# HY框架介绍

四川久远银海软件股份有限公司

互联网产品部 刘惠涛

2016年03月

目录

[HY框架介绍 1](#_Toc445825752)

[零、 框架的由来 2](#_Toc445825753)

[一、Struts2与SpringMVC的区别 3](#_Toc445825754)

[二、HY框架使用说明 5](#_Toc445825755)

[1、框架整体结构介绍 5](#_Toc445825756)

[2、框架目录介绍 6](#_Toc445825757)

[3、关于Apache Shiro安全管理 7](#_Toc445825758)

[4、关于码表 8](#_Toc445825759)

[5、关于分页 9](#_Toc445825760)

[三、HY框架前台界面 9](#_Toc445825761)

[1、总体介绍 9](#_Toc445825762)

[2、标准jsp引用 9](#_Toc445825763)

[3、关于加快前台界面渲染速度 10](#_Toc445825764)

[四、代码片段 15](#_Toc445825765)

[1、BaseController.java 15](#_Toc445825766)

[2、SpringMVC当中文件上传 16](#_Toc445825767)

[3、Shiro管理的session 17](#_Toc445825768)

[4、SpringMVC websocket配置 17](#_Toc445825769)

## 框架的由来

在现今的互联网项目中，越来越多的企业和个人采用SpringMVC作为自己的MVC框架。这是因为Spring的影响力和魅力，也是因为SpringMVC对于原生的restful风格支持，在这一点上，真的是让许多程序员们无法自拔。

我们为什么要做这个新的框架？

首先，我们从框架结构上，TA框架采用struts2作为MVC当中的‘C’，同时集成了润乾报表（在Ta3+后续的版本已经可以选择来配置是否使用润乾），在互联网项目中这几乎是不可见的，去掉这两部分使得框架瘦身不少。Ibatis相对于Mybatis来说也是如此，Mybatis是Ibatis的升级，无论从性能上，代码便利度上还是适用性上都要优于Ibatis。

第二点，相对于传统的组织结构，用户，角色，权限。在现在互联网项目中，并没有TA框架所实现的那么复杂。一个简单的互联网后台应用很可能是没有组织架构乃至于分中心这些概念的。在此框架中，并没有否认这些概念，但是也并没有实现，等待开发者根据具体项目需求去关联实现。

举个例子，在TA框架中（MYSQL）版本，菜单，角色，人员，部分均是外键关联的存在，这使得如果不是特别熟悉TA的人来说，上手比较困难，使得在使用TA框架的过程中，经常会抛出一些无法预知的错误。

第三点，SpringMVC性能上由于Struts2。这一点后面会提到。

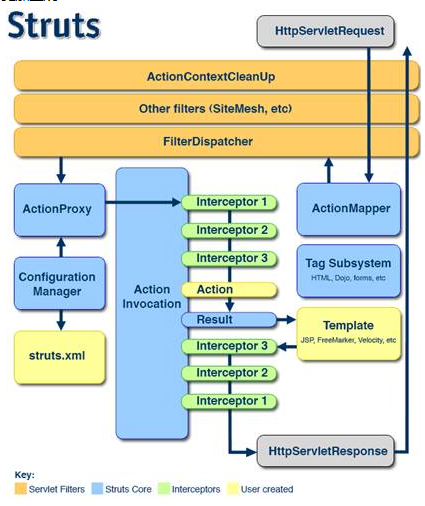
第四点，在互联网技术高速发展的前提下，对程序员的要求越来越高。高效、简洁、快速的开发是必不可好的。SpringMVC的代码更加简洁、清晰。从一个简单的文件上传就可以看得出来。（后面会贴上代码）

最后，对于分布式、集群的需求。SpringMVC对于分布式、集群更加友好。使用SpringMVC可以非常简单的集成使用类似于Dubbo这一类的分布式框架。Dubbo文档明确指出，推荐使用Spring AOP的方式,将Dubbo服务提供者暴露的服务注册到Spring 容器当中,SpringMVC可以毫无压力的拿来使用。

