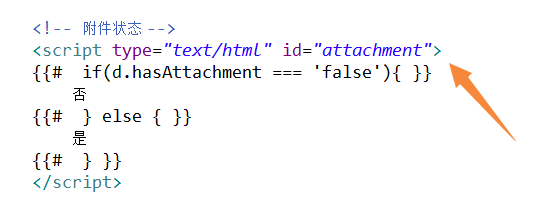
Layui Table使用特定的方法返回数据

Table的js实现



Mybatis PageHelper插件的分页统计



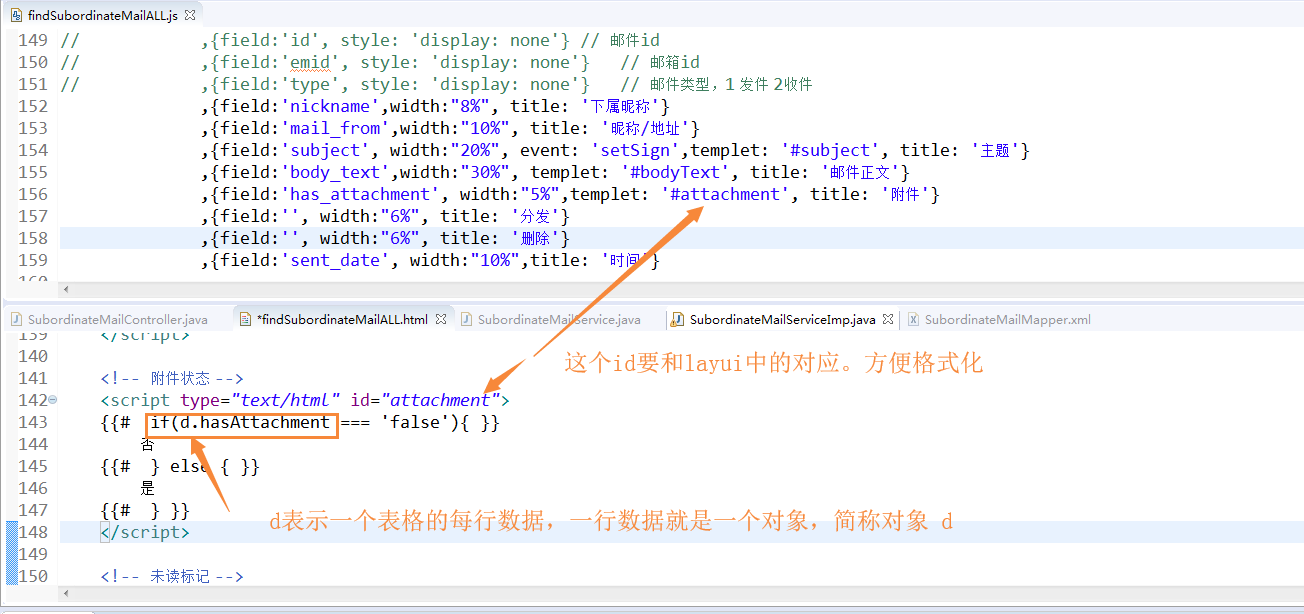
Mybatis使用IN关键字时，用${}来获取值，不需要用#{}来进行占位（预编译）

Thymeleaf在前台取出后台存到request | session | application中的值

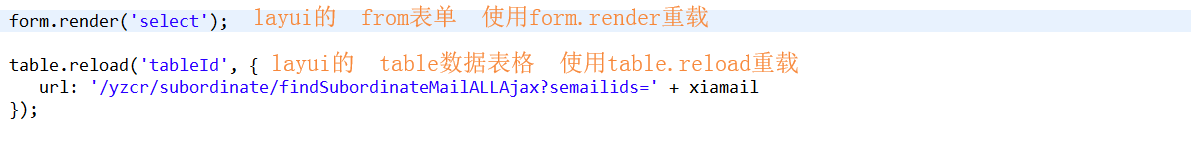
以[[${variable}]] | [[${session.variable}]]的形式取后台传过来的参数

[Layui laytpl模板引擎](https://www.layui.com/doc/modules/laytpl.html)

其中的d表示数据(data)



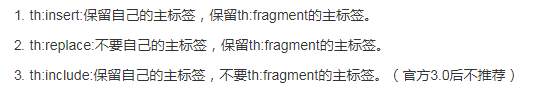
Layui 表单渲染



Thymeleaf中的

**th:selected、th:if、th:value、th:onclick标签实践，th:onclick：作用同html的onclick**。

t**h:include标签与html标签中的include意义相同（包含另一个页面），不过要与th:fragment标签结合使用**





百度的富文本编辑器Ueditor

**ueditor的serverUrl**



例如图片的上传，那个接口就可以写后台的图片上传接口。

jQuery

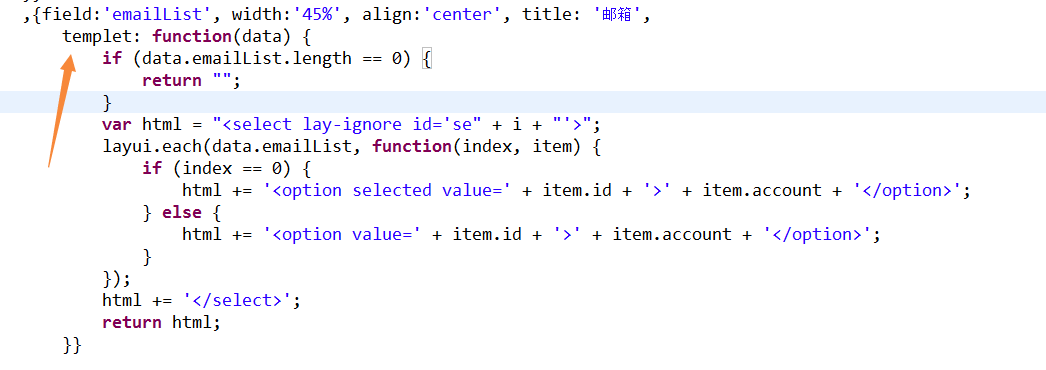
**$("input[type='text'][name='subject']").val(tname);**

