数据库第九次上机

2022/5/25

本次上机概览

本次的上机内容为：

- 常见web框架中的数据库应用
  - 项目搭建
  - 实体设计
  - 接口实现

- 本次上机为最后一次上机，持续时间为2周，难度与开放性都会相对更大一些，因此大家遇到问题请积极上网查询资料 或 在群里提问与同学老师助教一起讨论；

本次上机任务

经过本学期前几次上机的实验，我们已经学习数据库技术和原理等理论知识与实践，但在实际项目开发中，**数据库工具的应用往往还需要与web框架进行进一步的结合配置**，才能实现高效的业务使用；

以“前后端分离”的软件架构为例，常见的web后端框架有：

- Java: SpringBoot
- Python: Django / Flask / …
- JavaScript: Express.js / Node.js …
- PHP
- Ruby
- …

本次实验需要同学们基于任意一种web框架（推荐SpringBoot / Django / Node.js），完成本地工程项目的创建、并配置连接数据库，基于需求完成数据库实体设计、接口编写和测试等任务；

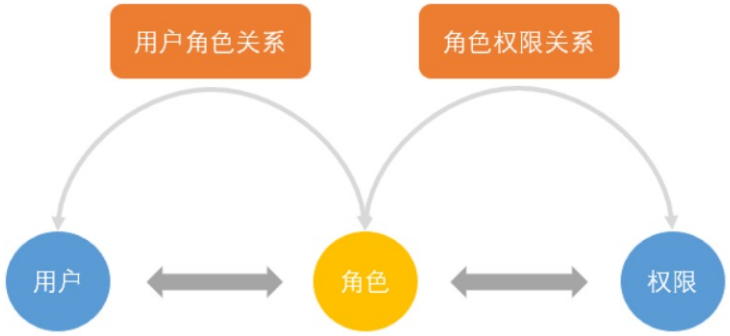
本次上机任务

模拟案例需求描述：（只使用数据库表来实现实体设计与管理，与数据库内核安全等完全无关）

现在要求实现一个简单的基于RBAC的权限管理系统，能够对用户添加相应的角色，以达到权限管理和控制的功能；每个用户可以同时有多个角色，每一种角色可以同时有多种权限；

Q1：数据库表结构的设计，完成所需要表格的创建，并自行模拟数据插入；

（此步骤可以在可视化工具中完成，**表设计可参考下一页ppt**，注意提供创建后的表截图，每个表需要有其各自必要的信息，例如id等，其中两个关系表都要求建立相应的外键约束）



本次上机任务

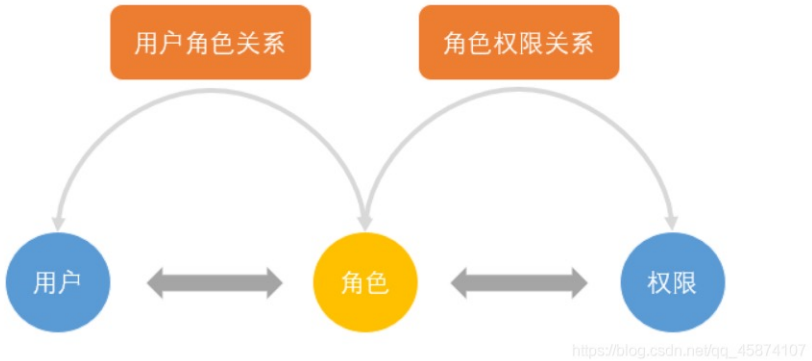
用户表（uid, uname, age, phone, city） // uid自增

