

😊 1、spring-shiro-training项目简介与导入

📖 公众号java-mindmap

git地址

<http://git.oschina.net/wangzhixuan/spring-shiro-training>

官方简介

基于Maven构建的springmvc、spring、mybatis-plus、shiro、log4j2、easyui简单实用的权限系统

演示地址

<http://shiro.dreamlu.net> 🔗

账号test，密码test

导入过程

1 复制克隆地址，打开eclipse通过git导入项目。

2 把database目录下的shiro_mysql.sql导入mysql数据库

3 修改/spring-shiro-training/src/main/conf/dev/config/application.properties
配置项中数据库的信息

4 右键项目Dubug as输入tomcat7:run启动项目

😊 2、主要技术概况

公众号java-mindmap

前端

easyui

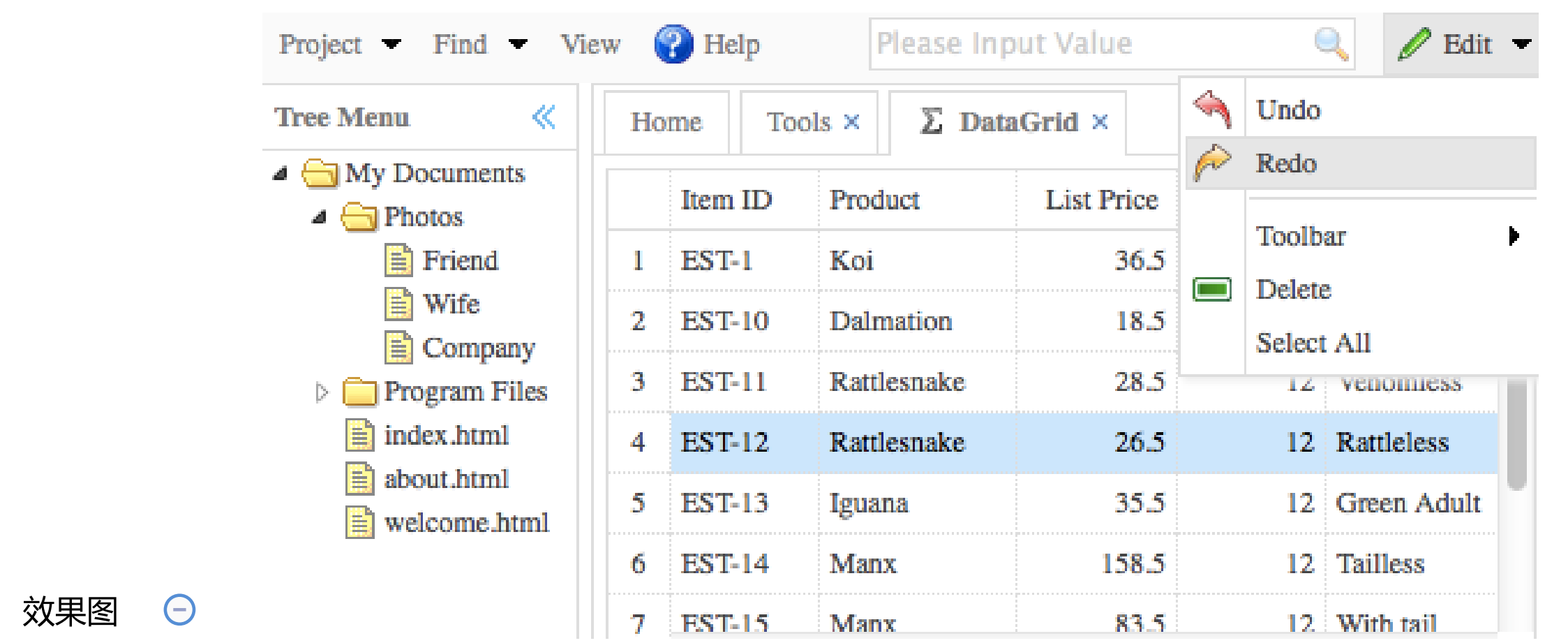
文档地址 <http://www.jeasyui.com/documentation/index.php>

效果图

好处

jQuery EasyUI是一组基于jQuery的UI插件集合，而jQuery EasyUI的目标就是帮助web开发者更轻松的打造出功能丰富并且美观的UI界面。

My97DatePicker 一个简单实用的时间日期组件



Item ID	Product	List Price			
1	EST-1	Koi	36.5		
2	EST-10	Dalmation	18.5		
3	EST-11	Rattlesnake	28.5		
4	EST-12	Rattlesnake	26.5	12	Rattleless
5	EST-13	Iguana	35.5	12	Green Adult
6	EST-14	Manx	158.5	12	Tailless
7	EST-15	Manx	83.5	12	With tail

