



**QG工作室项目报告**

学 院 信息工程学院

专 业 电子信息类

班 级 电信4班

组 别 后台组

姓 名 颜钦煜

学 号 3123002217

2023年 月 日

广东工业大学计算机学院制

目录

[1. 项目简介 1](#_Toc30766)

[2. 设计思路 2](#_Toc24252)

[3. 详细功能设计 3](#_Toc20914)

[4. 程序测试 3](#_Toc7265)

[5. 项目亮点 3](#_Toc19276)

[6. 心得体会 3](#_Toc25747)

# 项目简介

这在线学习平台旨在为学生和教师提供一个灵活和互动的教育体验。学生可根据个人兴趣和学习需求，在平台上浏览、选择教师开设的多样选修课程，并通过课程章节学习、在线答题等方式深化知识。教师则能够轻松创建和管理课程内容，监控学生学习进度和表现，包括课程章节的得分情况、平均分统计以及学生的学习记录和反馈。这一平台旨在促进教学效率和学习成效，为教育教学提供创新的解决方案。

# 设计思路

本项目依托Javase,javaweb,mysql,tomcat等技术进行开发,整个项目分为前端，后台，与数据库交互三个部分。

1. 前端:使用element ui和Vue框架进行开发
2. 后台:基于baseServlet进行开发
3. 与数据库交互：基于JDBC技术和动态代理，反射等思想设计类MyBatis来实现与数据库交互

技术选型

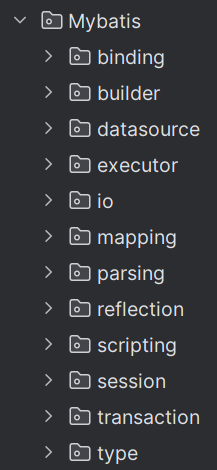
前端：HTML，CSS，JavaScript，Vue，Element Ui

后端：Javase，MySQL，Maven, Servlet,

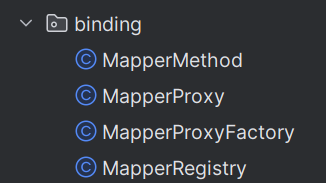
前后端交互：AJAX。

# 详细功能设计

1. 手写MyBatis框架：分为以下部分



一.映射:



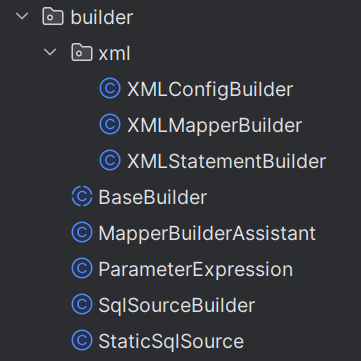
MapperMethod:判断方法的类型（如Select,Update）,并获取方法的参数

MapperProxy: 负责实现 InvocationHandler 接口的 invoke 方法，最终所有的实际调用都会调用到这个方法包装的逻辑。

MapperProxyFactory 是对 MapperProxy 的包装，对外提供实例化对象的操作。当我们后面开始给每个操作数据库的接口映射器注册代理的时候，就需要使用到这个工厂类了。

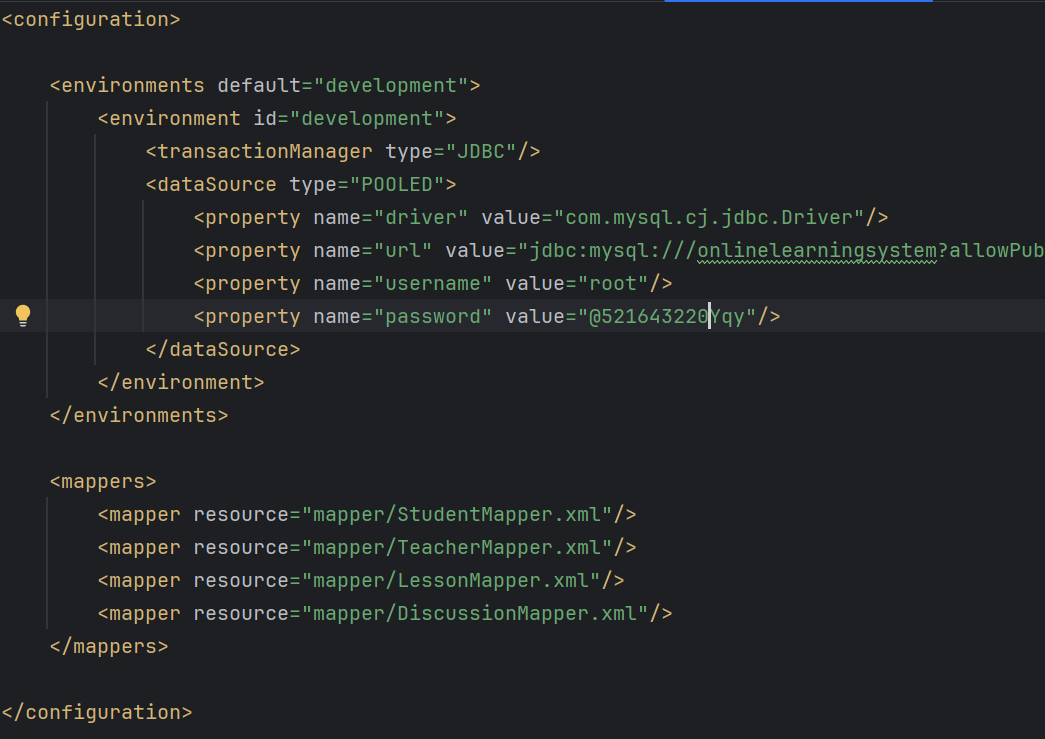
MapperRegistry 提供包路径的扫描和映射器代理类注册机服务，完成接口对象的代理类注册处理，