角色表（rid, rname） // rid自增, rname为角色名称

权限表（pid, pname） // pid自增， pname为权限名称

用户角色表（id, uid, rid）// 外键约束

角色权限表（id, rid, pid）// 外键约束



自行模拟数据插入，可参考如下例子：

创建多个角色，例如团长、士兵长、战士、炊事员；

创建多个权限，例如组织会议、带队出征、战斗训练、吃饭、买菜、做饭；

其中团长可以组织会议、带队出征、吃饭

士兵长可以带队出征、战斗训练、吃饭

战士可以战斗训练、吃饭

炊事员可以买菜、做饭

本次上机任务

Q2：任选一种web后端框架，链接并控制数据库，在其暴露四个接口，以实现对用户表的增删改查

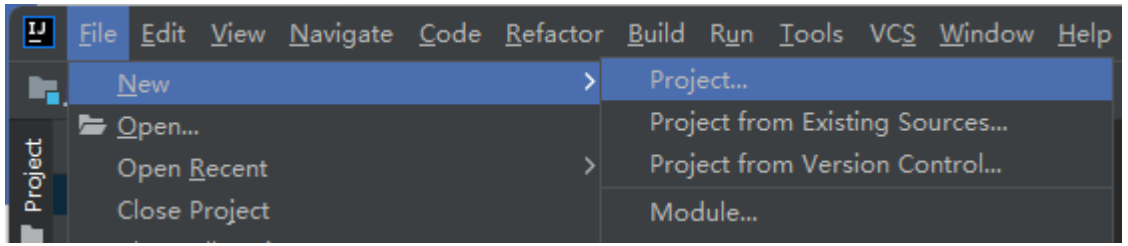
以“前后端分离”的软件架构为例，常见的web后端框架有：

- Java: SpringBoot
- Python: Django / Flask / …
- JavaScript: Express.js / Node.js …
- PHP
- Ruby
- …