基于以上种种，HY框架诞生了。接下来将一一介绍该框架一些其他的特性。

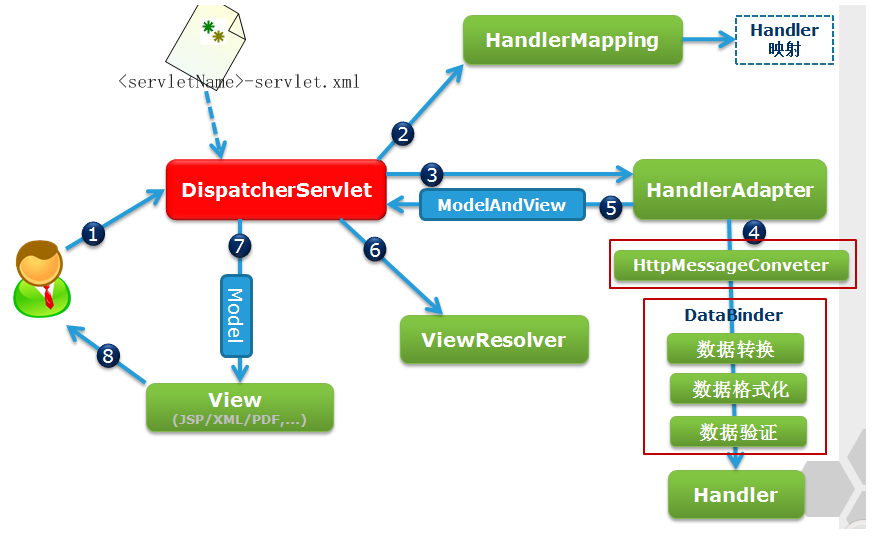
## 一、Struts2与SpringMVC的区别

我们先来看Struts2以及spring mvc的流程图



图例1：Struts2流程图

这是我们非常熟悉的Struts2的流程图，在当前互联网应用开发的环境下，使用一个更加高效、安全、简易，并且更具扩展性的mvc框架是非常有必要的。因此我们选择了springmvc。下面再看springmvc的流程图：



图例2：springmvc流程图

我们可以看到springmvc的设计思路是将整个处理流程规范化，并把每一个处理步骤分派到不同的组件中进行处理，处理流程规范化是目的，对于处理过程的步骤划分和流程定义则是手段。因而处理流程规范化的首要内容就是考虑一个通用的Servlet响应程序大致应该包含的逻辑步骤：

* 步骤1—— 对Http请求进行初步处理，查找与之对应的Controller处理类（方法）   ——HandlerMapping
* 步骤2—— 调用相应的Controller处理类（方法）完成业务逻辑                    ——HandlerAdapter
* 步骤3—— 对Controller处理类（方法）调用时可能发生的异常进行处理            ——HandlerExceptionResolver
* 步骤4—— 根据Controller处理类（方法）的调用结果，进行Http响应处理       ——ViewResolver

我们从SpringMVC的特点来说明它与Struts2的区别，通过这些相信大家应该对SpringMVC有一个很好的了解：

1. 从设计角度来说，SpringMVC更加清晰。即使我们去对比Struts2的原理图和SpringMVC的类图，它依然让人很困惑，远没有SpringMVC更加直观。
2. spring会稍微比struts快。spring3 mvc是方法级别的拦截，拦截到方法后根据参数上的注解，把request数据注入进去，在spring3 mvc中，一个方法对应一个request上下文。而struts2框架是类级别的拦截，每次来了请求就创建一个Action，然后调用setter getter方法把request中的数据注入。
3. 和Spring 其他框架无缝集成，是其它Web框架所不具备的。我相信你的项目肯定少不了spring。
4. 可适配，通过HandlerAdapter可以支持任意的类作为处理器。
5. 可定制性，HandlerMapping、ViewResolver等能够非常简单的定制。
6. 功能强大的数据验证、格式化、绑定机制。
7. 利用Spring提供的Mock对象能够非常简单的进行Web层单元测试。
8. SpringMVC对于RESTful风格的支持。再也不必使用urlRewrite这类东东了。
9. 简单的文件上传。SpringMVC当中的文件上传比Struts2方便了不少。框架当中有具体的例子。
10. 基于注解的零配置支持。
11. Ajax请求的友好支持。在SpringMVC当中，我们通过使用@RestController @ResponseBody注解，方法返回Object即可做到友好的Ajax返回。
12. SpringMVC的代码更加简洁、清晰。相信通过一定时间的使用大家都能有切身的体会。
13. 最后一点，也是我觉得最重要的一点。由于Spring的特性，Spring能够很好的与Spring之外的框架、组件集成。Spring与SpringMVC又是一个上下文Context的关系（可以理解为继承的关系），SpringMVC能够使用上一级Context当中的内容。这些技术包括但不限于以下几种：shiro、redis、ehcache、websocket、rabbitMq、mybatis等等。

