JPA（Java Persistence API）定义了一系列对象持久化的标准，目前实现这一规范的产品有Hibernate、TopLink等。

【1】配置依赖

在pom文件中，配置如下依赖：

spring-boot-starter-web

spring-boot-starter-test

mysql-connector-java

spring-boot-starter-thymeleaf

spring-boot-starter-data-jpa

【2】设置相关配置

在application.yml文件中，设置如下配置：

server:

port: 8080 （设置端口号）

spring:

datasource: （配置数据库连接）

driver-class-name: com.mysql.jdbc.Driver

url: jdbc:mysql://localhost:3306/mydata?characterEncoding=utf8

username: root

password: ty914321

jpa: （配置spring-data）

hibernate:

ddl-auto: none

show-sql: true

mvc: （配置静态页面的扫描路径）

view:

prefix:

classpath: /templates/

suffix: .html

static-path-pattern: /static/\*\*

【3】配置Jpa

新建一个JpaConfiguration类，代码如下：

/\*\*

\* JPA配置类

\*/

@Order(Ordered.HIGHEST\_PRECEDENCE)

@Configuration

@EnableTransactionManagement(proxyTargetClass = true)

@EnableJpaRepositories(basePackages = "com.tianyu.dao") /\*设置dao的路径\*/

@EntityScan(basePackages = "com.tianyu.pojo") /\*设置pojo的路径\*/

public class JpaConfiguration {

@Bean

PersistenceExceptionTranslationPostProcessor persistenceExceptionTranslationPostProcessor(){

return new PersistenceExceptionTranslationPostProcessor();

}

}

【4】配置项目启动入口

新建一个Entry类，代码如下：

/\*\*

\* 项目启动入口，配置包根路径

\*/

@SpringBootApplication

@ComponentScan("com.tianyu")

public class Entry {

public static void main(String args[]) throws Exception{

SpringApplication.run(Entry.class);

}

}

【5】dao层

//spring-data方式

public interface UserRepository extends JpaRepository<User,Long>{

//登陆

@Query("select t from User t where t.userName = :userName and t.userPwd = :userPwd")

User userLogin(@Param("userName") String userName,@Param("userPwd") String userPwd);

}

【6】service层

新建UserService类，代码如下：

@Service

public class UserService {

@Autowired

private UserRepository userRepository;

//登陆

public Boolean userLogin(String userName,String userPwd){

User user = null;

try {

user = userRepository.userLogin(userName,userPwd);

}catch (Exception e){

e.printStackTrace();

}

if(user==null){

return false;

}else{

return true;

}

}

}

【7】controller层

新建UserController类，代码如下：

@Controller

@RequestMapping("/user")

public class UserController {

@Autowired

private UserService userService;

@RequestMapping("/userLogin")

public String userLogin(@RequestParam(value = "userName") String userName, @RequestParam(value = "userPwd") String userPwd, HttpSession session, Model model){

Boolean flag = userService.userLogin(userName,userPwd);

if(flag){

session.setAttribute("userName",userName);

return "Hello";

}else {

model.addAttribute("msg","用户名不存在或者密码错误！");

return "Login";

}

}

}

【8】静态页面

在templates文件夹下（即在application.yml文件中配置的静态页面扫描路径），新建页面。

其中，html页面从后台取值可以使用thymeleaf标签，例如：

<span th:text="${msg}"></span>

其中，后台传值方式可以使用ModelMap，Model，Session或者json

ModelMap和Model使用th:text="${变量名}"方式

Session使用th:text="${session.变量名}"方式

json使用