二:解析mapper文件

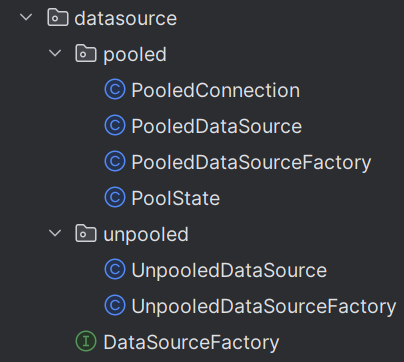


使用dom4j和xpath来实现解析mapper文件，

MyBatisConfig.xml文件如下：



三：数据库连接池



# 在项目中，包括两部分分为无池化的 UnpooledDataSource 实现类和有池化的 PooledDataSource 实现类，池化的实PooledDataSource 以对无池化的 UnpooledDataSource 进行扩展处理。把创建出来的链接保存到内存中，记录为空闲链接和活跃链接，在不同的阶段进行使用

使用PooledConnection ，通过invoke方法的反射调用，对关闭的链接进行回收处理，并使用 notifyAll 通知正在等待链接的用户进行抢链接。用户想要获取链接的时候，则会从连接池中进行获取，同时判断是否有空闲链接、最大活跃链接多少，以及是否需要等待处理或是最终抛出异常

1. 数据库表格设计：

Student:

1. Id
2. Username(姓名)
3. Password(密码)
4. PersonInformation(个人介绍)
5. StudentNumber(学号)
6. Grade(年级)

Teacher:

1. Id
2. Username(姓名)
3. Password(密码)
4. PersonInformation(个人介绍)
5. qq(QQ号)
6. Email(邮箱地址)

Lesson:

1. Id
2. Name(课程名)
3. TeacherName(教师名)
4. InformationId(课程详细信息Id)
5. CurrentEnrollment(当前报名人数)
6. RegistrationPopulation(总报名人数)

Chapter:

1. Id
2. Name(章节名)
3. LessonId(章节所属课程的id)
4. Information(章节信息)
5. ImagePath(图片路径)

Question:

1. Id
2. Name(问题名)
3. ChapterId(问题所属章节的id)
4. Information(问题信息)
5. ImagePath(图片路径)
6. QuestionType(问题类型)
7. RightAnswer(正确答案)
8. Score(分数)

LessonInformation:

1. id
2. BeginTime(课程开始时间)
3. Information(信息)
4. EndTime(课程结束时间)
5. Name(课程名)
6. RegistrationPopulation(总报名人数)

Discussion:

1. id
2. DiscussName(评论人姓名)
3. Information(评论信息)
4. LessonId(评论所属课程Id)
5. Time(评论时间)

RegisterStudent:

1. id
2. StudentName(学生姓名)
3. LessonId(课程Id)
4. GetScore(在该课程的得分)

CurrentChapter:

1. id
2. StudentName(学生姓名)
3. ChapterId(章节Id)
4. ChapterFinish(章节完成程度)
5. Information(章节信息)
6. LessonId(课程Id)
7. GetScore(章节得分)
8. ImagePath(图片路径)

CurrentQuestion:

1. id
2. StudentName(学生姓名)
3. QuestionFinish(问题完成程度)
4. Answer(学生提交的答案)
5. Information(问题信息)
6. QuestionId(问题Id)
7. GetScore(问题得分)
8. ImagePath(图片路径)
9. ChapterId(章节Id)
10. Utils工具类：
11. TimeUtils:此工具类的方法有
12. 获取现在时间并生成为格式为 年-月-日的字符串；2.接收开始时间与结束时间，判断现在时间是否在区间之内
13. SaltMD5Util：

generateSaltPassword(String password):