下文给出SpringBoot框架和Node.js框架为例，大家可自行选择自己熟悉的语言的框架；

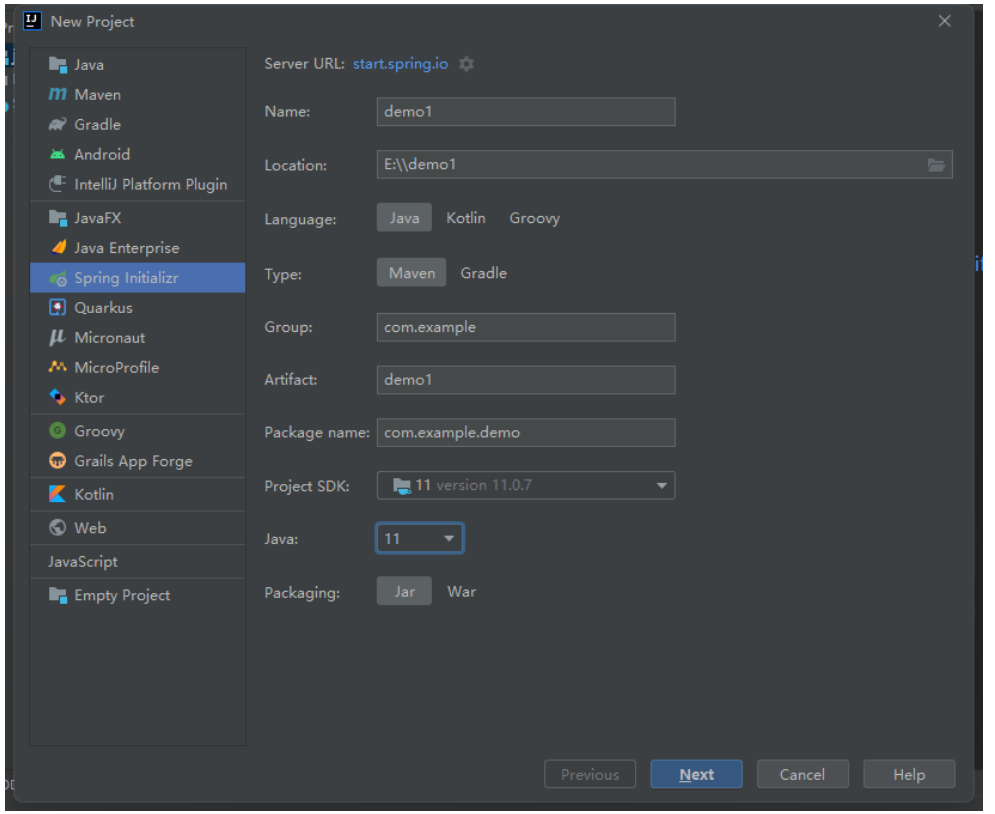
SpringBoot

准备环境和依赖，新建工程项目，以Java的SpringBoot框架为例，使用IDEA完成项目创建：



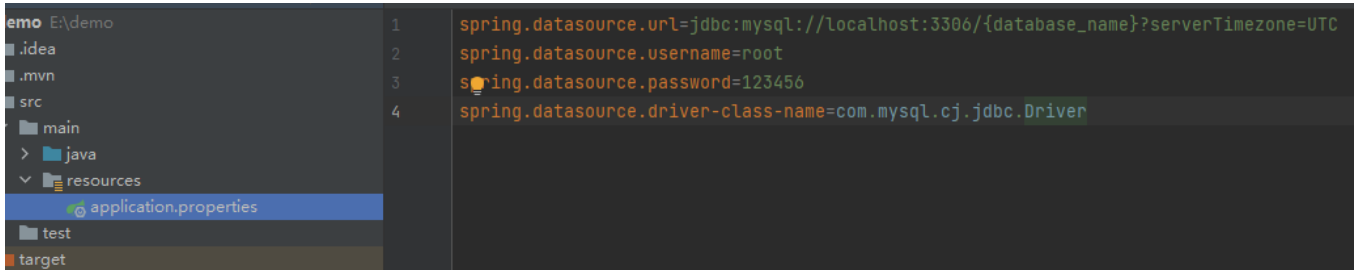
新建项目后向 pom.xml 中添加依赖

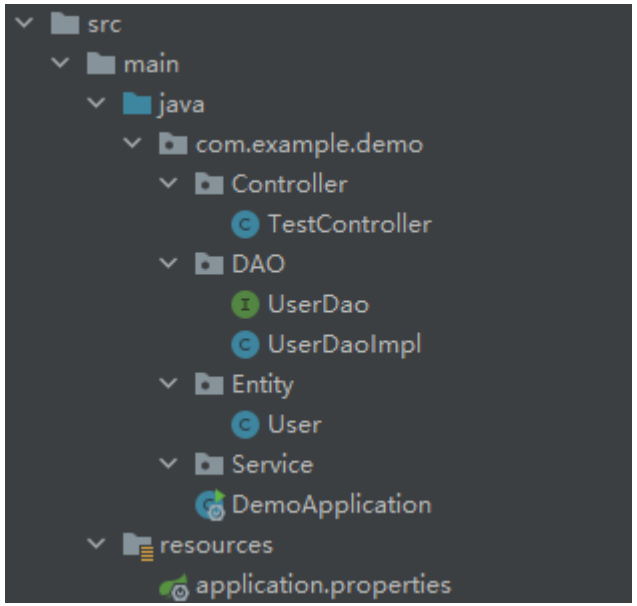
```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
</dependency>
```



SpringBoot

配置连接信息以及项目目录结构





参考文章：Springboot中的Service层、Controller层、Dao层、Entity层的功能总结

具体的一个web项目中是：Controller层调用了Service层的具体功能方法，Service层调用Dao层的方法，其中调用的参数是使用Entity层进行传递的。

我们需要创建实体层、DAO层以及Controller层，在Controller中暴露四个接口，以实现对User表的增删改查

注意：

1. 由于本次实验的项目内容比较单一，可以暂不涉及对service层的设计；
2. 具体各个类如何创建可以参考 zip文件；
3. 建议同学可以结合网上的教程资料，实际动手搭建，实验目的主要是帮助大家理解框架中的数据库的应用；
4. 更详细的Springboot+MySQL的项目搭建流程大家可以自行上网搜索，到处都是

SpringBoot

注意：
1. 由于本次实验的项目内容比较单一，可以暂不涉及对service层的设计；
2. 具体各个类如何创建可以参考 zip文件；
3. 建议同学可以结合网上的教程资料，实际动手搭建，实验目的主要是帮助大家理解框架中的数据库的应用；
4. 更详细的Springboot+MySQL的项目搭建流程大家可以自行上网搜索，到处都是

实体层 // User.java

```java
@Data
@AllArgsConstructor
@NoArgsConstructor
public class User {
    private int uid;
    private String uname;
    private String age;
    private String phone;
    private String city;
}
```

DAO层 // UserDao.java

```java
import ...

public interface UserDao {
    List<User> findall();
    User findByname(String name);
    boolean addUser(User user);
    boolean updateByName(String name,String age);
    boolean deleteByName(String name);
}
```

// UserDaoImp.java

```java
    //查询所有数据
    @Override
    public List<User> findall() {
        String sql = "select * from test_user";
        return jdbcTemplate.query(sql, new BeanPropertyRowMapper<User>(User.class));
    }
```