日志

log4j2

简介

与log4j的区别

配置文件不同

文件渲染方式不同

log4j 2采用的是.xml，json或者.json

log4j主配置文件是.properties；

log4j想要生效，我们需要在web.xml配置监听

log4j2只需要把log4j2.xml放到工程resource目录下

<http://blo...>

apache的一个开源日志管理工具，可以记录开发阶段的调试信息、运行时的日志记录及错误等

框架

spring

特征

轻量

控制反转

面向切面

容器

spring mvc

与struts2区别（来着网络）

spring MVC和spring是无缝的

Struts2是类别级的拦截，SpringMVC是方法级别的拦截

SpringMVC的入口是servlet，而Struts2是filter

SpringMVC开发效率和性能高于Struts2

√进行更简洁的Web层的开发；

√天生与Spring框架集成（如IoC容器、AOP等）；

√提供强大的约定大于配置的契约式编程支持；

√能简单的进行Web层的单元测试；

√支持灵活的URL到页面控制器的映射；

mybatis-plus

与mybatis相比的优势

一款MyBatis的增强 crud 工具包，简化 增 删 改 查 操作。启动加载 XML 配置时注入单表 SQL 操作，为简化开发工作、提高生产率而生。Mybatis-Plus 启动注入非拦截实现、性能更优。

增强了mybatis通用单表操作的能力，只需要继承通用的service或者dao就能拥有单表增删改查的能力

Mybatis-plus 启动注入非拦截实现、性能更优

多种 ID 选择，类型 ID_WORKER，UUID 全局唯一、支持实现分库

完全继承原生，易理解、扩展使用，支持自定义表、字段、ID动态注解

<http://mp...>

shiro

一个强大易用的Java安全框架,提供了认证、授权、加密和会话管理等功能，可为任何应用提供安全保障 - 从命令行应用、移动应用到大型网络及企业应用

替换DBCP和C3P0。Druid提供了一个高效、功能强大、可扩展性好的数据库连接池。

可以监控数据库访问性能，Druid内置提供了一个功能强大的StatFilter插件，能够详细统计SQL的执行性能，这对于线上分析数据库访问性能有帮助。

数据库密码加密。直接把数据库密码写在配置文件中，这是不好的行为，容易导致安全问题。DruidDruiver和DruidDataSource都支持PasswordCallback。

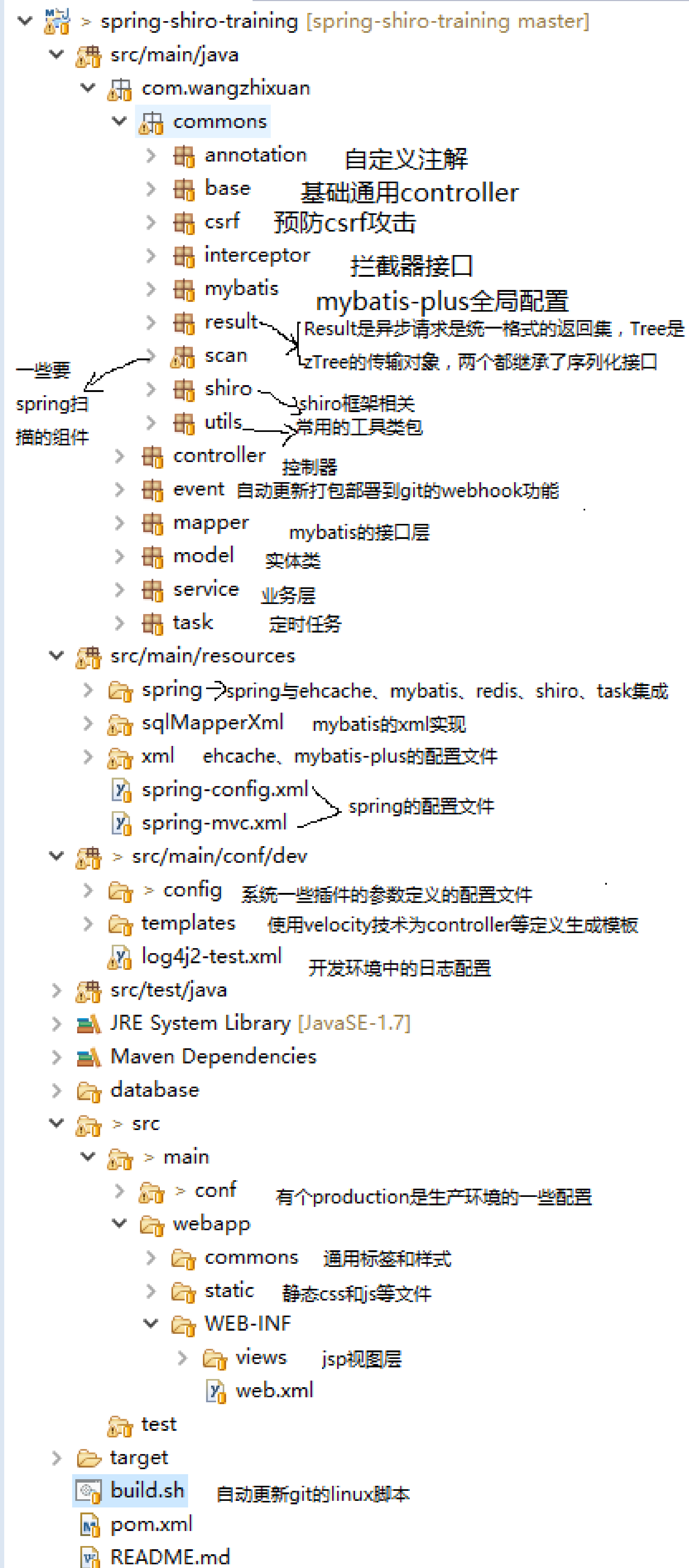
SQL执行日志，Druid提供了不同的LogFilter，能够支持Common-Logging、Log4j和JdkLog，你可以按需要选择相应的LogFilter，监控你应用的数据库访问情况。