**$("input[type='text'],[name='subject']").val(tname);**

**筛选input标签，前者是与条件，后者或条件**

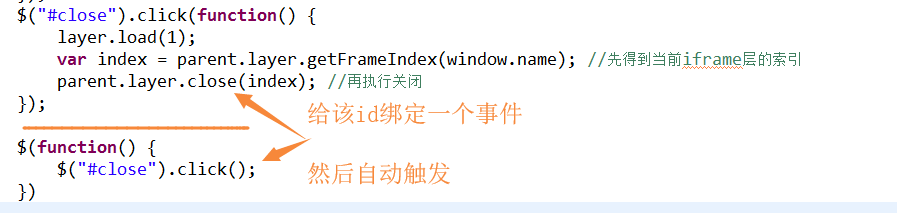
**Layui自定义模板**

**数据表格中使用id然后在html中进行格式话，或者直接操作，后边要return出去**



jQuery

使用jquery自定义事件然后自动触发



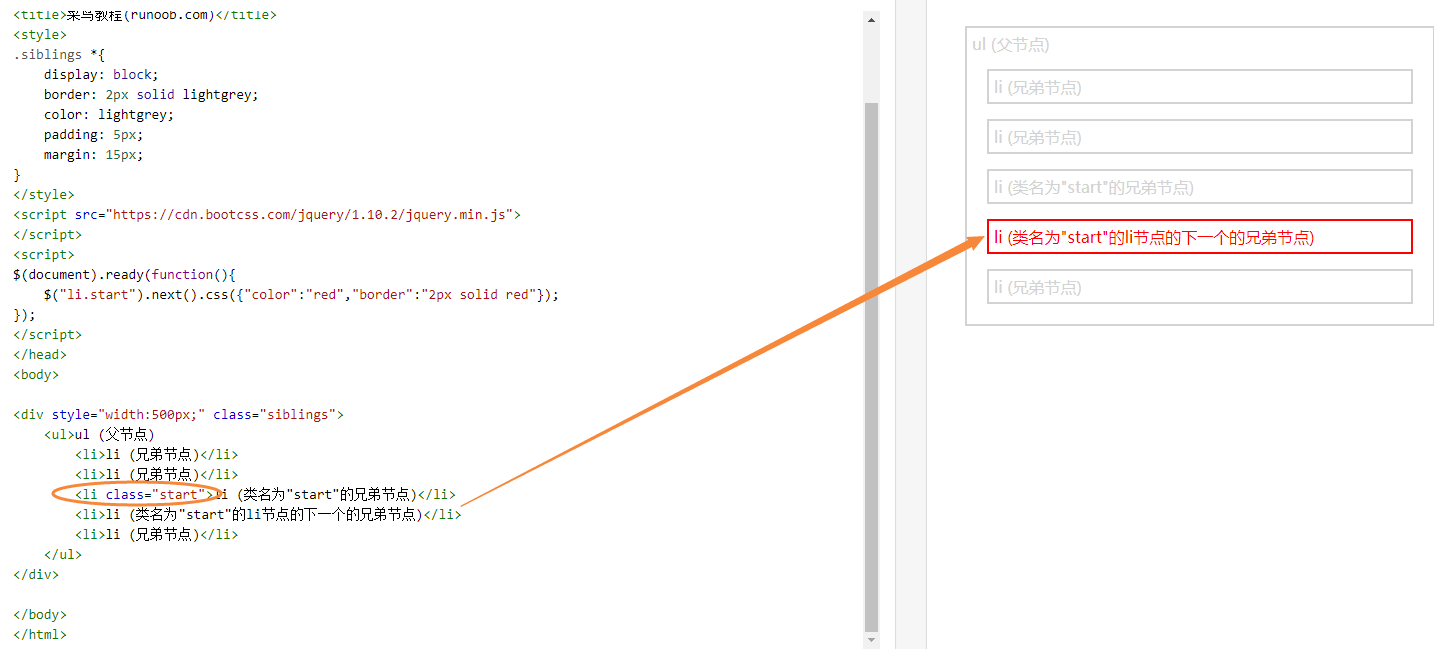
[$().each](http://jquery.cuishifeng.cn/each.html) &&  [$.each](http://jquery.cuishifeng.cn/jQuery.each.html)

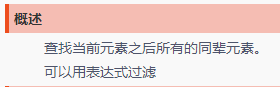
前者更多是用于处理DOM对象，如遍历某div下所有的span；

后者更多是用于处理数据，如遍历数组、对象。

jQuery

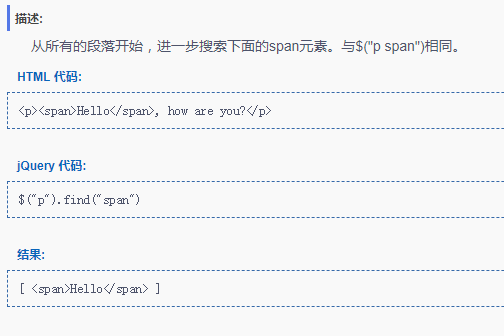
**Jquery的next()，表示该元素的紧邻的下一个同辈元素**



nextAll() 



find() 用于查找元素



Layui表格重载，table.reload(filterId);

Spring注解

@Value("${}")，用于获取properties属性文件的属性

@Value("#{}")，用于获取类的属性。初步认识

@Resource，默认根据byName进行注入，找不到再根据类型注入

@Autowire，根据类型byType注入

**Jquery的removeClass()方法，移除类名，**

**Layui的layue.closeAll();关闭所有弹窗(layer.open())**

**mybatis的Example类及一些常用方法。**

**使用这个类，可以搭建无限的where条件。**

**一般可以使用，and() 多个条件AND连接、or()多个条件OR连接**

**·排序example.setOrderByClause("sent\_date desc")、**

**已有的方法**

**·andEqualTo("userId", userId)**

**·andIn("state", list)**

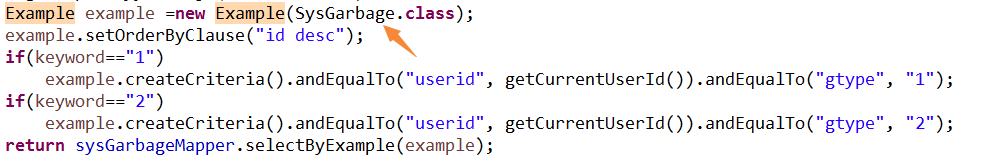
**·andLike("bodyText", word)**

**或者自定义条件**

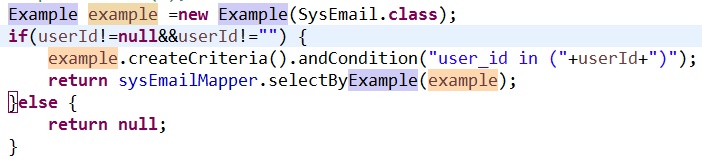
**·example.createCriteria().andCondition("user\_id in ("+userId+")");**

**·example.and().andCondition("user\_id in ("+userId+")");**

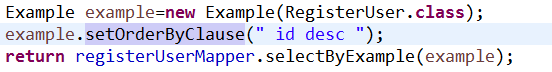
可以



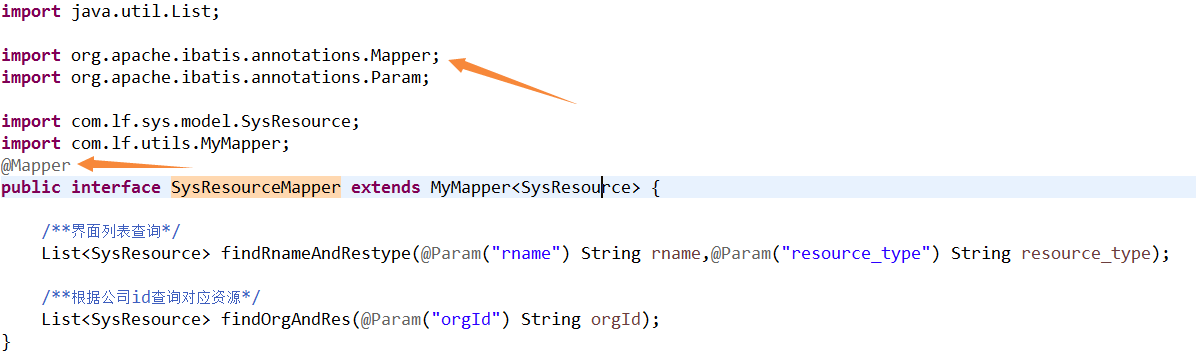
或者



再或者



Mapper文件



**jQuery移除事件**

[**$('#btn').unbind("click");**](http://jquery.cuishifeng.cn/unbind.html) **//移除click**