Springboot

注意：
1. 由于本次实验的项目内容比较单一，可以暂不涉及对service层的设计；
2. 具体各个类如何创建可以参考 zip文件；
3. 建议同学可以结合网上的教程资料，实际动手搭建，实验目的主要是帮助大家理解框架中的数据库的应用；
4. 更详细的Springboot+MySQL的项目搭建流程大家可以自行上网搜索，到处都是

Controller层

如下暴露出了/all 接口，启动后端后，浏览器访问 localhost:8080/all 则会请求数据库中User表的所有的记录

```java
@RestController
public class TestController {
    @Autowired
    private UserDaoImpl userDaoImpl;

    @RequestMapping(value = "/all",method = RequestMethod.GET)
    public String findAll() {
        List<User> list = userDaoImpl.findall();
        for (User user : list) {
            System.out.println(user.getName() + " " + user.getAge());
        }
        return "查询所有！";
    }
}
```

Node.js

将来找后端开发的同学选择SpringBoot是较好的，但是如今前端开发也不意味着不会接触数据库操作和服务端等，如今Node也已经成为了前端面试常考甚至明确要求的技能，如果将来想要走前端方向，学习并选择Nodejs完成本次实验是一个不错的选择。

简单的说 Node.js 就是**运行在服务端的 JavaScript**。

Node.js 是一个基于 Chrome JavaScript 运行时建立的一个平台。

Node.js 是一个事件驱动 I/O 服务端 JavaScript 环境，基于 Google 的 **V8 引擎**，V8 引擎执行 Javascript 的速度非常快，性能非常好。

本次实验主要涉及Nodejs的数据库连接部分操作。

Node.js环境配置

Nodejs安装请参考（官网下载然后一路确定即可）：

https://www.runoob.com/nodejs/nodejs-install-setup.html

首先，创建一个用于存储node.js应用程序的项目文件夹，并使用**npm init**命令创建package.json文件：

nodejs项目目录下cmd运行：**npm init**

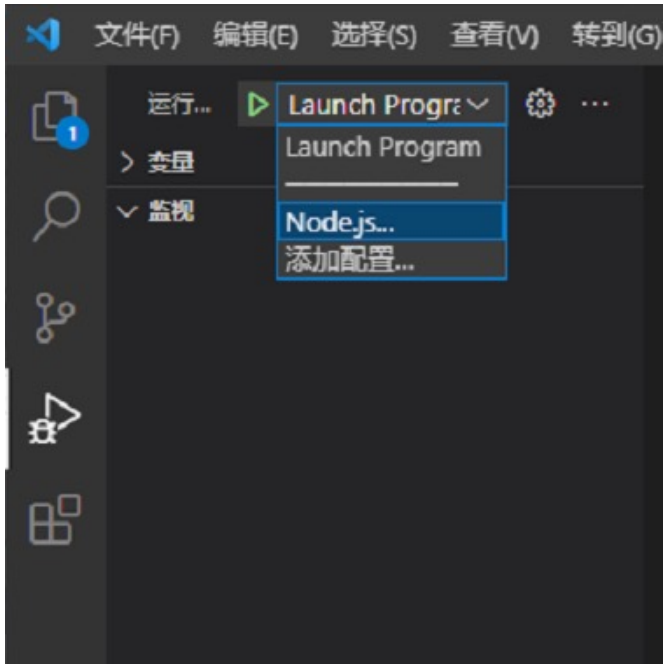
安装Node.js mysql驱动程序：

nodejs项目目录下cmd运行：**npm install mysql**

IDE选择：

https://www.php.cn/website-design-ask-484696.html

VSCode就可以方便进行Nodejs调试等



Node.js数据库连接

通过以下代码（示例为connect.js文件）完成本机mysql连接：

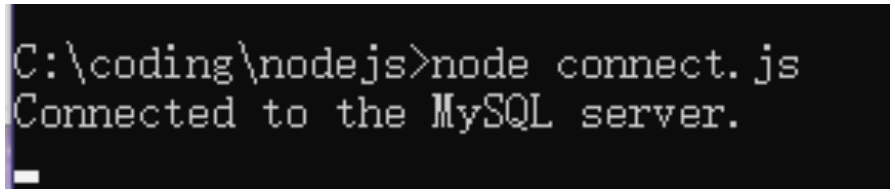
```
let mysql = require('mysql');

let connection = mysql.createConnection({
    host: 'localhost',
    user: 'root',
    password: '你的密码',
    database: '连接的数据库'
});

connection.connect(function(err) {
    if (err) {
      return console.error('error: ' + err.message);
    }

    console.log('Connected to the MySQL server.');
  });
```