扩展JDBC，如果你要对JDBC层有编程的需求，可以通过Druid提供的Filter机制，很方便编写JDBC层的扩展插件。

<http://...>

😊 3、项目结构解析

公众号java-mindmap



😊 4、项目常用注解

公众号java-mindmap

spring与 spring mvc

- @Controller 用于标注控制层组件
 - @Service 服务层组件，用于标注业务层组件
 - @Component 泛指组件，当组件不好归类的时候，我们可以使用这个注解进行标注
 - @GetMapping method为get请求的@RequestMapping简写
 - @PostMapping method为post请求的@RequestMapping简写
 - @RequestMapping 处理请求地址映射的注解，可用于类或方法上
用于类上，表示类中的所有响应请求的方法都是以该地址作为父路径
 - @Autowired 对类成员变量、方法及构造函数进行标注，完成自动装配的工作
通过 @Autowired的使用来消除 set，get方法
默认是按照byType进行注入的
 - @ResponseBody 表示该方法的返回结果直接写入HTTP response body中
一般在异步获取数据时使用，在使用@RequestMapping后，返回值通常解析为跳转路径，加上@responsebody后返回结果不会被解析为跳转路径，而是直接写入HTTP response body中。比如异步获取json数据，加上@responsebody后，会直接返回json数据。
 - @Scheduled spring定时任务注解，使用cron表达式
 - @InitBinder 绑定请求参数到指定的属性编辑器，解决类型的转换问题
- http://...

spring aop

- @Pointcut 声明切入点表达式，一般在该方法中不再添加其他代码
 - @Around 环绕通知，围绕着方法执行
 - @Aspect 配置在类上，把这个类声明为一个切面：需要把该类放入到IOC容器中，再声明为一个切面
 - @Order 指定切面的优先级，当有多个切面时，数值越小优先级越高
- http://sh...

mybatic plus

- @TableName 表名注解
 - @TableField 字段注解
 - @TableId 主键注解
- http://mp.bao...

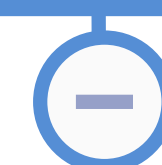
其他

- @CsrfToken 完成检测登录时是否带有token，反正跨站请求伪造攻击
 - @SuppressWarnings 对被批注的代码元素内部的某些警告保持静默
 - @JsonProperty 此注解用于属性上，作用是把该属性的名称序列化为另外一个名称，如把trueName属性序列化为name，@JsonProperty("name")。
 - 拓展
 - @JsonIgnore 此注解用于属性上，作用是进行JSON操作时忽略该属性。
 - @JsonFormat 此注解用于属性上，作用是把Date类型直接转化为想要的格式，如@JsonFormat(pattern = "yyyy-MM-dd HH-mm-ss")。
- http://blog...