[**$('#btn').unbind();**](http://jquery.cuishifeng.cn/unbind.html) **//移除所有,3.0版本中该方法废弃，推荐使用[off()](http://jquery.cuishifeng.cn/off.html)**

**unbind(type [,data]) //data是要移除的函数**

**('#btn').one("click",function(){.......}); // 只触发一次的事件**

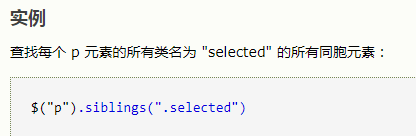
Thymeleaf的一个内置对象

strings 如使用到它来截取分隔的字符，注，使用内置对象要${#}这样的形式

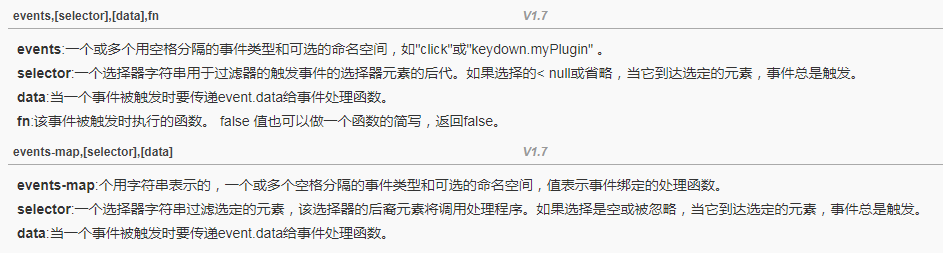
${#strings.arraySplit}，其中一个。分隔



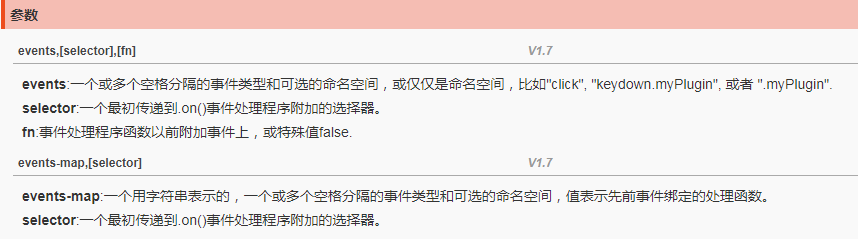
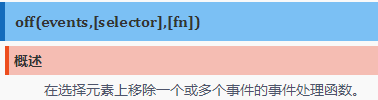
**Jquery的siblings用于选择所有的同胞元素**



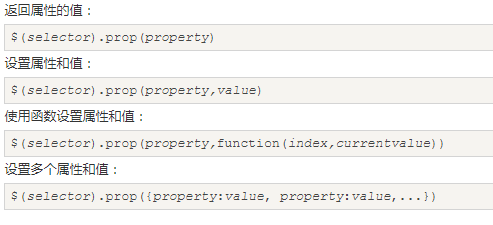
Jquery的on()，绑定事件推荐使用这个，不推荐bind()



Jquery的off()，用于卸载事件，不推荐unbind()



**Jquery prop()方法用于设置或返回被选元素的属性和值**



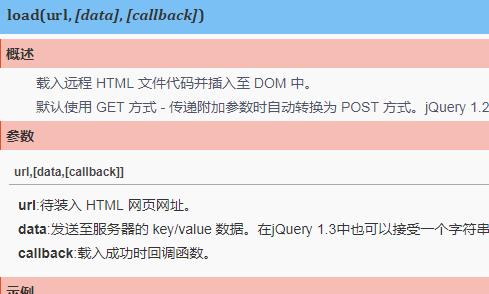
prop()设置的属性应该由removeProp()来进行移除

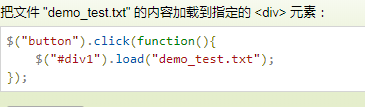


例如



[Jquery 的load方法](http://jquery.cuishifeng.cn/load.html)





SpringMVC框架中，controller方法返回类型为void，那么就会找和url**名字相同的页面**

<http://www.iteye.com/problems/109836>

<https://blog.csdn.net/weixin_38617682/article/details/78754602>

默认调用dispatcher方法吗？

jquery序列化追加数据，可以这样

$(“from-id”).serialize() + “&parame=” + value;

[**@Transient**](https://www.cnblogs.com/zhangzhiqin/p/7998001.html)**注解**

**表示该属性不是表中的字段，但是联表的时候可以用上。在类属性上进行注解，然后在mapper中也配置下，联表查询时可以接受该字段了。**

Layui弹窗（open()）

**layui.open()时，弹出的content可以时id也可以时class**

**content:$(“#id”) || content:$(“.class”);**

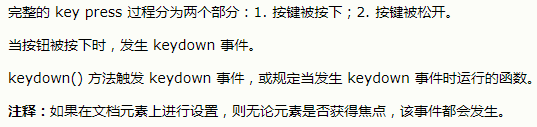
[bootstrap table更改样式通过data-formatter来进行](https://bootstrap-table.com/docs/api/column-options/#formatter)

**可以修改样式，如data-formatter=”style()”，js中function style()**

**jquery键盘事件keypress、keydown和keyup**

**指键盘按下-放开这个过程、键盘按下、键盘放开**

[**keypress:**](http://jquery.cuishifeng.cn/keypress.html)



[**keydown:**](http://jquery.cuishifeng.cn/keydown.html)



[**keyup:**](http://jquery.cuishifeng.cn/keyup.html)



**jQuery的each跳出循环，**

**$.each (list, function(index, item) {**

**console.log(“跳出循环(等同于break):return false;”);**

**console.log(“跳出当前循环(等同于continue): return true;”);**

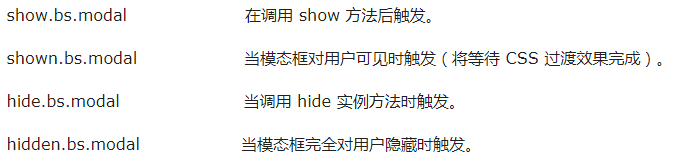
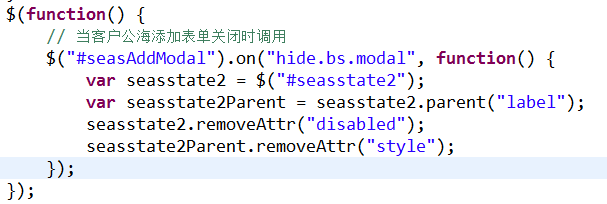
**});**

