|  |
| --- |
| [公司名称] |
| 标题 |
| 副标题 |

|  |
| --- |
| 姚斯元  2018-10-30 |

目录

[一、 123 1](#_Toc3539499)

[1.1 1 1](#_Toc3539500)

[1.1.1 3 1](#_Toc3539501)

# 环境搭建

## 需要的jar包

shiro需要的jar包就一个，我这里用的是shiro-all-1.3.2.jar

## 在web.xml中加入filter

|  |
| --- |
| <!-- shiro过滤器 -->  <!-- 这个filter要写在所有filter的最前面，保证他是过滤器中第一个起作用的-->  <filter>  <filter-name>shiroFilter</filter-name>  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>  <init-param>  <param-name>targetFilterLifecycle</param-name>  <param-value>true</param-value>  </init-param>  </filter>  <!-- 缺省为false，表示由SpringApplicationContext管理生命周期，置为true则表示由ServletContainer管理 -->  <filter-mapping>  <filter-name>shiroFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>  <!-- Spring和mybatis的配置文件 -->  <context-param>  <param-name>contextConfigLocation</param-name>  <param-value>  classpath:spring/spring-mybatis.xml  classpath:spring/spring-shiro.xml  </param-value>  </context-param> |

## shiro.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="http://www.springframework.org/schema/context"  xmlns:mvc="http://www.springframework.org/schema/mvc"  xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd  http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd  http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd">  <!-- 配置shiro -->  <bean id="securityManager" class="org.apache.shiro.web.mgt.DefaultWebSecurityManager">  <!-- 指定Shiro验证用户登录的类为自定义的Realm（若有多个Realm，可用[realms]属性代替） -->  <property name="realm">  <bean class="com.xx.shiro.MyRealm"/>  </property>  </bean>  <!-- Shiro Filter-->  <bean id="simplePermFilter" class="org.apache.shiro.web.filter.authz.PermissionsAuthorizationFilter"></bean>  <bean id="shiroFilter" class="org.apache.shiro.spring.web.ShiroFilterFactoryBean ">  <!-- Shiro的核心安全接口，这个属性是必须的 -->  <property name="securityManager" ref="securityManager"/>  <!-- 要求登录时的链接（可根据项目的URL进行替换），非必须的属性，默认会找Web工程根目录下的[/login.jsp] -->  <property name="loginUrl" value="/sys/toLogin"/>  <!-- 登录成功后要跳转的连接（本例中此属性用不到，因为登录成功后的处理逻辑已在LoginController中硬编码为main.jsp） -->  <property name="successUrl" value="/sys/login"/>  <!--用户访问未授权的资源时，所显示的连接 -->  <property name="unauthorizedUrl" value="/sys/toLogin"/>  <!-- 权限配置 -->  <property name="filters">  <map>  <entry key="roles" value-ref="simplePermFilter"/>  </map>  </property>  <!--  anon：它对应的过滤器里面是空的，什么都没做，另外.do和.jsp后面的\*表示参数，比方说[login.jsp?main]这种  authc：该过滤器下的页面必须验证后才能访问，它是内置的org.apache.shiro.web.filter.authc.FormAuthenticationFilter  注意：对于相似的资源，需要将anon的设置放在authc前面，anon才会生效，因为Shiro是从上往下匹配URL的，匹配成功便不再匹配了  -->  <property name="filterChainDefinitions">  <value>  /sys/toLogin = anon  /sys/Login = anon  /sys/videoList = authc,rolse[管理员]  </value>  </property>  </bean>  <!-- 保证实现了Shiro内部lifecycle函数的bean执行 -->  <bean id="lifecycleBeanPostProcessor" class="org.apache.shiro.spring.LifecycleBeanPostProcessor"/>  <!--自定义拦截器，目前用不到，后面再说可以不写-->  <!--<bean id="anyRoles" class="com.xx.shiro.CustomRolesAuthorizationFilter" /> -->  </beans> |