## 二、HY框架使用说明

### 1、框架整体结构介绍

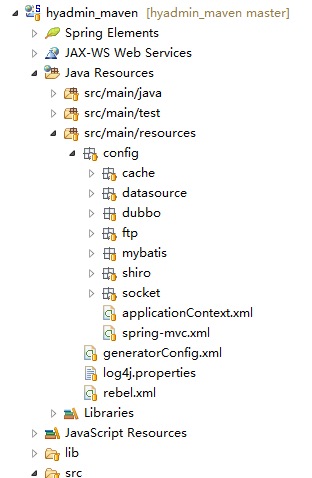
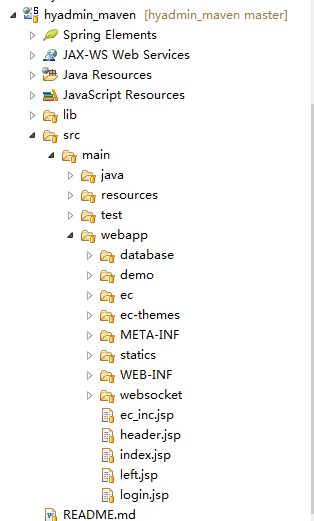
框架整体采用springmvc+spring+mybatis(SSM)的架构,集成了shiro,druid,websocket,ehcache,redis等..HY框架是一个易集成易拆分的整体，我们可以快速的集成其他的引用，我们也能快速的删除已经集成的引用。

安全方面:采用Apache Shiro进行WEB安全管理,实现了session独立管理、以及基于角色到菜单的权限管理,并且支持增删改查级别的权限管理,可将权限精确到Controller的方法一级。

同时对异常采用log4j记录，对系统内的同步、异步异常均已处理，业务代码内无需自行处理。除如特殊需求外需要自己处理的异常，应该在自行处理完成后，视情况抛出相应的异常，供系统记录。

认识此框架请从web.xml开始,然后applicationContext.xml 读懂配置文件是关键。

### 2、框架目录介绍



此次框架结构采用Maven项目结构，方便项目最终转变成maven项目。通过maven我们可以很好的进行项目依赖、版本管理等等。

重点要说明的是resources/config目录（整个框架的配置文件，包括spring以及其他插件的配置文件，对于一些业务的组件扫描可能需要自行添加到扫描的范围内）：

* applicationContext.xml spring容器加载的主要(默认配置文件),在这里是spring所有配置文件的汇总（入口）。
* spring-mvc.xml springmvc配置文件,包括异常 文件上传 视图解析 !!!重要的是需要扫描的Controller包路劲要在这里面加
* cache/\* 缓存相关配置文件
* datasource/\* 数据库连接相关配置文件
* ftp/\* ftp相关文件
* mybatis/\* mybatis配置文件,xml映射文件
* shiro/\* Apache Shiro相关配置文件包括 自定义密码验证 自动以BaseRealm类 securityManager 以及各种拦截器 权限等.. 重要的是需要进行安全管理的url要在此处添加。
* socket/\* websocket配置文件。

### 3、关于Apache Shiro安全管理

权限采用Apache Shiro框架，具体学习可参考

<http://shiro.apache.org/>

<http://jinnianshilongnian.iteye.com/blog/2018398>

之所以选择Shiro 而不是Spring Security，原因在于Shiro相较于Spring Security来说更加轻量级，学习成本更低。同时功能亦是满足互联网应用的需求的。这也是绝大多数WEB应用选择它的原因。它通过配置拦截器，Spring的Bean可以轻松集成到HY框架当中。

#### 2.3.1 session管理

通过Shiro的会话管理模块，将session从servlet容器中独立出来自行管理。默认情况下，session通过shiro对ehcache的支持放在缓存当中。可以通过修改applicationContext.xml中redis session相关配置文件可将session存放于redis当中。（已实现，具体实现请参考cache相关配置文件）