😊 5、项目功能

📁 公众号java-mindmap

权限管理



角色管理

用户管理

部门管理

资源管理

日志监控



图标管理

登陆日志

Druid监控

登录退出

😊 6、代码生成器

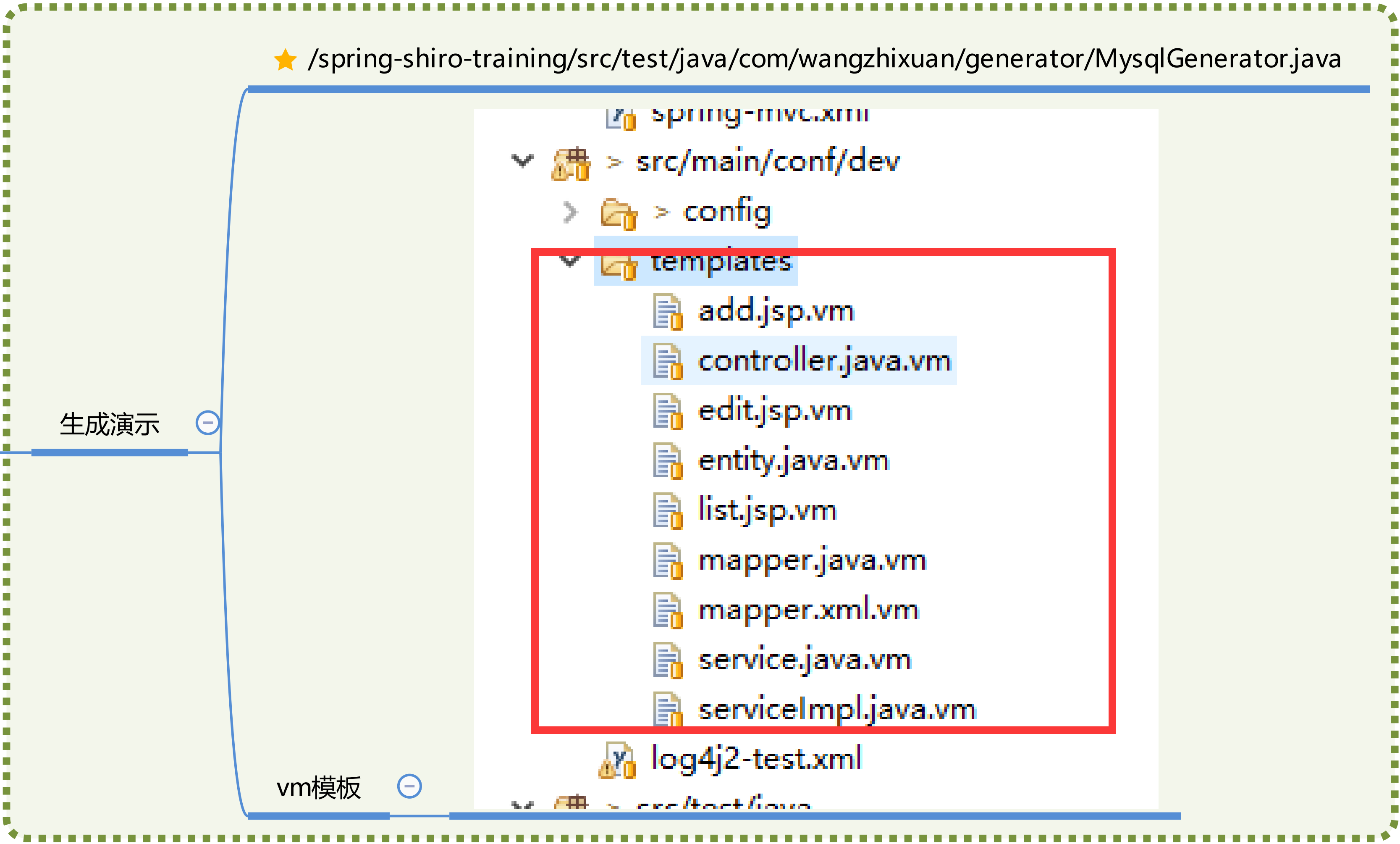
公众号java-mindmap

velocity简介

http://blog.cs...

