# 1.前后台数据交互 ModelAndView

## 1.controller里面代码

@RequestMapping(value = **"/mng/login"** ,method = RequestMethod.***GET***)  
**public** ModelAndView findall(){  
 **testSqlite** = **new** TestSQLite();  
 ModelAndView mv = **new** ModelAndView(**"find"**); html文件名称  
 List<Mnxra> mnxras = **testSqlite**.queryAll();   
 mv.addObject(**"mnxras"**,mnxras);  
 **return** mv;  
}

## 2.html接收

## 2.1头部必须写明（因为使用了thymeleaf依赖才具有的功能）

<**html class="no-js" lang="zh-CN" xmlns:th="http://www.thymeleaf.org"**>

## 2.2html代码

<**tr th:each="mnxra:${mnxras}"**>  
 <**td th:text="${mnxra.id}" id="id"**> </**td**>  
 <**td th:text="${mnxra.name}"**></**td**>  
 <**td th:text="${mnxra.subdomain}"**></**td**>  
 <**td th:text="${mnxra.href}"**></**td**>  
 <**td th:text="${mnxra.createtime}"**></**td**>  
 <**td**><**input type="button" onclick="***update*()**" value="更新"**></**input**></**td**>  
 <**td**><**input type="button" onclick="**delete()**" value="删除"**></**input**></**td**>  
  
</**tr**>

# 2.简单string传入前台

## 1.html

<**input th:value="${href}"**> </**input**>

## 2.controller

ModelAndView mv = **new** ModelAndView(**"index"**);  
mv.addObject(**"href"**,href+**"yyyy"**);

# 3.redirect 实现跳转

**public** String home() {

**return "redirect://www.minixiao.com"**;

}

# 4.实现自定义接口方法查询数据库

**public interface** SubdomainRepository **extends** CrudRepository<Subdomain,Integer> {  
 *//根据子域名查找跳转链接* @Query(**"select href from Subdomain where sub=?1"**)  
 String findBySub(String sub);  
 *//根据关键字查找链接信息* @Query(**"select s from Subdomain s where s.name like %?1%"**)  
 List<Subdomain> findByKeywords(String keyword);  
 *//查找全部信息* List<Subdomain> findAll();  
 *//根据id查找* Subdomain findById(**int** id);  
}

# 5.打印日志进行调试

**private** Logger logger = LoggerFactory.*getLogger*(HomeController.**class**);

**logger**.info(**"index"**);

# 6.实体类连接数据库

## 1.配置数据源

**spring.jpa.show-sql** = **true   
spring.jpa.hibernate.ddl-auto** = **update  
spring.datasource.url**=**jdbc:h2:file:~/.h2/s\_routine***#spring.datasource.url = jdbc:h2:file:~/.h2/testdb***spring.datasource.username** = **sa  
spring.datasource.password** = **sa  
spring.datasource.driverClassName** =**org.h2.Driver  
  
spring.h2.console.enabled**=**true**

## 2.编写实体类

@Entity  
@Table(name=**"subdomains"**)  
**public class** Subdomain {  
 @Id  
 @GeneratedValue(strategy= GenerationType.IDENTITY)  
 **private** Integer **id**; *//id* @Column(length = 50,nullable = **true**)  
 **private** String **name**;  
 @Column(length = 50,nullable = **true**)  
 **private** String **sub**;  
 @Column(length = 255,nullable = **true**)  
 **private** String **href**;  
 @Column(nullable = **true**)  
 **private** Date **createOn**;*//创建时间* **public** Subdomain(){}

# 7. @SessionAttributes

@Controller

@RequestMapping("/user")

//将ModelMap中名为user的属性放到Session中，以便这个属性可以跨请求访问

@SessionAttributes("user")

**public** **class** UserController {

//处理登录请求

@RequestMapping(value="/login")

**public** String login(Users user,ModelMap model){

//将user对象以user为键放入到model中

model.put("user", user);

**return** "success";

}