

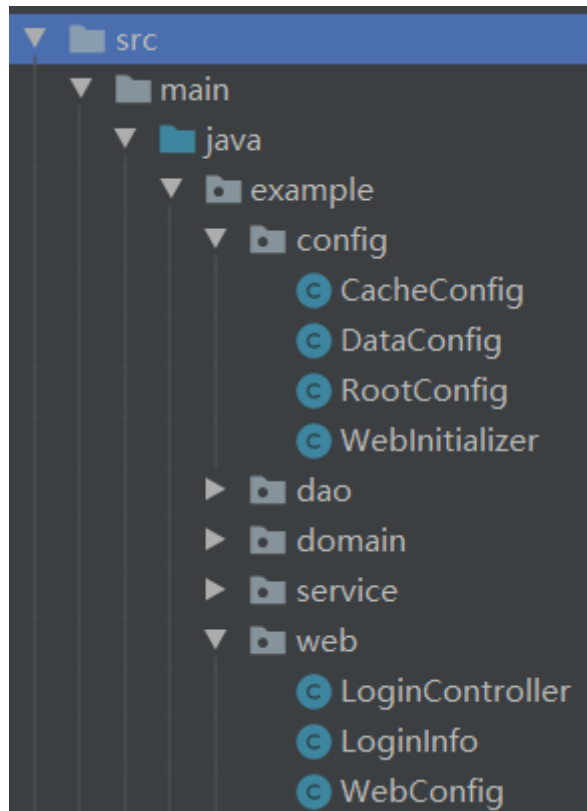
# 综合作业报告

## 实验要求及完成情况

所有实验要求均完成，具体情况如下：

1. ☒ 清空web.xml配置，改成java配置类

通过添加 CacheConfig、DataConfig、RootConfig、WebInitializer、WebConfig几个类对项目运行所需要的环境进行了配置，从而清空web.xml文件，如图所示



2. ☒ 优化控制层的loginCheck方法，并且添加数据校验功能（@Valid），用户名3-6个字符，口令6个字符，并有错误提示

- 在 loginCheck 方法添加了数据校验功能，并修改了该方法的返回类型为String即视图名

```
@RequestMapping(value = "/loginCheck.html")
public String loginCheck(HttpServletRequest request, @Valid LoginInfo loginInfo, Errors errors, Model model) throws Exception {
```

- 能够在出错的情况下在网页进行错误提示

- 输入不符合校验规则时：

username must be between 3 and 6 characters long.  
password must be 6 characters long.

用户名:

密 码:

- 用户名或密码错误时：

Incorrect username or password

用户名:

密码:

3. ☒ 数据库由原来的mysql改成H2内嵌数据库，不要有外部数据库访问依赖，`exampledb.sql`数据库脚本同步修改

通过 `DataConfig` 配置类对数据源以及所使用的数据库进行了配置；`exampledb.sql` 同步修改为符合H2语法的数据脚本

4. ☒ DAO层实现由现在的jdbc改成JpaRepository自动实现，方法名可以改变  
DAO层的类通过继承 `JpaRepository` 类进行了实现

5. ☒ DAO层的 `findUserByUserName` 添加缓存功能，缓存用EhCache实现

- EhCache实现在 `CacheConfig` 类中进行了配置
- `findByName` 方法通过增加 `@Cacheable` 注解添加缓存功能

6. ☒ 测试改进：service层的测试将现在直连数据库改成使用mock取代dao层

通过 `@RunWith(MockitoJUnitRunner.class)` 注解以及对DAO层添加 `@Mock` 注解使测试时不用直连数据库即可完成service层的测试，测试结果如图：