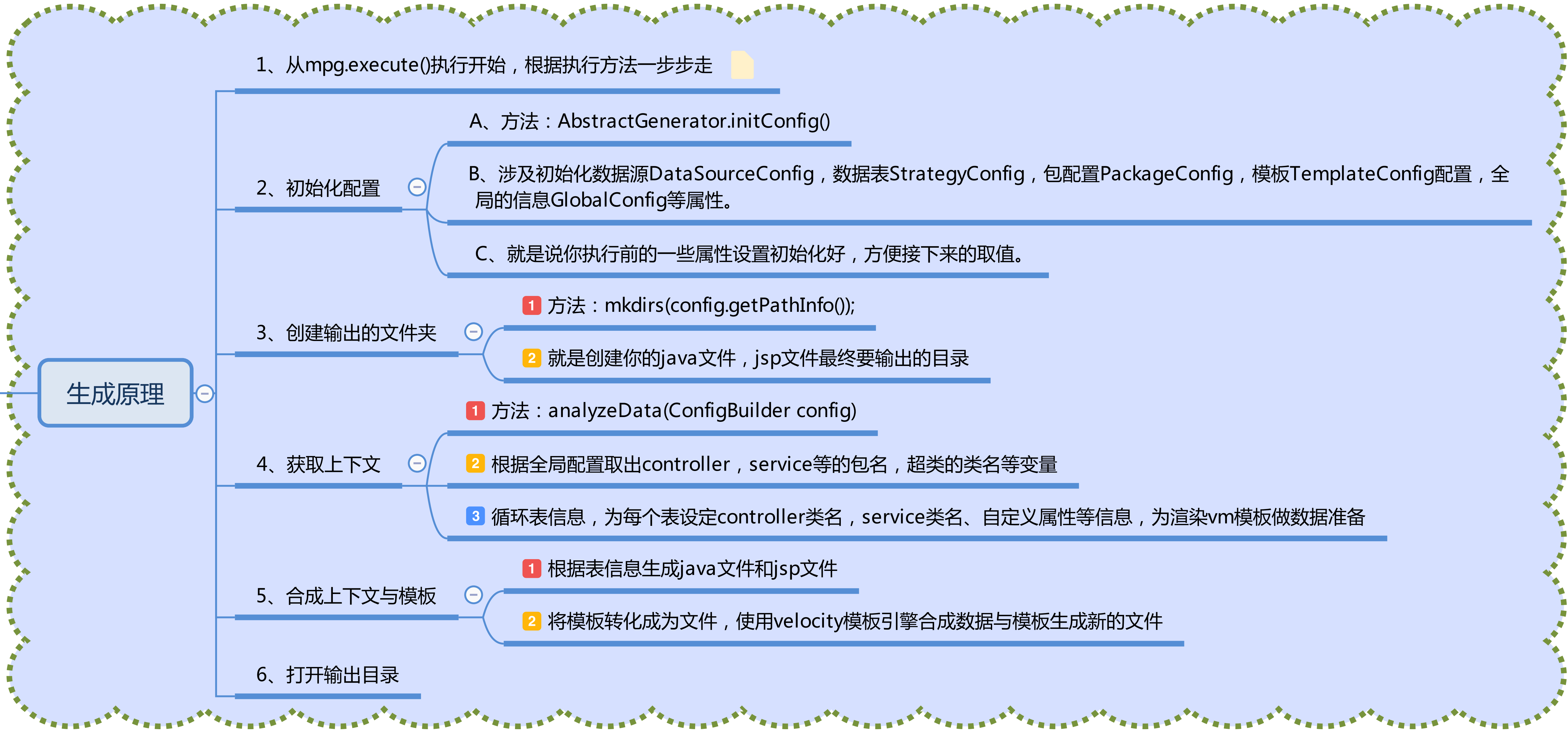
一个基于Java的模板引擎（template engine）。它允许任何人仅仅简单的使用模板语言（template language）来引用由java代码定义的对象。

main方法



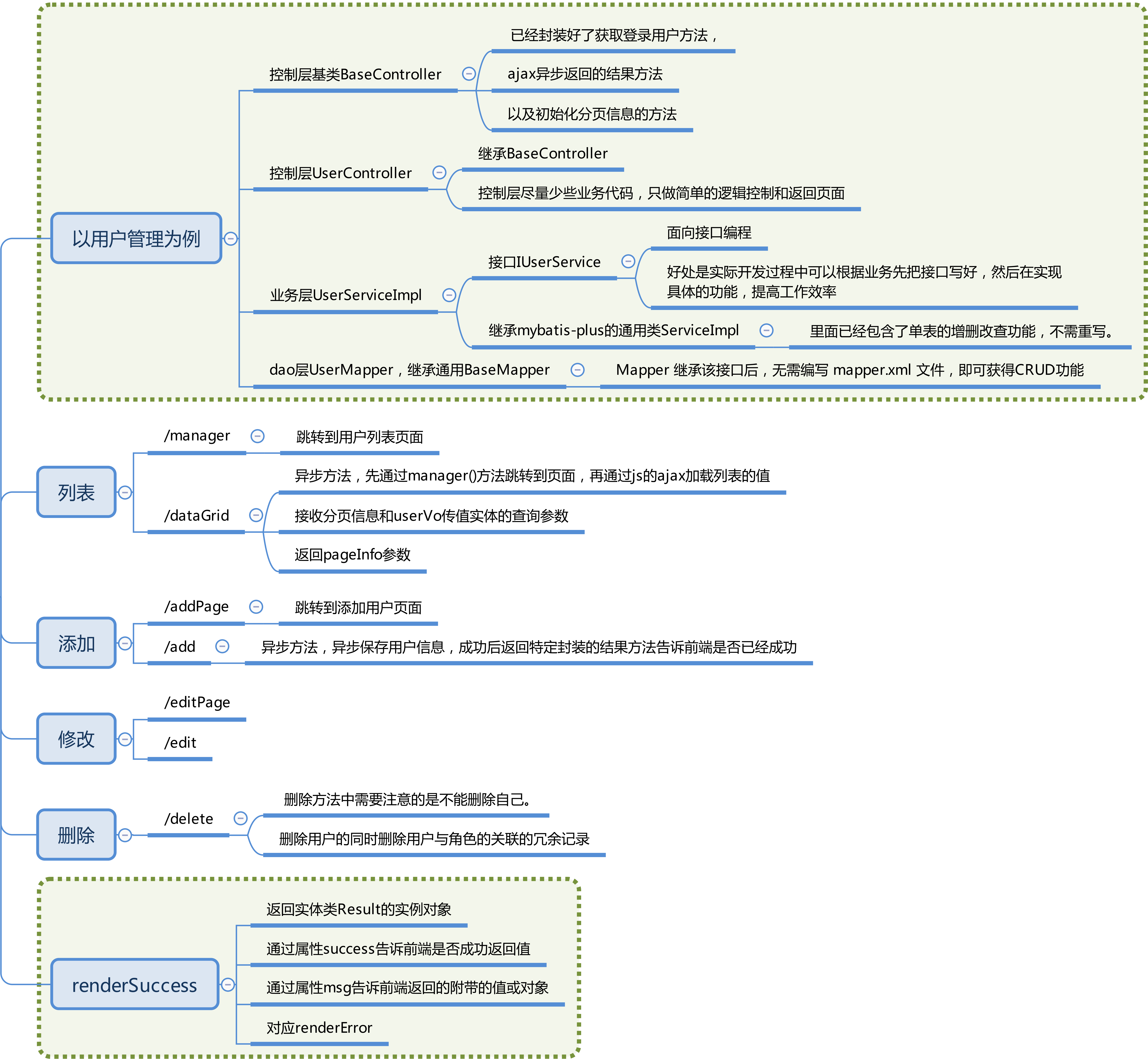
- 全局配置
- 配置是否开启二级缓存和activeRecord模式等
- 自定义文件命名
- 自定义生成的java代码包括controller、service等的命名规则
- 数据源配置
- 配置读取的数据库的信息，才能更加表生成对应的代码
- 策略配置
- 规定表名生成结构，和代码的继承关系等。
- 包目录配置
- 配置代码生成的包路径
- 注入自定义配置
- 自定义vm模板中等取到的值
- 执行生成

