从数据库获取数据时，

需要通过java建立与数据库进行交互的代码，在这个代码我们要通过resource中定义的application-dev.properties中的一些创建数据库连接的标签，定义数据库的连接/账号密码/url等信息，然后通过这些信息建立与数据库交互的连接。

简单来说就是：在application.properties中写好数据库的配置信息，代码通过读取它来获取与数据库的连接

Common层：此层代码里放置webMvcConfigurer,拦截请求

controller：控制器层：主要功能就是获取service层传递过来的数据，然后向前端传递，或者从前端接到数据，向service传递前端参数以及数据

constant:主要放置一些代码里定义的常量

interceptor:设置一些拦截器拦截定义的前端请求

举例：如何向传递数据：通过注解里的url路径来请求或者发送需要的接口数据。

service : 业务处理层，主要功能为处理业务代码，比如进行增删改查操作写好具体的方法，service会通过具体方法去调用数据访问层的代码，serviceImpl是业务逻辑处理的实现类，实现类实现具体方法，而service本身是个interface接口，不是类，这样写的目的是为了给维护后续代码(Mapper层同理)

util：公共方法类，放置一些公共方法。

JPA教程

https://www.w3cschool.cn/java/jpa-entitymanager.html

前端：

前端所有资源都在resource中,包括静态资源文件和页面文件

看代码的逻辑：

建议后端代码通过controller -> service -> serviceImpl -> dao-> entity这个逻辑去查看代码

调用逻辑为controller类通过接受到前端的请求然后去调用service层，service通过业务处理代码，此时找到具体的dao的接口，dao通过JPA进行和数据库的交互，然后将数据库的操作后的值都封装在entity中进行存储实体。

前端看代码的逻辑，首先我们想看哪部分的页面如何实现，就在哪部分的页面，用谷歌浏览器的F12功能来查看代码。