可能出现的问题：
error: ER_NOT_SUPPORTED_AUTH_MODE: Client does not support authentication protocol requested by server; consider upgrading MySQL client

亲测解决方案：
https://www.jianshu.com/p/c8eb6d2471f8

代码详细：
https://www.yiibai.com/mysql/nodejs-connect.html

nodejs运行方式 js文件路径下cmd运行：
node 要运行的js文件名



Node.js增删改查

网上相关教程很多 搜索关键词 **Nodejs+Mysql**即可 下面这个是个能快速上手的教程：
https://www.yiibai.com/mysql/nodejs.html

增删改查本质还是执行SQL语句，比如创建表操作：

```
let createTodos = `create table if not exists todos(
                        id int primary key auto_increment,
                        title varchar(255)not null,
                        completed tinyint(1) not null default 0
                    )`;

connection.query(createTodos, function(err, results, fields) {
  if (err) {
    console.log(err.message);
  }
});
```

Node.js增删改查

插入操作（其他操作请自行查阅解决！）：

插入一行并返回插入的ID

以下 `insert-lastid.js` 程序将一个新行插入到 `todos` 表中，并返回插入的 `id` 。

```js
let mysql = require('mysql');
let config = require('./config.js');
let connection = mysql.createConnection(config);

let stmt = `INSERT INTO todos(title,completed)
            VALUES(?,?)`;
let todo = ['Insert a new row with placeholders', false];

// execute the insert statment
connection.query(stmt, todo, (err, results, fields) => {
  if (err) {
    return console.error(err.message);
  }
  // get inserted id
  console.log('Todo Id:' + results.insertId);
});

connection.end();
```

要将数据传递给SQL语句，请使用问号(`?`)作为占位符。

在这个例子中，我们分别使用两个问号(`?` , `?`)作为 `title` 和 `completed` 字段。

Node.js服务端构建

之前的操作都是如何通过js操作mysql数据库，但要想完成本次实验至少需要**写一个持续运行的服务端程序**，实现url请求的响应。
nodejs实现一个简单的回应请求的服务端非常容易，只需导入http模块并完成url的解析和处理即可，可参考：
https://www.jianshu.com/p/81a5032eacc2

但这样的项目可用性太差，如果想完善服务端项目，可以考虑实现较为完整的路由跳转：
https://www.runoob.com/nodejs/nodejs-router.html

或者使用基于 Node.js 平台的 web 开发框架 Express实现更商业化的开发：
http://expressjs.com/

NodeJS+Express+mySQL开发教程一则：
https://www.pianshen.com/article/6372175301/

请视个人能力完成服务端开发即可

关于作业提交

Q1：完成基于RBAC项目背景的数据库设计，要求给出5张表格的SQL建表语句；
Q2：任选一种框架，实现建立数据库连接，并实现4个接口，完成对 User 实体的增删改查；
（给出浏览器访问接口截图、返回信息与数据库状态变化截图）
选做 Q3：在Q2的基础上，补充其他的实体类与接口，以完成RBAC的权限控制（例如新建角色、用户添加角色、新建权限、为角色添加权限、用户权限鉴别判断等接口）
选做 Q4：选择其他的web框架，完成Q2的接口实验；

请在**PDF/WORD等任何方便助教阅读查看的文档**中按照各个作业要求提交相关内容，**记得标清题号**。

若为PDF/WORD单文档文件直接提交即可，其他提交压缩包，命名为 “学号_姓名_第*次实验” 。

提交网址：软件学院云平台**第九次上机** https://cloud.beihangsoft.cn/#/security/login
（按要求提交）

作业截止时间为**周日24:00之前，提交方式为提交到云平台。**

相关参考

一般SQL语法：
http://www.w3school.com.cn/sql/index.asp

官方文档：
SQL Server: https://docs.microsoft.com/zh-cn/sql/t-sql/language-elements/transactions-transact-sql?view=sql-server-2017
MySQL: https://dev.mysql.com/doc/refman/8.0/en/sql-syntax-transactions.html

多用搜索引擎：
https://cn.bing.com/
https://www.google.com/
https://www.baidu.com/

以及数据库课程PPT