也可以通过此基础，实现单点登陆、session共享（暂未实现）。

#### 2.3.2 Controller

SpringMVC当中Controller相当于struts2当中的action。但是为了达到之前所说的权限控制力度，对于Controller做了以下约定：

1.Controller统一采用注解配置 @Controller标明一个Controller组件

@RestController标明一个rest风格的Controller组件，其间所有方法都是异步的。

2.Controller注解后紧跟@RequestMapping("system/menu/menuManager")标明组件'系统/包名/功能',之后在各个方法之上@RequestMapping("/default")标明方法路劲.

注意:在此约定,默认的进入页面的方法都是default,并且不可缺少

3.对于增删查改可用的路劲头的约定：

增:create、new、save、add

删：delete、remove

查：query、view、select、get、got

改：update、edit、fix

#### 2.3.3 关于Shiro标签

在需要的jsp当中引用<%@ taglib uri="http://shiro.apache.org/tags" prefix="shiro"%>

可使用shiro提供的权限标签。可隐藏或者显示相关的资源、按钮等。。

### 4、关于码表

码表是有的 但是不是一开始就全部加载到内存当中,只有当第一次查询的时候才会查出来放到内存当中。（具体实现请参考cache当中相关spring-ehcache配置文件）

### 5、关于分页

表格分页已经通过mybatis插件的形式,集成到baseService当中,只需要继承直接使用即可 sql只需要写一个查询的sql即可。

需要注意的是.分页的sql id当中必须包含listPage字符串,这是插件生效的条件。

注意：该分页只适用于easyui的表格分页

## 三、HY框架前台界面

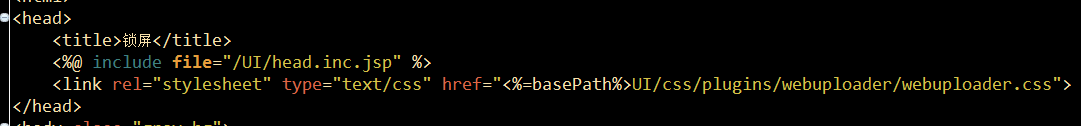
### 1、总体介绍

框架前台集成了bootstrap以及bootstrap一些相关插件。（插件列表稍后提供）。

表格采用easyui表格。封装了easyui表格、tab页、tree通过tag标签的形式放在WEB-INF下面以供使用。

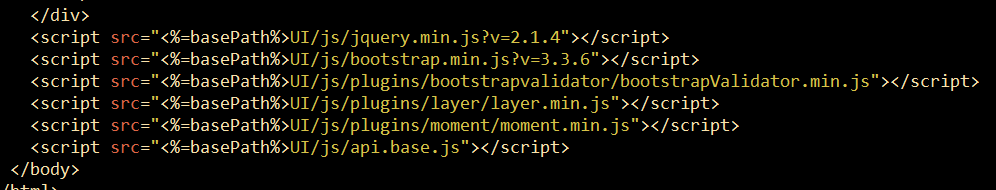
建议:不管是bootstrap的插件还是easyui的插件,使用都会造成一定的性能开销，建议做到需要用的才引用，尤其是标签。如果你对easyui的组件使用已经比较熟悉，建议就不要再用tag了。使用jsp的tag所造成的开销远远大于js、css等静态文件的引用。总结：引用的静态资源越少越好，js、css尽量压缩，image图片资源请使用雪碧技术，javascript代码尽量不要放在jsp当中，请在静态文件当中引用，因为js是可以压缩传输的。（后面会讲到）

### 2、标准jsp引用









注意:js要引用在文档的末尾。Body之前

如果你不想已jsp的形式引用js文件,可以每个页面都单据引用需要的js资源。Basic.inc.jsp当中的就是必须的js资源。加上你想要使用的其他plugin即可。

Plugins存放路：UI/js/plugins 对应的插件样式：UI/css/plugins

### 3、关于加快前台界面渲染速度

#### 3.3.1 静态资源压缩（第一次压缩）

不管是bootstrap的插件还是easyui的插件,使用都会造成一定的性能开销，建议做到需要用的才引用，尤其是标签。如果你对easyui的组件使用已经比较熟悉，建议就不要再用tag了。使用jsp的tag所造成的开销远远大于js、css等静态文件的引用。总结：引用的静态资源越少越好，js、css尽量压缩，image图片资源请使用雪碧技术，javascript代码尽量不要放在jsp当中，请在静态文件当中引用，因为js是可以压缩传输的。

