* 快速搭建Spring Boot + Apace Shiro环境
* 常用Case实现

Apache Shiro和Spring Security不同，它没有自带的登录页面和基于内存的权限验证。所以我们使用jsp去编写简单的登录页面，使用Mybatis连接mysql数据库进行用户及其权限和角色的信息的存储。

（1）首先在IDEA中，创建一个Spring Boot工程

（2）选择需要的模块

（3）项目创建完成后，补充相应的依赖，pom.xml文件中配置的完整依赖如下：

（4）我们需要在数据库中进行用户及其权限和角色信息的存取，并且我们将按照RBAC模型完成文中Case的开发，所以首先需要创建数据库表格及向表格插入一些数据。

-- 权限表 --  
CREATE TABLE permission (  
 pid int(11) NOT NULL AUTO\_INCREMENT,  
 name VARCHAR(255) NOT NULL DEFAULT '',  
 url VARCHAR(255) DEFAULT '',  
 PRIMARY KEY (pid)  
) ENGINE = InnoDB DEFAULT CHARSET = utf8;  
  
INSERT INTO permission VALUES ('1', 'add', '');  
INSERT INTO permission VALUES ('2', 'delete', '');  
INSERT INTO permission VALUES ('3', 'edit', '');  
INSERT INTO permission VALUES ('4', 'query', '');  
  
-- 用户表 --  
CREATE TABLE user(  
 uid int(11) NOT NULL AUTO\_INCREMENT,  
 username VARCHAR(255) NOT NULL DEFAULT '',  
 password VARCHAR(255) NOT NULL DEFAULT '',  
 PRIMARY KEY (uid)  
) ENGINE = InnoDB DEFAULT CHARSET = utf8;  
  
INSERT INTO user VALUES ('1', 'admin', '123');  
INSERT INTO user VALUES ('2', 'demo', '123');  
  
-- 角色表 --  
CREATE TABLE role(  
 rid int(11) NOT NULL AUTO\_INCREMENT,  
 rname VARCHAR(255) NOT NULL DEFAULT '',  
 PRIMARY KEY (rid)  
) ENGINE = InnoDB DEFAULT CHARSET = utf8;  
  
INSERT INTO role VALUES ('1', 'admin');  
INSERT INTO role VALUES ('2', 'customer');  
  
-- 权限角色关系表 --  
CREATE TABLE permission\_role (  
 rid int(11) NOT NULL ,  
 pid int(11) NOT NULL ,  
 KEY idx\_rid (rid),  
 KEY idx\_pid (pid)  
) ENGINE = InnoDB DEFAULT CHARSET = utf8;  
  
INSERT INTO permission\_role VALUES ('1', '1');  
INSERT INTO permission\_role VALUES ('1', '2');  
INSERT INTO permission\_role VALUES ('1', '3');  
INSERT INTO permission\_role VALUES ('1', '4');  
INSERT INTO permission\_role VALUES ('2', '1');  
INSERT INTO permission\_role VALUES ('2', '4');  
  
-- 用户角色关系表 --  
CREATE TABLE user\_role (  
 uid int(11) NOT NULL ,  
 rid int(11) NOT NULL ,  
 KEY idx\_uid (uid),  
 KEY idx\_rid (rid)  
) ENGINE = InnoDB DEFAULT CHARSET = utf8;  
  
INSERT INTO user\_role VALUES (1, 1);  
INSERT INTO user\_role VALUES (2, 2);

1. 创建与表格所对应的pojo类
2. 然后创建dao层的mapper接口：

public interface UserMapper {  
  
 /\*\*  
 \* description:   
 \* @date 2018/11/27 9:01  
 \* @param username :   
 \* @return : com.mmall.demo2.model.User  
 \*\*/  
 User findByUsername(@Param("username") String username);  
}

1. 以及编写与之对应的xml文件

*<?*xml version="1.0" encoding="UTF-8" *?>*<!DOCTYPE mapper  
 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
 "http://mybatis.org/dtd/mybatis-3-mapper.dtd"*>*<mapper namespace="com.mmall.demo2.mapper.UserMapper">  
  
 <resultMap id="userMap" type="com.mmall.demo2.model.User">  
 <id property="uid" column="uid" />  
 <result property="username" column="username" />  
 <result property="password" column="password" />  
 <collection property="roles" ofType="com.mmall.demo2.model.Role">  
 <id property="rid" column="rid" />  
 <result property="rname" column="rname" />  
 <collection property="permission" ofType="com.mmall.demo2.model.Permission">  
 <id property="pid" column="pid" />  
 <result property="name" column="name" />  
 <result property="url" column="url" />  
 </collection>  
 </collection>  
 </resultMap>  
  
 <select id="findByUsername" parameterType="string" resultMap="userMap">  
  
 </select>  
</mapper>

1. 接着是service层接口
2. 编写实现类来实现UserService接口