文件结构

```
\Redis_MySQL_web\src\main
|
├─java
|  └─com
|      └─example
|          └─redis_mysql_web
|              |  RedisMySqlWebApplication.java    // 项目启动文件
|              |
|              ├─cache                            // 缓存预热
|              |      RedisWarmUp.java
|              |
|              ├─config      // 存放配置相关的类，配置Redis、Redisson等的相关设置
|              |      RedisConfig.java
|              |      RedissonConfig.java
|              |
|              ├─controller  // 控制器层，接收用户请求并返回响应，负责处理具体的
HTTP 请求
|              |      CourseController.java
|              |      GradeController.java
|              |      StuController.java
|              |
|              ├─mapper      // 数据访问层，负责与数据库的交互，使用 MyBatis 等框架
实现数据操作
|              |      CourseMapper.java
|              |      GradeMapper.java
|              |      StuMapper.java
|              |
|              ├─pojo                  // 存放项目中的实体类，定义数据库中的表结构
|              |      Course.java      // 课程 (course_information关系表)
|              |      Grade.java       // 成绩 (student_grade关系表)
|              |      Stu.java         // 学生 (students关系表)
|              |
|              ├─service     // 服务层，负责业务逻辑的处理定义了具体的业务接口和实现
类
|              |  |  CourseService.java
|              |  |  GradeService.java
|              |  |  StuService.java
|              |  |
|              |  └─impl     // 存放接口的实现类，具体实现了服务层定义的业务逻辑
|              |          CourseServiceImpl.java
|              |          GradeServiceImpl.java
|              |          StuServiceImpl.java
|              |
|              ├─util        // 存放工具类，提供辅助功能的实现
|              |      CourseBloomFilterUtil.java
|              |      GradeBloomFilterUtil.java
|              |      StuBloomFilterUtil.java
|              |
|              └─website    // 存放前端静态文件和配置类
|                      Course.html
|                      Grade.html
|                      Stu.html
```

```
|                       WebConfig.java
|
└─resources
    |  application.yml                      // Spring Boot 的配置文件，定义了项目的配置信息
    |
    └─mappers                               // MyBatis 的映射文件，定义数据库操作的 SQL 语句
            CourseMapper.xml
            GradeMapper.xml
            StuMapper.xml

```

Quick Start

1 集群配置

1. `conf_cluster`: 集群的配置文件 (3主3从)
2. `cmd.txt`: 集群创建与启动命令
3. 上述的文件位于`files`文件夹中，并需要修改其中文件的ip

2 数据库建立

数据库文件位于`/files/stu.sql`，直接导入即可创建需要的数据库

3 项目配置文件

需要修改application.yml的ip

```yaml
  data:
    redis:
      cluster:
        nodes:
          - 192.168.178.23:6380
          - 192.168.178.23:6381
          - 192.168.178.23:6382
          - 192.168.178.23:6383
          - 192.168.178.23:6384
          - 192.168.178.23:6385
```

4 页面启动

需要修改端口号:

1. 先打开界面，如 StuHtml.html，可以从地址栏里获得端口号 (localhost:后即为需要的字符串)
2. 将`WebConfig.java`中的端口号进行对应的修改 (内容如下)

```java
public class WebConfig implements WebMvcConfigurer {
    @Override
    public void addCorsMappings(CorsRegistry registry) {
        // 不太了解后端这部分的内容，因此电脑每次重启后都需要先修改该类
        // 需要先打开网页查看当前url的端口号，然后修改localhost:所对应的端口号
        registry.addMapping("/**").allowedOrigins("http://localhost:63342");
 // Replace with your frontend URL
    }
}
```