## ehcache

1、加入依赖

<dependency>

<groupId>org.apache.shiro</groupId>

<artifactId>shiro-ehcache</artifactId>

<version>${shiro.version}</version>

<exclusions>

<exclusion>

<artifactId>slf4j-api</artifactId>

<groupId>org.slf4j</groupId>

</exclusion>

</exclusions>

</dependency>

2、在项目的resource目录下新建立：ehcache-shiro.xml

<?xml version="1.0" encoding="UTF-8"?>

<ehcache updateCheck="false" name="shiroCache">

<!-- <diskStore path="C:\shiro\ehcache" /> -->

<diskStore path="java.io.tmpdir"/>

<!--

eternal：缓存中对象是否为永久的，如果是，超时设置将被忽略，对象从不过期。

maxElementsInMemory：缓存中允许创建的最大对象数

overflowToDisk：内存不足时，是否启用磁盘缓存。

timeToIdleSeconds：缓存数据的钝化时间，也就是在一个元素消亡之前， 两次访问时间的最大时间间隔值，这只能在元素不是永久驻留时有效，如果该值是 0 就意味着元素可以停顿无穷长的时间。

timeToLiveSeconds：缓存数据的生存时间，也就是一个元素从构建到消亡的最大时间间隔值，这只能在元素不是永久驻留时有效，如果该值是0就意味着元素可以停顿无穷长的时间。

memoryStoreEvictionPolicy：缓存满了之后的淘汰算法。

diskPersistent:设定在虚拟机重启时是否进行磁盘存储，默认为false

diskExpiryThreadIntervalSeconds: 属性可以设置该线程执行的间隔时间(默认是120秒，不能太小

1 FIFO，先进先出

2 LFU，最少被使用，缓存的元素有一个hit属性，hit值最小的将会被清出缓存。

3 LRU，最近最少使用的，缓存的元素有一个时间戳，当缓存容量满了，而又需要腾出地方来缓存新的元素的时候，那么现有缓存元素中时间戳离当前时间最远的元素将被清出缓存。

-->

<defaultCache

maxElementsInMemory="10000"

eternal="false"

timeToIdleSeconds="120"

timeToLiveSeconds="120"

overflowToDisk="false"

diskPersistent="false"

diskExpiryThreadIntervalSeconds="120"

/>

<cache name="activeSessionCache"

maxElementsInMemory="10000"

eternal="true"

overflowToDisk="false"

diskPersistent="true"

diskExpiryThreadIntervalSeconds="600"/>

<cache name="shiro.authorizationCache"

maxElementsInMemory="100"

eternal="false"

timeToLiveSeconds="600"

overflowToDisk="false"/>

</ehcache>

3、在shiro的配置文件中 securityManager 的bean中 增加 cacheManager 属性配置：如 下 文件中 标红的 属性

1 <!-- 1、安全管理器 -->

2 <bean id="securityManager" class="org.apache.shiro.web.mgt.DefaultWebSecurityManager">

3 <property name="realm" ref="shiroDbRealm"></property>

4 <!-- 设置缓存管理器为 ehcache -->

5 <property name="cacheManager" ref="shiroEhcacheManager"></property>

6 </bean>

4、增加Ehcache的配置：如

1 <!-- 用户授权信息Cache, 采用EhCache -->

2 <bean id="shiroEhcacheManager" class="org.apache.shiro.cache.ehcache.EhCacheManager">

3 <property name="cacheManagerConfigFile" value="classpath:shiro/ehcache-shiro.xml"></property>

4 </bean>

3、在shiro的配置文件中 securityManager 的bean中 增加 cacheManager 属性配置：如 下 文件中 标红的 属性

第二：怎么清空系统的缓存

1、编写自定义的realm：如： ShiroDBRealm

2、注解 ShiroDBRealm 为一个 @Component 组件

3、编写方法清理缓存：

/\*\*

\*

\* @Description: 权限修改生效后，立即刷新清空缓存，则可以实现用户不退出生效新的权限

\*

\* @author admin

\* @date 2016年9月29日 下午9:34:07

\*/

public void clearCache() {

PrincipalCollection principals = SecurityUtils.getSubject().getPrincipals();

super.clearCache(principals);