**jQuery ajax的timeout设置**

**可限制ajax的执行时间，到了超时时间自动执行error回调**

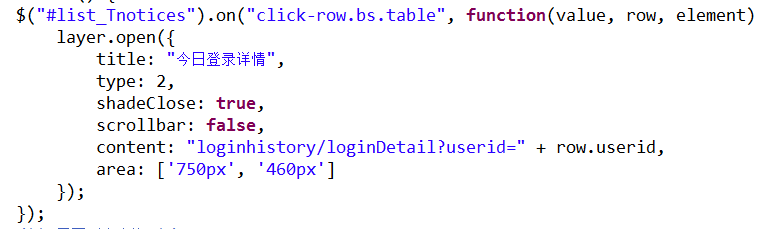
**Bootstrap模态窗口关闭，可监听**

**模态窗口关闭时调用**



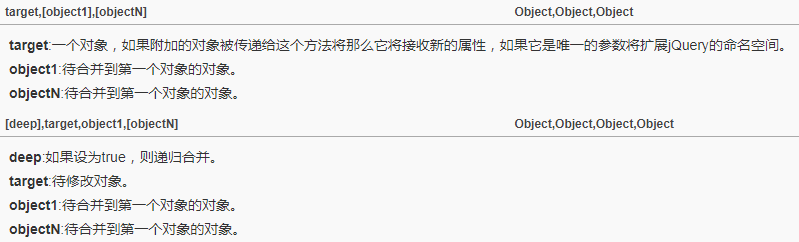
[**https://blog.csdn.net/wn464319127/article/details/50432965**](https://blog.csdn.net/wn464319127/article/details/50432965)

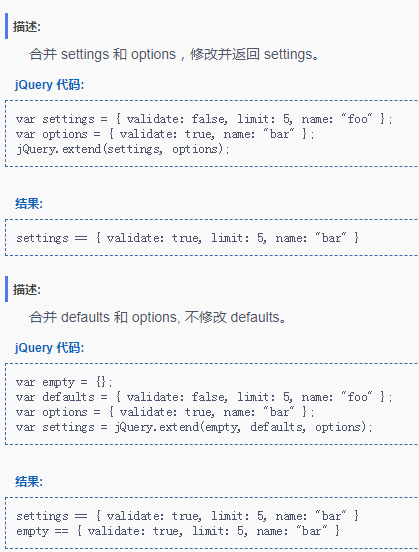
**Bootstrap表格的行监听**



[jQuery.extend](http://jquery.cuishifeng.cn/jQuery.extend.html)

用于合并一个对象





Layui数据表格的update

更新指定行，重新渲染该行（可更新数据）。

不会重新请求，但是会刷新



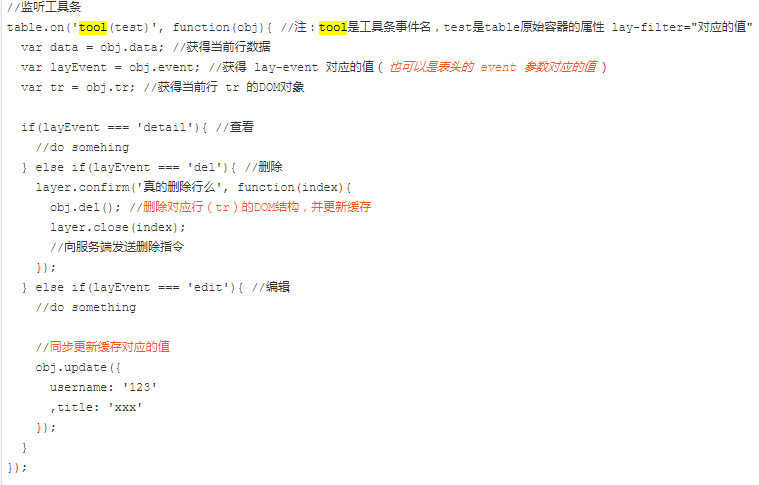
百度富文本编辑器Ueditor

ue.setContent(content, true)，true表示在原有的基础上追加内容

layui数据表格的event

监听工具条(table.on("tool(tableId)", function() {}))





jQuery中empty和remove的区别之一是

移除元素时包不包括自己

[jQuery中attr和prop的区别](https://www.cnblogs.com/Showshare/p/different-between-attr-and-prop.html)

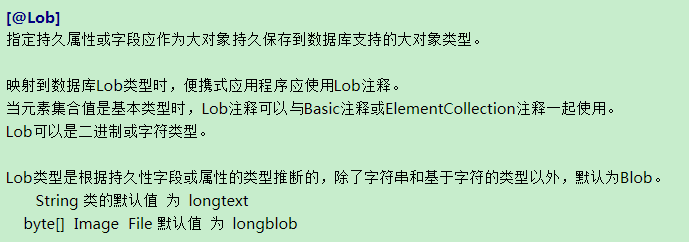
固有属性：标签元素本身的属性，

自定义属性：自己定义的属性，如age=”21”，age就是自定义属性

* 对于HTML元素本身就带有的固有属性，在处理时，使用prop方法。
* 对于HTML元素我们自己自定义的DOM属性，在处理时，使用attr方法。



@Lob注解





jQuery中判断是否含有某个类

1. is(".classname")

2. hasClass("classname")

使用layui时要注意哪些问题，**比如**使用form表单那块，那么就就记得渲染.

layui.use(“form”, function() {

var form = layui.form;

Form.render();

});

[@Param](https://www.cnblogs.com/thomas12112406/p/6217211.html)



富文本编辑器会对输入的内容进行格式处理。这是一个**细节**，假如你现在在做着长字符串替换的事情。

在富文本编辑器(Ueditor)会对输入的内容进行格式处理，需要进行文字匹配时要以编辑器处理过的格式进行匹配。

jQuery 的on用于动态绑定事件

$("#primary").on("click", "p", function() {})

那么这里给p绑定了点击事件

<div id=”primary”><p></p></div>

JQuery操作select（取值、回显数据）

<select id=”a” onchange=”change()”><option value=”1”>1</option></select>

获取选中的option值

$(“#a”).val();

<https://www.cnblogs.com/zhangym118/p/5509673.html>

点击option时触发，绑定onchange事件。

function change() {}

或者 $(“#a option:selected”).val();

bootstrap表格重载

$("#id").bootstrapTable('refresh',{url:'url'});

# [jQuery trigger()](http://www.runoob.com/jquery/event-trigger.html)

点击当前元素可触发另一元素某事件



Mapper中使用模糊查询

LIKE关键字配合使用

1、直接在传入xml前进行拼接(“\_”或者”%”)

LIKE #{field}

2、xml中使用CONCAT进行拼接

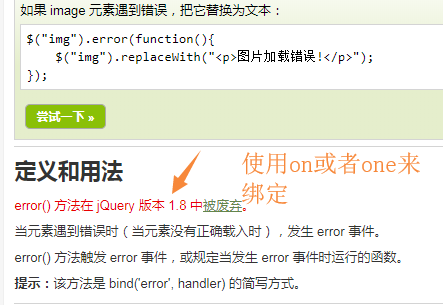
LIKE CONCAT("%", #{field}, "%")

3、使用单引号进行拼接

LIKE '%field%'