生成原理



😊 7、业务增删改查

公众号java-mindmap



😊 8、异常处理

公众号java-mindmap

必要性

出现异常并不可怕，可怕的是出现了异常，你却不知道，也没有进行异常处理

异常的处理是对用户友好的一个重点。用和看不懂你的500和404，你应该返回一个系统繁忙或者找不到页面的漂亮的页面给他们。

spring异常处理

实现方法

要处理异常的话，需要实现HandlerExceptionResolver这个接口类，并且重写resolveException方法，在resolveException方法中处理自己的异常逻辑。

/spring-shiro-training/src/main/java/com/wangzhixuan/commons/scan/ExceptionHandlerResolver.java

全局处理器

继承HandlerExceptionResolver，并添加注解@Component交给Spring处理

resolveException方法里面需要做的事情

- 1 使用log4j2记录异常，把异常信息记录到日志里面，方便查看日志找出问题
- 2 判断方法是否是异步请求
- 3 异步请求应该返回json数据并弹窗提示，同步请求返回到错误页面

java web项目自带

```
<error-page>
  <error-code>404</error-code>
  <location>/WEB-INF/views/error/404.jsp</location>
</error-page>
<error-page>
  <error-code>500</error-code>
  <location>/WEB-INF/views/error/500.jsp</location>
</error-page>
```