#### 3.3.2 开启tomcat gzip功能（第二次压缩）

HTTP 压缩可以大大提高浏览网站的速度，它的原理是，在客户端请求服务器对应资源后，从服务器端将资源文件压缩，再输出到客户端，由客户端的浏览器负责解压缩并 浏览。相对于普通的浏览过程HTML ,CSS,Javascript , Text ，它可以节省40%左右的流量。更为重要的是，它可以对动态生成的，包括CGI、PHP , JSP , ASP , Servlet,SHTML等输出的网页也能进行压缩，压缩效率也很高。

修改%TOMCAT\_HOME%/conf/server.xml，修订节点如下：

<Connector port="80" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" executor="tomcatThreadPool" URIEncoding="utf-8"

compression="on"

compressionMinSize="2048" noCompressionUserAgents="gozilla, traviata"

compressableMimeType=" text/html,text/xml,text/javascript,application/x-javascript,application/javascript,text/css,text/plain" />

•compression="on" 打开压缩功能

•compressionMinSize="2048" 启用压缩的输出内容大小，默认为2KB

•noCompressionUserAgents="gozilla, traviata" 对于以下的浏览器，不启用压缩 •compressableMimeType　哪些资源类型需要压缩，我们只压缩html、css、Js

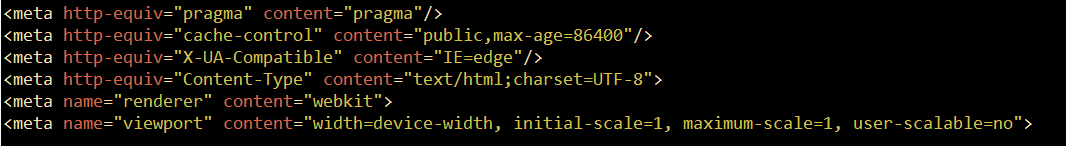
#### 3.3.3 启用HTTP协议当中的缓存

启用浏览器缓存的好处

* 减少了冗余的数据传输，节省了流量。
* 2. 减少了服务器的负担， 大大提高了网站的性能 。
* 加快了客户端加载网页的速度。



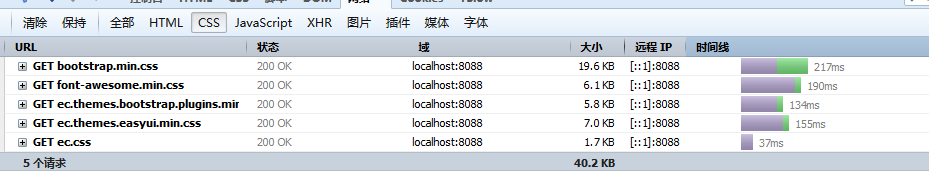
HY框架中的界面缓存配置统一设置在ec.themes.import.jsp当中。