功能： 生成经过盐加密的密码。

参数：password：待加密的密码。

返回值： 返回一个包含盐和密码哈希值的字符串。

实现：首先生成一个随机的16位盐，然后将盐和密码拼接起来进行 MD5 哈希处理，最后将盐和哈希后的密码混合在一起返回。这样生成的字符串包含了盐和密码的哈希值，提高了密码的安全性。

verifySaltPassword(String password, String md5):

功能： 验证经过盐加密的密码是否与明文密码匹配。

参数：password：待验证的明文密码。

md5：经过盐加密的密码字符串。

返回值： 如果密码匹配，则返回 true；否则返回 false。

实现：从经过盐加密的密码字符串中提取出盐和哈希后的密码，然后使用相同的盐和明文密码进行哈希处理，最后将哈希后的结果与提取出的密码哈希值进行比较，以验证密码是否匹配。

md5Hex(String src):

功能： 对字符串进行 MD5 哈希处理。

参数：src：待处理的字符串。

返回值： 返回经过 MD5 哈希处理后的字符串。

实现：使用 Java 的 MessageDigest 类获取 MD5 摘要实例，然后对输入字符串进行字节流的 MD5 哈希处理，最后将处理后的字节数组转换成字符串返回。

这些方法结合了盐加密和哈希处理的技术，提高了密码存储的安全性，并且通过验证方法可以有效地验证用户输入的密码是否正确。

3.JwtUtils：方法有生成Jwt令牌，解析Jwt令牌

4.Filter:

LoginFilter是一个Java Servlet过滤器，用于对Web应用程序中的请求进行过滤，以确保只有经过身份验证的用户才能访问受保护的资源。该过滤器实现了Filter接口，并覆盖了init、doFilter和destroy方法。

在doFilter方法中，首先获取了HTTP请求对象，并提取了请求的URL。然后，它定义了两个字符串数组urls，分别存储了允许直接访问的URL模式。如果请求的URL匹配到了这些允许访问的模式之一，则直接放行，不进行进一步的处理。

如果请求的URL不属于允许直接访问的模式之一，接着会检查是否是需要进行身份验证的URL模式。如果是需要身份验证的URL模式之一(比如/studentHome.html和/teacherHome.html），则会检查请求中是否包含Authorization头或Authorization参数，如果包含了有效的Authorization信息，使用JwtUtils工具类解析JWT令牌。如果JWT解析成功且包含有效的用户名信息，则允许请求通过，否则拦截请求。

并返回未授权的响应。

该过滤器的作用是确保对于需要身份验证的资源，只有携带有效的身份验证信息的用户才能访问，以提高Web应用程序的安全性。

1. 登录与注册:

LoginServlet:

用户请求参数解析:使用ReadJSONUtils.parseJsonToClass(req, User.class)方法将HTTP请求中的JSON数据解析为User对象。

数据验证：使用Regex.validateName()和Regex.validatePassword()方法验证用户输入的用户名和密码是否符合规则。

业务逻辑处理：

调用PersonService接口的实现类PersonServiceImpl的方法，根据用户提供的用户名和角色类型查询数据库中的用户信息。

如果查询到了用户信息，并且用户输入的密码与数据库中存储的密码匹配，则执行以下操作：将用户对象的密码设置为数据库中存储的密码。

生成JWT令牌，将用户名和角色信息添加到JWT声明中。

创建JSON对象，将用户对象和JWT令牌添加到其中。

输出登录成功的消息和JSON响应。

如果未查询到用户信息或密码不匹配，则返回HTTP 401 Unauthorized响应，并向客户端发送错误消息。

异常处理：

如果用户名或密码验证失败，则向客户端发送"Regex Error"消息。

这个Servlet处理了登录请求，包括验证用户身份、生成JWT令牌以及返回相应的JSON数据或错误消息。

RegisterServlet:

请求参数解析：使用JSONObject.parseObject()方法将HTTP请求中的JSON数据解析为Student对象。

数据验证：使用Regex.validateName()等方法验证用户输入的用户名、密码和学号是否符合规则。

业务逻辑处理：

调用PersonServiceImpl类的实例方法，该类实现了PersonService接口，用于处理与用户相关的业务逻辑。

调用getStudentBtNumber()方法，根据学号查询数据库中是否已存在相同学号的学生信息。如果存在相同学号的学生信息，则向客户端发送"Error"消息。

如果不存在相同学号的学生信息，则调用addPerson()方法将新的学生信息添加到数据库中，并向客户端发送添加结果的消息。

异常处理：

如果用户名、密码或学号验证失败，则向客户端发送"Error"消息。

这个Servlet处理了注册请求，包括验证用户输入的信息、检查学号是否已存在以及将新的学生信息添加到数据库中

# 4.程序测试

# 5.项目亮点

# 6.心得体会