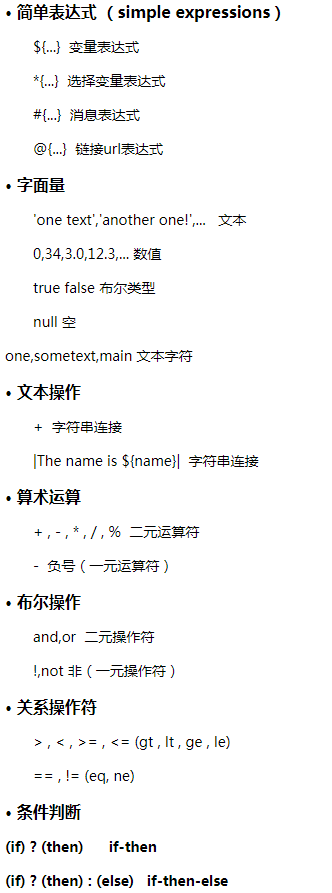
[onerror事件](http://www.runoob.com/jquery/event-error.html)

[jquery的error事件。](http://www.runoob.com/jquery/event-error.html)



Thymeleaf模板引擎

th:selected="${used.defaultmailbox != null} and ${used.defaultmailbox} eq ${semail.id}"

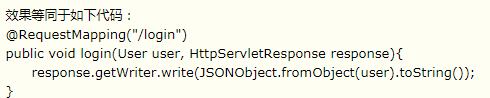


基本语法

<https://www.cnblogs.com/nuoyiamy/p/5591559.html>

[@ResponseBody](http://www.cnblogs.com/qiankun-site/p/5774325.html)

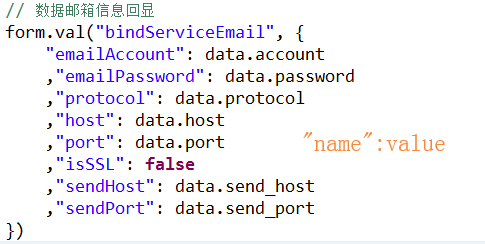
@ResponseBody注解的作用是将controller的方法返回的对象通过适当的转换器转换为指定的格式之后，写入到response对象的body区，通常用来返回JSON数据或者是XML数据，需要注意的呢，在使用此注解之后不会再走试图处理器，而是直接将数据写入到输入流中，他的效果等同于通过response对象输出指定格式的数据。



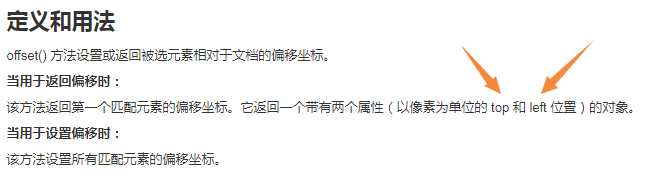
[layui表单重置，js重置表单, jquery重置表单](http://www.cnblogs.com/w-serendipity/p/9850843.html)



Layui form数据回显



[jquery offset()方法](http://www.runoob.com/jquery/css-offset.html%20%20)



ibatis返回整形时，若无数据则返回0

jQuery遍历对象

[·对元素进行遍历，](http://jquery.cuishifeng.cn/each.html)

$("元素").each(calback);

如：$("div").each(function(index) {

})

[·对数组进行遍历](http://jquery.cuishifeng.cn/jQuery.each.html)

$(“数组”).each(calback);

如：$(data, function(index, item) {

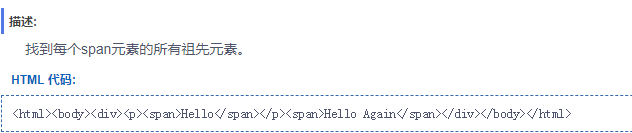
})

[thymeleaf的each用法](https://blog.csdn.net/u013845177/article/details/78568790)



使用JQuery查看祖先元素

parents()

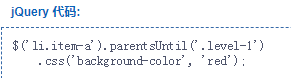
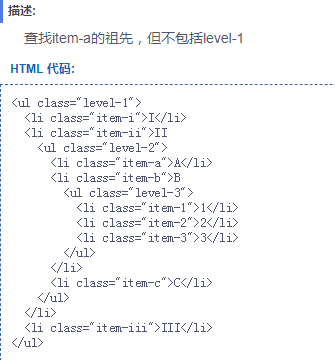


<http://jquery.cuishifeng.cn/parents.html>

parentsUntil

查找当前元素的所有的父辈元素，直到遇到匹配的那个元素为止。

如果提供的jQuery代表了一组DOM元素，.parentsUntil()方法也能让我们找遍所有元素的祖先元素，**直到**遇到了一个跟提供的参数匹配的元素的时候**才会停下来**。这个返回的jQuery对象里**包含了下面所有找到的父辈元素**，但**不包括那个选择器匹配到的元素。**



<http://jquery.cuishifeng.cn/parentsUntil.html>

thymeleaf的th:text="|字符拼接的方式输出${user.username}|"

或者th:text="'字符拼接方式输出'+${user.username}"

th:unless，和th:if是相反的。如果为false则执行



<https://blog.csdn.net/hry2015/article/details/73253080>

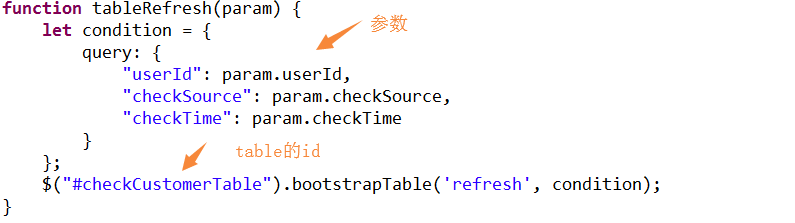
Swagger

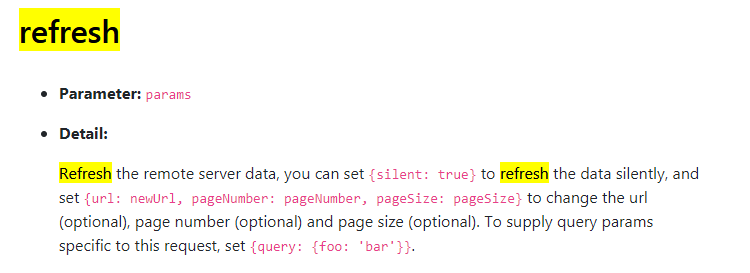
是一个开源项目，一个API文档生成工具，可用于生成API文档、维护和调试接口。



<https://blog.csdn.net/i6448038/article/details/77622977>

bootstrap表格带参数刷新





<https://blog.csdn.net/lanyang123456/article/details/55805478>

MyBatis的”if-else”写法



<https://www.cnblogs.com/a8457013/p/8033263.html>

bootstraps带参数刷新

data-query-params="tablequeryparams"

function tablequeryparams(params) {

let queryParams = {

limit: params.limit, //页面大小

offset: params.offset, //页码

sort: params.sort, //排序列名

order: params.order, //排位命令（desc，asc）

userId: $("#userId").val(),

checkSource: $("#checkSource").val(),

checkTime: $("#checkTime").val()

};

return queryParams;

}

jquery的ready()与window.onload()

尽管两者都是等到页面加载完成后再执行，但是，前者只需要加载完DOM结构即可执行，后者则是加载时更彻底些(包括图片等)。



[jquery 的ready() 与window.onload()的区别](https://www.cnblogs.com/wsun/p/3916487.html)