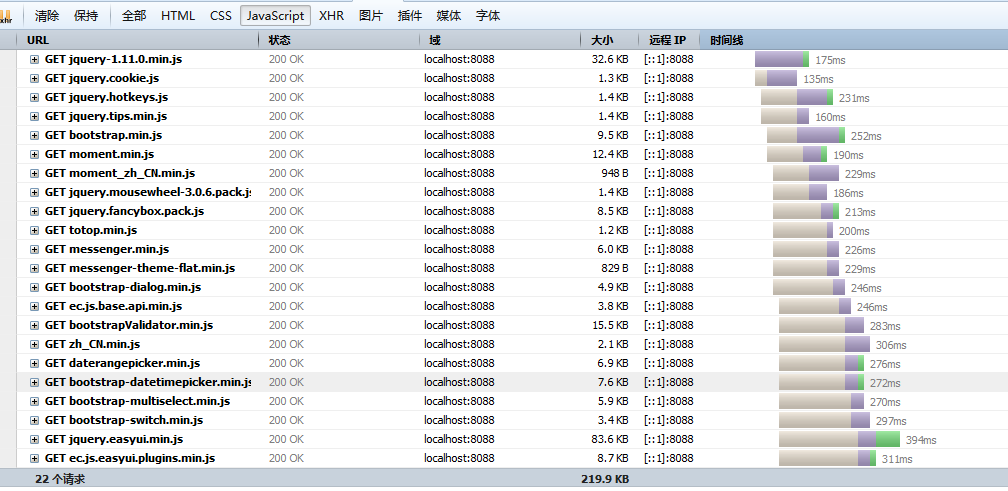


max-age=86400 30分钟，30分钟后会从服务器重新发起请求。

#### 3.3.4 效果截图

当以上三部都优化以后，我们Ctrl+F5强制刷新会看到以下效果：



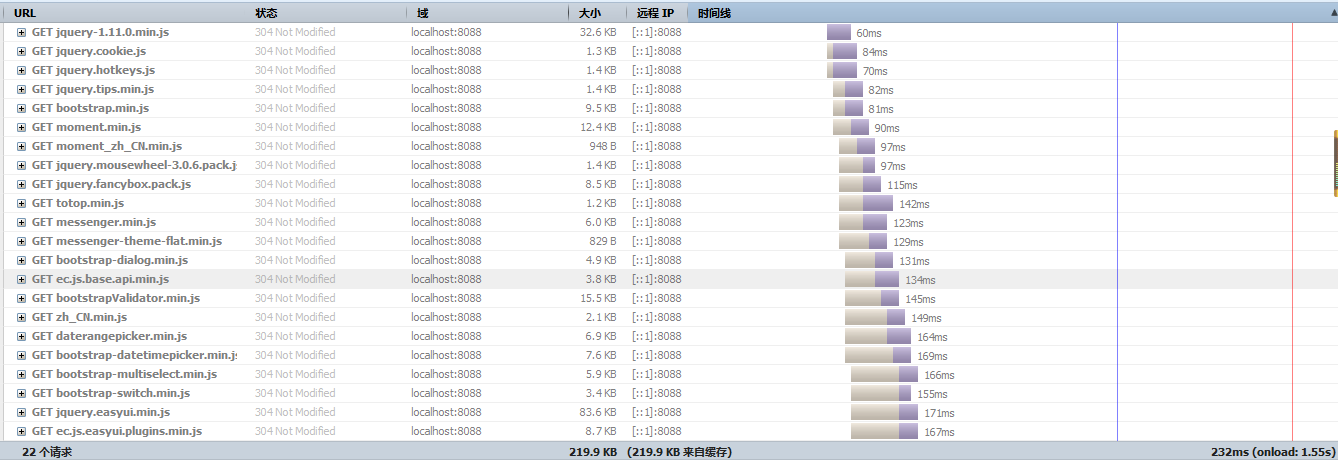


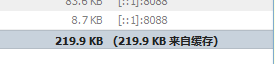
强制刷新，我们看到get请求200，是从服务器请求的数据。但是此时我们已经可以看到gzip压缩是有效的。



Jquery-1.11.0.min.js的大小从95kb压缩到了32.6kb,这个压缩比是相当大的。我们可以注意到此时的请求头信息是Cache-Control=no-cache的。

此时我们再次点击刷新按钮（非强制刷新）



我们可以看到，状态码变成了304 not modified，说明资源来自缓存。并且是压缩gzip压缩过后的资源。

这样就大大降低了服务器的压力。加快了客户端网页的加载速度。效果还是很明显的。

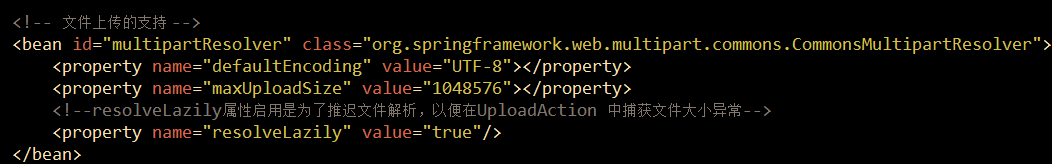
## 四、代码片段

### 1、BaseController.java

BaseController是所有Controller需要继承的类。它封装了参数（PageParam）、异步返回bean（ResultBean）、用户信息（UserDomain）、Shiro（session，subject）、上传文件、获取码表。你也可以自定义属于自己的BaseController。但是Shiro的SecurityUtils用法都是一样的，建议封装起来，不要每个Controller都去直接取。