web.xml

web项目的简单异常指引，一般来说有了异常处理之后可以不配这个。

简单说说日志

System.out.println()是指输出信息到控制台，不会记录到日志文件里面

在生成环境中，一般每天会生成一个日志文件，记录系统的运行信息

😊 9、shiro权限控制

公众号java-mindmap

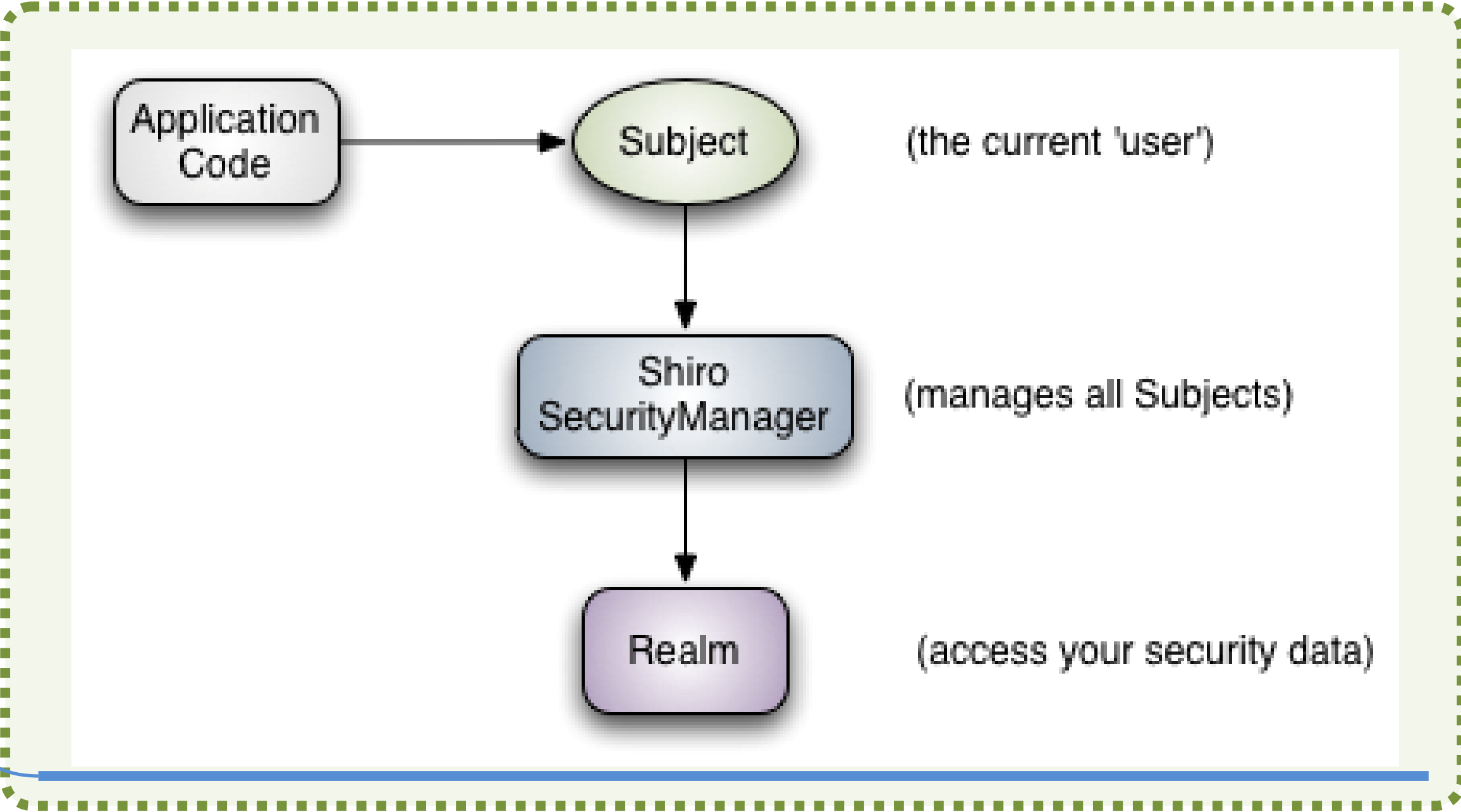
简单了解

http://...

★帮助系统完成认证、授权、加密、会话管理、与Web集成、缓存等

常用功能

- Authentication : 身份认证/登录, 验证用户是不是拥有相应的身份;
- Authorization : 授权, 即权限验证, 验证某个已认证的用户是否拥有某个权限; 即判断用户是否能做事情, 常见的如: 验证某个用户是否拥有某个角色。或者细粒度的验证某个用户对某个资源是否具有某个权限;
- Session Manager : 会话管理, 即用户登录后就是一次会话, 在没有退出之前, 它的所有信息都在会话中; 会话可以是普通JavaSE环境的, 也可以是如Web环境的;
- Caching : 缓存, 比如用户登录后, 其用户信息、拥有的角色/权限不必每次去查, 这样可以提高效率;
- Remember Me : 记住我, 这个是非常常见的功能, 即一次登录后, 下次再来的话不用登录了。



工作流程

★认识一些名词

- Subject : 主体, 代表了当前“用户”
- SecurityManager : 安全管理器; 即所有与安全有关的操作都会与SecurityManager交互; 且它管理着所有Subject
- Realm : 域, Shiro从从Realm获取安全数据 (如用户、角色、权限)
- SessionManager : 管理主体与应用之间交互的数据
- CacheManager : 缓存控制器, 来管理如用户、角色、权限等的缓存的
- Authrizer : 授权器, 或者访问控制器, 用来决定主体是否有权进行相应的操作
- Authenticator : 认证器, 负责主体认证的