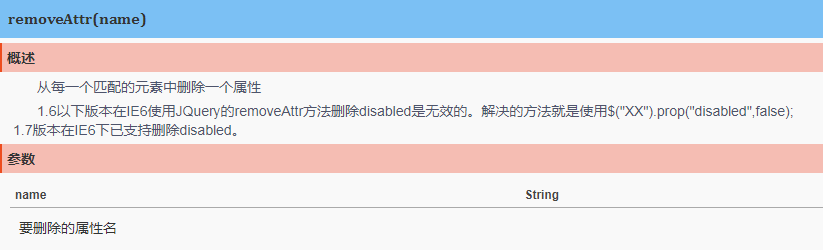
获取上一个父节点[parent([expr])](http://jquery.cuishifeng.cn/parent.html)

获取兄弟节点[prev([expr])](http://jquery.cuishifeng.cn/prev.html)

（expr:增加表达式进行筛选）

移除全部样式

可以使用[removeAttr(“style”)](http://jquery.cuishifeng.cn/removeAttr.html)，这样



移除单个样式

可以使用

css(“样式名”, “”);

**jQuery Validation Engine插件，用于验证表单**

<from id=”fromId”>

<input type=”text” class=”validate[required,maxSize[16]]”

</from>

**如果验证失效，则在进行操作时再校验一次。调用**

$("#fromId").validationEngine("validate")，符 合 条 件 返回true

jQuery绑定多个事件，执行同一方法或者执行不同方法

$(“div”).on(“click keypress”, function() {});

$(“div”).on({

click: function() {},

keypress: function() {}

})

<https://blog.csdn.net/pangchengyong0724/article/details/53186127>

**jQuery(鼠标按下、鼠标松开、鼠标左键点击、鼠标双击)事件**

**mousedown、mouseup、click、dbclick**

[**https://www.jb51.net/article/71712.htm**](https://www.jb51.net/article/71712.htm)

Spring-SpringMVC-MyBatis整合

1.web.xml（SpringIoC初始化、配置文件加载、servlet注册、请求映射）

2.spring-mvc.xml（包扫描、注解驱动、视图解析器）

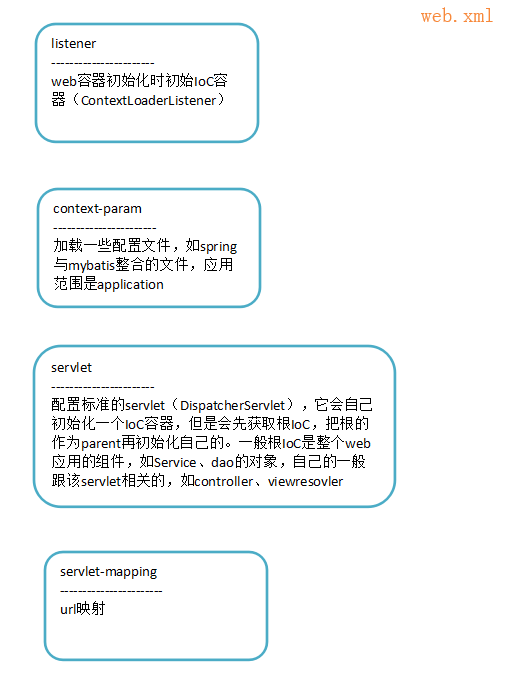
3.spring-mybatis.xml（包扫描、数据源配置（先加载属性文件）、SqlSessionFactory配置、dao接口映射）

4.因为spring-mybatis.xml中的SqlSessionFactory配置中用到了mybatis的配置文件，所以要写下mybatis-config.xml。

注：

使用classpath书写文件路径时禁止带空格，否则报错







[【Spring】浅谈ContextLoaderListener及其上下文与DispatcherServlet的区别](https://www.cnblogs.com/weknow619/p/6341395.html)

创建springboot的三种方式。

1.在线

<https://start.spring.io>

2.通过IDE

IDEA或者STS（SpringToolSuit，Spring 整的开发工具,在 eclipse 的基础上做了些优化,让开发 spring 项目时更爽）都可以。

3.通过maven项目

新建maven项目，添加相关springboot依赖，创建启动类

[创建一个 Spring Boot 项目，你会几种方法？](https://segmentfault.com/a/1190000018843438?utm_source=tag-newest)