### 2、SpringMVC当中文件上传

#### 4.1.1 spring-mvc.xml

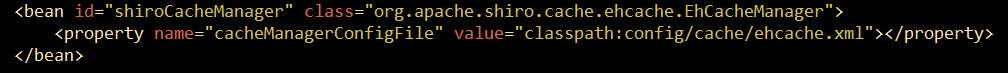


#### 4.1.2 Controller



### 3、Shiro管理的session

#### 4.3.1 定义shiroCacheManager



#### 4.3.2 定义sessionDao



#### 4.3.3 sessionDao注入sessionManager

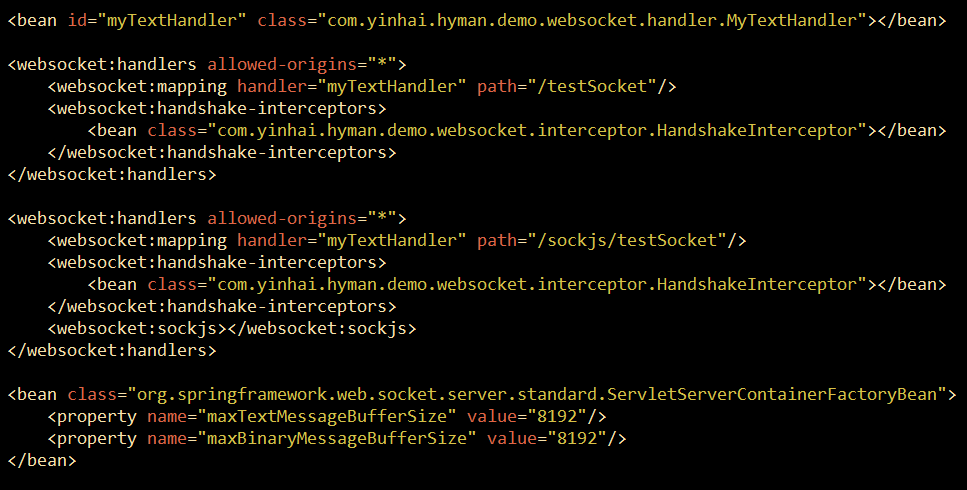
#### 4.3.4 sessionManager统一给到Shiro SecurityManager

### 4、SpringMVC websocket配置

#### 4.4.1 注解配置

参照WebSocketConfigurer.java，但是要在spring-mvc xml当中添加该注解配置的扫描。

#### 4.4.2 XML配置



XML配置生效以后，在前台就可以直接与websocket建立连接。具体请参照webapp/websocket/index2.html

## 五、数据权限

### 1、数据权限说明

框架中将数据权限绑定到角色权限当中,每个角色有单独的菜单权限以及数据权限。

菜单权限负责菜单的显示，以及更加细颗粒度的Controller层的增删查改的权限。具体参照2.3.2Controller。

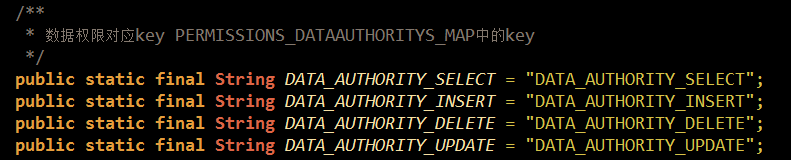
数据权限是单独的模块，与角色绑定。每个角色有自己单独的数据权限。

### 2、具体体现

用户登陆后，第一次访问url资源时，就会通过用户角色加载用户的菜单权限以及数据权限。

菜单权限放到shiro当中，不需要我们自己维护。

数据权限以ConcurrentMap放到shiro session当中。Key为 HYconst.PERMISSIONS\_DATAAUTHORITYS\_MAP。在这个map当中存放了4个key，分别对应该角色增删查改的权限，是以Set<String>方式存放数据库table\_name。



数据权限的具体使用，并没有实现，各个项目可根据自己的需要来使用，建议可以通过mybatis的拦截器来实现，获取jdbc的监听实现。