回到项目的shiro使用

从配置文件spring-shiro.xml看起, 通过这个配置文件集成spring与shiro

过滤器 Shiro Filter : 是整个配置文件的入口, 用于拦截需要安全控制的请求进行处理

属性主要包括登录url、登录成功url、没权限url、不需要验证的url和安全管理器等

安全管理器 securityManager

- realm : 自定义ShiroDbRealm
- 认证步骤 :
 - 重写doGetAuthenticationInfo()方法 : 主要验证登录密码是否正确, 已经获取用户的的权限和角色, 交给shiro
 - 1、用户提交用户名和密码
 - 2、shiro 封装令牌 (token,包含username和password信息)
 - 3、realm 通过用户名将密码查询返回
 - 4、shiro 自动去比较查询出密码和用户输入密码是否一致
 - 5、进行登陆控制
- matcher : RetryLimitCredentialsMatcher : 密码错误5次锁定
- cacheManager : 集成spring-cache作为shiro的缓存, 项目中的redis和ehcache用其中一个
- rememberMeManager : 记得登录状态管理器
- sessionManager : 会话管理器 : 主要配置登录超时的session管理

注意一点 : Shiro不会去维护用户、维护权限; 这些需要我们去设计/提供; 然后通过相应的接口注入给Shiro即可

😊 10、缓存机制

公众号java-mindmap

缓存是介于应用程序和物理数据源之间，其作用是为了降低应用程序对物理数据源访问的频次，从而提高了应用的运行性能。

简介

★ 该项目提供了2中缓存机制，redis和ehcache

spring-cache

它本质上不是一个具体的缓存实现方案，而是一个对缓存使用的抽象，

通过在既有代码中添加少量它定义的各种 annotation，即能够达到缓存方法的返回对象的效果

特点

能够使用 SpEL (Spring Expression Language) 来定义缓存的 key 和各种 condition

支持和主流的专业缓存例如 EHCACHE、redis集成

注解形式

/spring-shiro-training/src/main/java/com/wangzhixuan/service/impl/TestService.java

@Cacheable缓存方法返回的值

编码形式

/spring-shiro-training/src/main/java/com/wangzhixuan/task/TestTask.java

Cache cache = cacheManager.getCache("hour");

ehcache

配置文件ehcache.xml，配置缓存时间

流行的纯Java开源缓存框架，配置简单、结构清晰、功能强大

直接在jvm虚拟机中缓存，速度快，效率高

但是缓存共享麻烦，集群分布式应用不方便

redis

通过socket访问到缓存服务

效率比ecache低，比数据库要快很多

处理集群和分布式缓存方便，有成熟的方案。

spring-cache集成缓存

切换两种缓存

```
<import resource="classpath:spring/spring-ehcache.xml"/>
<!--
<import resource="classpath:spring/spring-redis.xml"/>
-->
```

只需要注释掉其中一个即可

★ 以后需要用到缓存，直接复制一个用即可，很方便

如果是单个应用或者对缓存访问要求很高的应用，用ehcache。

使用建议

如果是大型系统，存在缓存共享、分布式部署、缓存内容很大的，建议用redis。

http://bl...

😊 11、Druid数据源监控

📖 公众号java-mindmap

一个JDBC组件，它包括三部分

📖 <http://www.oschina.net/p/druid>

DruidDriver 代理Driver，能够提供基于Filter - Chain模式的插件体系。

DruidDataSource 高效可管理的数据库连接池。

SQLParser

可以做什么？

1 可以监控数据库访问性能，Druid内置提供了一个功能强大的StatFilter插件，能够详细统计SQL的执行性能，这对于线上分析数据库访问性能有帮助

2 替换DBCP和C3P0。Druid提供了一个高效、功能强大、可扩展性好的数据库连接池。

3 SQL执行日志，Druid提供了不同的LogFilter，能够支持Common-Logging、Log4j和JdkLog，你可以按需要选择相应的LogFilter，监控你应用的数据库访问情况。

4 扩展JDBC，如果你要对JDBC层有编程的需求，可以通过Druid提供的Filter-Chain机制，很方便编写JDBC层的扩展插件。

简单使用

1 spring-mybatis.xml中使用Druid连接池作为数据源

2 在web.xml中配置Druid过滤器

3 链接地址：http://localhost:8080/druid/index.html

看一看

N	SQL	执行数	执行时间	最慢	事务中	错误数	更新行数	读取行数	执行中	最大并发	执行时间分布	执行+RS时分布	读取行分布	更新行分布
1	select role_id AS roleid ...	1	7	7	1			1		1	[0,1,0,0,0,0,0]	[1,0,0,0,0,0,0]	[0,1,0,0,0]	[2,0,0,0,0,0]
2	SELECT id,login_name AS ...	1	6	6	1					1	[0,1,0,0,0,0,0]	[1,0,0,0,0,0,0]	[1,0,0,0,0]	[2,0,0,0,0,0]
3	SELECT DISTINCT s.id AS i...	1	103	103	1			12		1	[0,0,0,1,0,0,0]	[1,0,0,0,0,0,0]	[0,0,1,0,0]	[2,0,0,0,0,0]
4	SELECT COUNT(?) FROM (SEL...	1	25	25				1		1	[0,0,1,0,0,0,0]	[1,0,0,0,0,0,0]	[0,1,0,0,0]	[1,0,0,0,0,0]