springboot屏蔽内置tomcat

1、修改打包方式

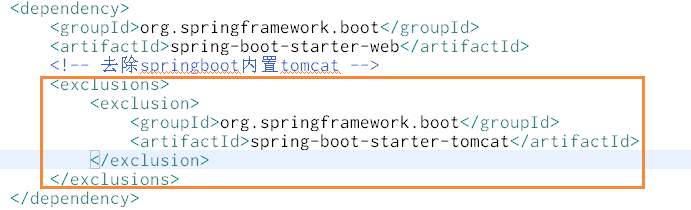
2、屏蔽内置tomcat，两种方式

3、修改启动类，继承SpringBootServletInitializer，重写configure(SpringApplicationBuilder)方法

4、使用eclipse进行打包（war），右键项目，maven clean，maven install

1、<packaging>war</packaging>

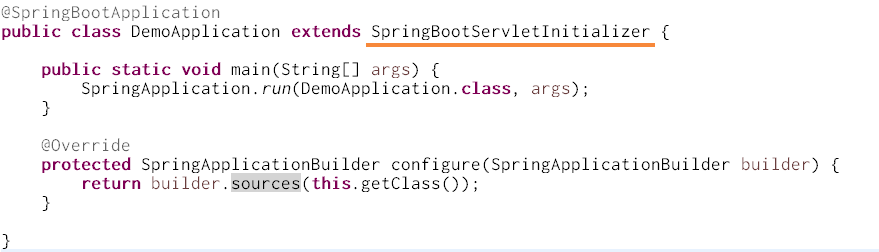
2、



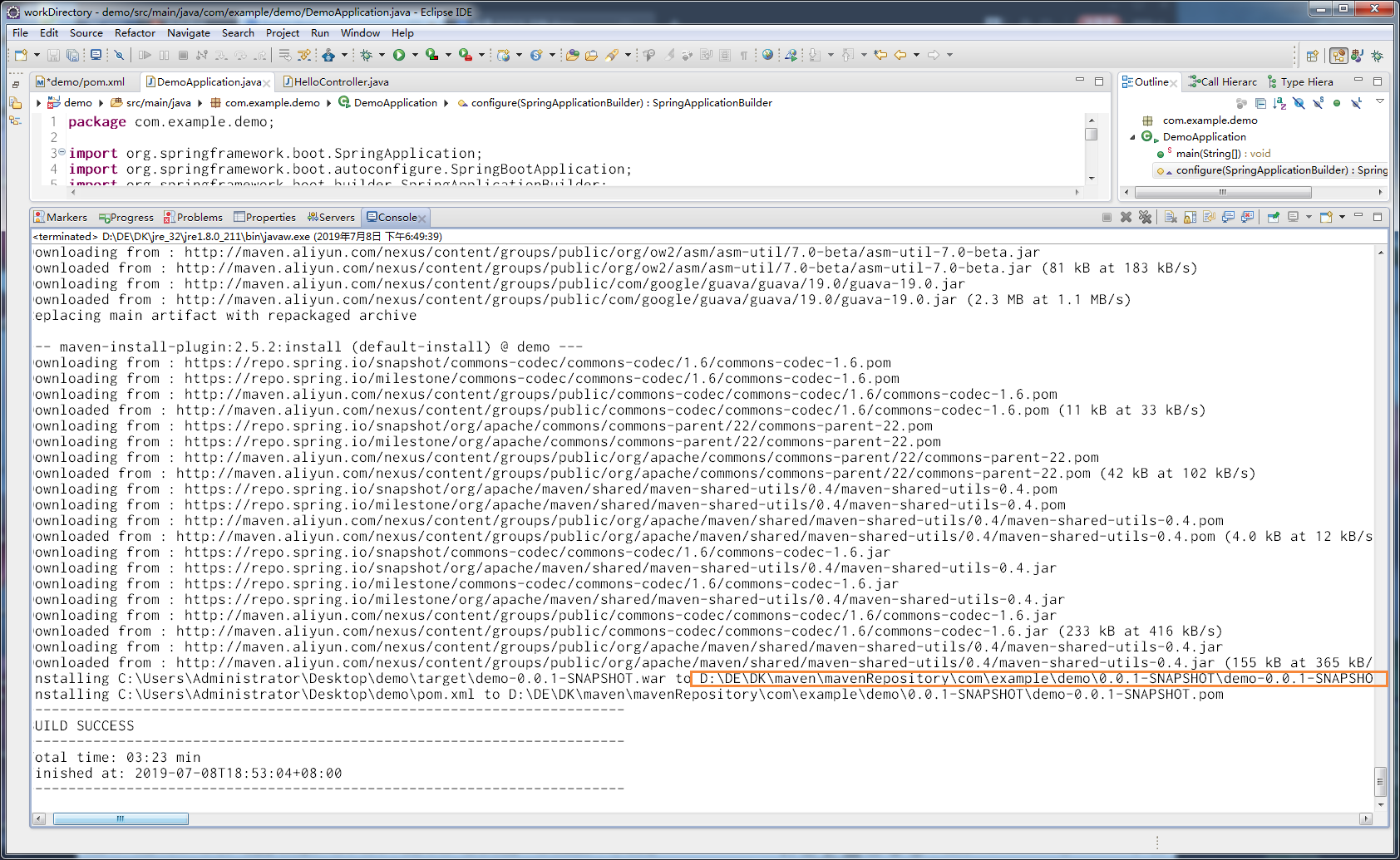
或者这样



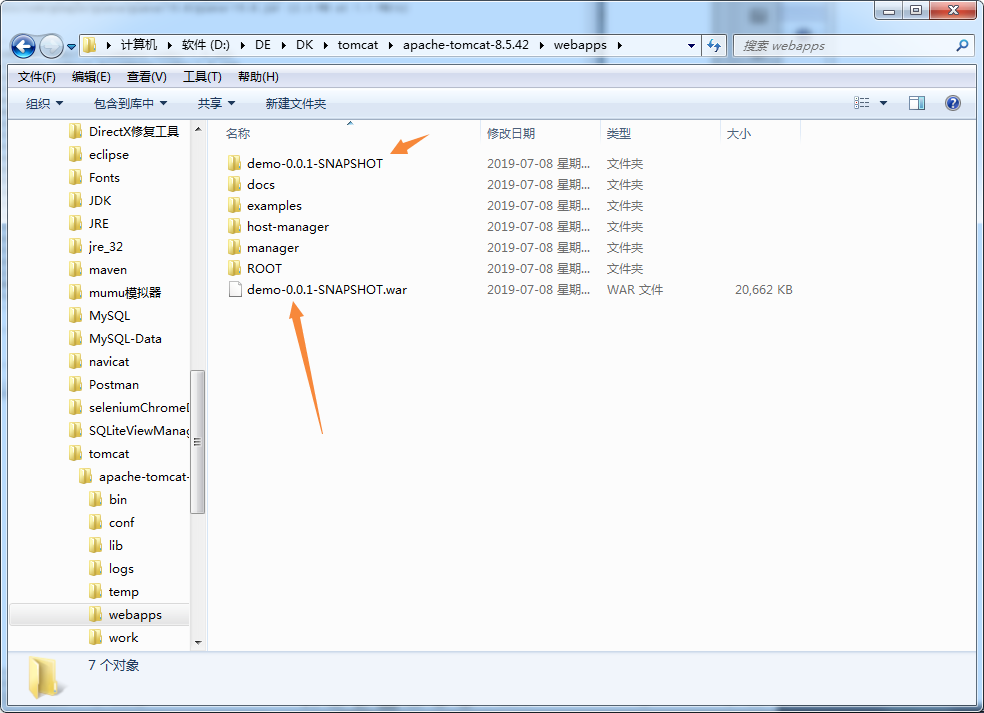
3、



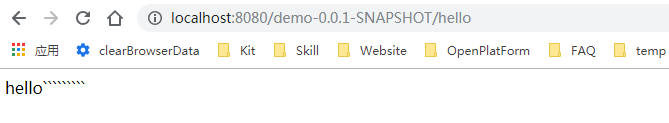
4、



5、放到tomcat的webapps目录下，tomcat自动时自动解压



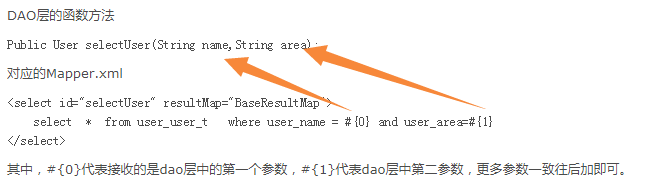
6、访问http://localhost:8080/demo-0.0.1-SNAPSHOT或其目录下路径



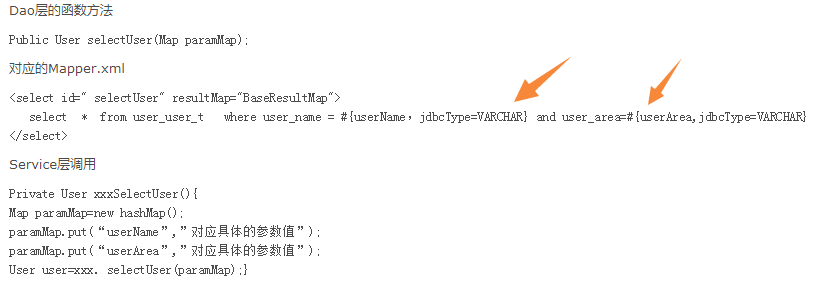
# [mybatis传递参数到mapping.xml](https://www.cnblogs.com/zhuawang/p/5927851.html)

四种方式，注意：参数后面可以省略参数类型，即jdbcType属性

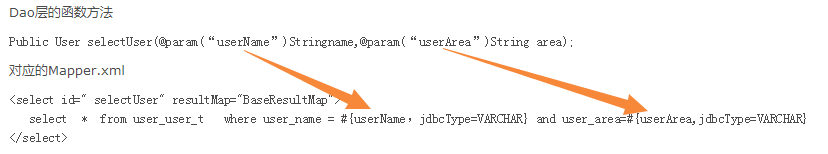
1、参数下标方式，从0开始



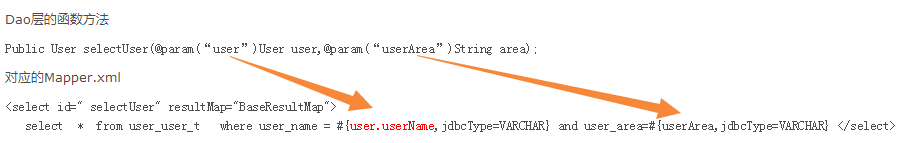
2、map方式



3、@param注解方式之单个参数名传递



4、@param注解方式之参数对象传递



# [mybatis的一对多，多对一，以及多对对的配置和使用](https://www.cnblogs.com/jimisun/p/9414148.html)

一对一关系使用<association></association>标签 // javaType属性

一对多关系使用<collection></collection>标签 // ofType属性

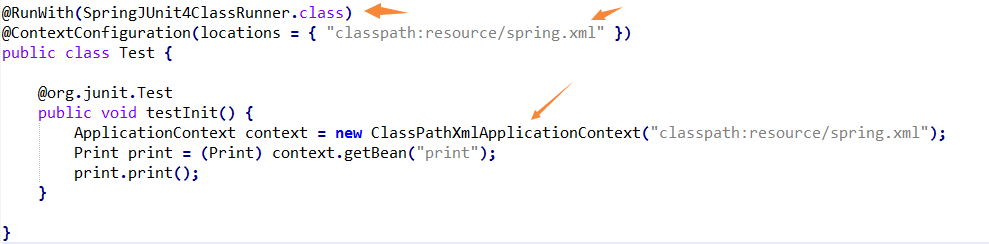
对Spring进行测试

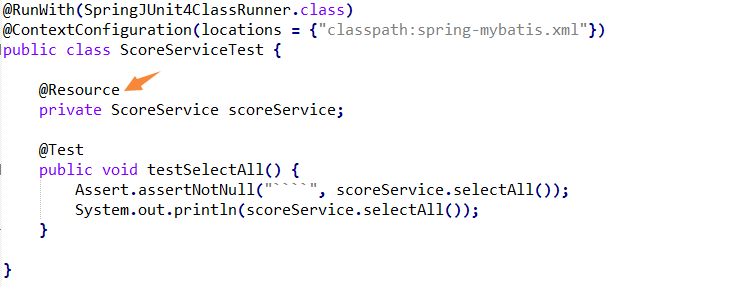
1、修改测试运行器@Runwith(SpringJUnit4ClassRunner.class)，改为spring与junit4整合的测试运行器

2、设置spring上下文环境，加载需要的配置文件

3、通过解析配置文件从容器中获取被管理的bean对象

4、获取被管理的bean对象也可以使用注解的方式进行，如下图2





# [SpringMVC 测试 mockMVC](https://www.cnblogs.com/lyy-2016/p/6122144.html)

1、创建环境（StandaloneMockMvcBuilder和DefaultMockMvcBuilder）

2、发送请求(mockmvc.perform(请求信息))

3、对返回结果进行断言(ResultActions.andExpect)

4、或者进行下一步异常处理(ResultActions.andDo | ResultActions.andReturn)

**1.集成Web环境方式（**MockMvcBuilders.webAppContextSetup(WebApplicationContext context)：指定WebApplicationContext，将会从该上下文获取相应的控制器并得到相应的MockMvc；**）**

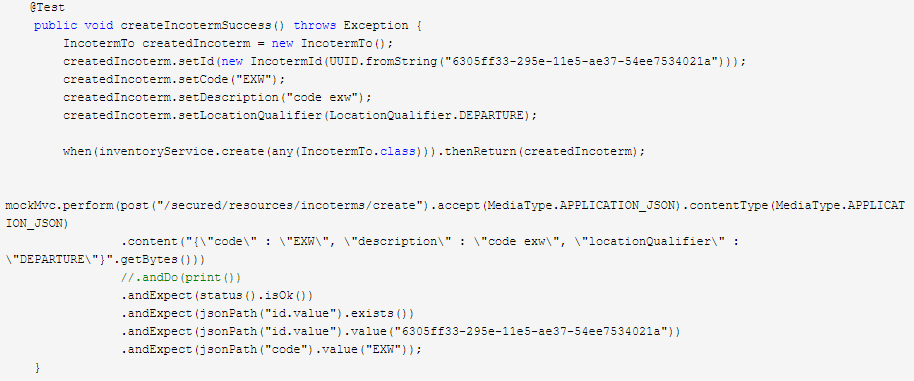
MockMvcBuilder是用来构造MockMvc的构造器，其主要有两个实现：StandaloneMockMvcBuilder和DefaultMockMvcBuilder，分别对应两种测试方式，即**独立安装**和**集成Web环境测试**（此种方式并不会集成真正的web环境，而是通过相应的Mock API进行模拟测试，无须启动服务器）。对于我们来说直接使用静态工厂MockMvcBuilders创建即可。



注意：  
(1)@WebAppConfiguration：测试环境使用，用来表示测试环境使用的ApplicationContext将是WebApplicationContext类型的；value指定web应用的根；  
(2)通过@Autowired WebApplicationContext context：注入web环境的ApplicationContext容器；  
(3)然后通过MockMvcBuilders.webAppContextSetup(context).build()创建一个MockMvc进行测试；

**2.独立测试方式（**MockMvcBuilders.standaloneSetup(Object... controllers)：通过参数指定一组控制器，这样就不需要从上下文获取了；**）**

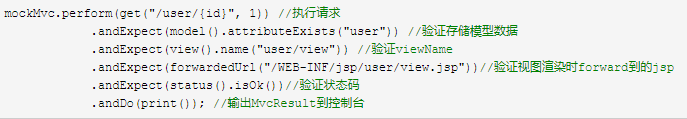
例子



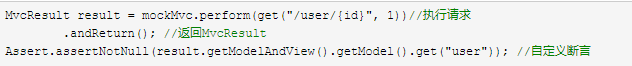
调用MockMvc.perform(RequestBuilder requestBuilder)后将得到ResultActions，通过ResultActions完成如下三件事：  
ResultActions andExpect(ResultMatcher matcher) ：添加验证断言来判断执行请求后的结果是否是预期的；  
ResultActions andDo(ResultHandler handler) ：添加结果处理器，用于对验证成功后执行的动作，如输出下请求/结果信息用于调试；  
MvcResult andReturn() ：返回验证成功后的MvcResult；用于自定义验证/下一步的异步处理；

常用测试例子（前置条件对MockMvcRequestBuilders、MockMvcResultMatchers、MockMvcResultHandles静态导包）：

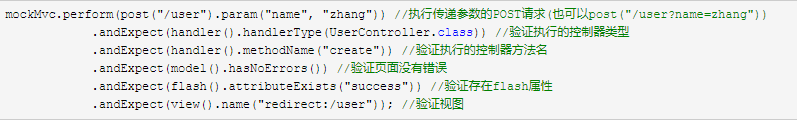
1、测试普通控制器



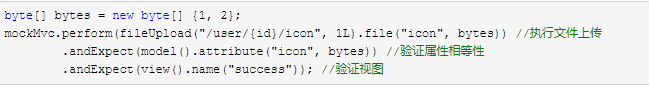
2.得到MvcResult自定义验证



3.验证请求参数绑定到模型数据及Flash属性



4.文件上传



5.JSON请求/响应验证