}

# 动态权限控制

### sql

|  |
| --- |
| drop database ssm\_shiro;  # 创建数据库 ssm\_shiro  CREATE DATABASE ssm\_shiro DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci;  # 使用数据库 ssm\_shiro  USE ssm\_shiro;  drop table if EXISTS t\_user;  drop table if EXISTS t\_role;  drop table if EXISTS t\_user\_role;  drop table if EXISTS t\_resource;  drop table if EXISTS t\_role\_resource;  # 创建数据表 t\_user  CREATE TABLE t\_user(  id TINYINT PRIMARY KEY AUTO\_INCREMENT comment '用户 ID',  username VARCHAR(30) NOT NULL comment '用户名',  `password` VARCHAR(32) NOT NULL comment '密码',  nickname VARCHAR(30) NOT NULL comment '昵称',  `status` TINYINT not null comment '状态:1 启用,2 禁用'  )ENGINE=INNODB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8 COMMENT='用户信息表';  # 创建数据用于测试  INSERT INTO t\_user(username,`password`,nickname,`status`)  VALUES('admin','a66abb5684c45962d887564f08346e8d','超级管理员',1);  INSERT INTO t\_user(username,`password`,nickname,`status`)  VALUES('dev','c43812121e594f158520698ba706118f','开发工程师',1);  INSERT INTO t\_user(username,`password`,nickname,`status`)  VALUES('test','47ec2dd791e31e2ef2076caf64ed9b3d','测试工程师',1);  INSERT INTO t\_user(username,`password`,nickname,`status`)  VALUES('doc','5afd1e481507a2a181decc3860b32d15','文档工程师',1);  # 创建数据表 t\_role  # name 字段用于显示给人看, sn 字段用在代码中做角色匹配  CREATE TABLE t\_role(  id TINYINT PRIMARY KEY AUTO\_INCREMENT comment '角色表 ID',  `name` VARCHAR(20) NOT NULL comment '角色名称',  sn VARCHAR(20) NOT NULL comment '角色字符串'  )engine=innodb auto\_increment=1 DEFAULT charset=utf8 comment='角色信息表';  # 创建数据用于测试  INSERT INTO t\_role(`name`,`sn`) VALUES('管理员','admin'),('开发工程师','dev'),('测试工程师','test'),('文档工程师','doc');  # 创建数据表 t\_user\_role  CREATE TABLE t\_user\_role(  id TINYINT PRIMARY KEY AUTO\_INCREMENT comment '用户角色关联表 ID',  user\_id TINYINT NOT NULL,  role\_id TINYINT NOT NULL  )engine=innodb auto\_increment=1 charset=utf8 comment='用户角色关联表';  # 创建数据用于测试  INSERT INTO `t\_user\_role`(`user\_id`,`role\_id`)  VALUES(1,1),(2,2),(3,3),(4,4);  # 创建资源表 t\_resource  # 资源在本项目中的含义就是 "权限"  CREATE TABLE t\_resource(  id TINYINT PRIMARY KEY AUTO\_INCREMENT comment '资源 ID',  `name` VARCHAR(20) NOT NULL comment '资源名称,一般是中文名称(给人看的)',  permission VARCHAR(40) NOT NULL comment '资源权限字符串,一般是 Shiro 默认的通配符表示(给人看的)',  url VARCHAR(40) NOT NULL comment '资源 url 表示,我们设计的系统让 Shiro 通过这个路径字符串去匹配浏览器中显示的路径'  )engine=innodb auto\_increment=1 charset=utf8 comment='资源表';  # 创建数据用于测试  INSERT INTO t\_resource(`name`,permission,url)  VALUES('系统管理','admin:\*','/admin/\*\*'),  ('用户管理','user:\*','/admin/user/\*\*'),  ('用户添加','user:add','/admin/user/add'),  ('用户删除','user:delete','/admin/user/delete'),  ('用户修改','user:update','/admin/user/update'),  ('用户查询','user:list','/admin/user/list'),  ('用户资源查询','user:resources:\*','/admin/user/resources/\*'),  ('角色管理','role:\*','/admin/role/\*\*'),  ('角色添加','role:add','/admin/role/add'),  ('角色删除','role:delete','/admin/role/delete'),  ('角色修改','role:update','/admin/role/update'),  ('角色查询','role:list','/admin/role/list'),  ('角色资源查询','role:resources:\*','/admin/role/resources/\*'),  ('资源管理','resource:\*','/admin/resource/\*\*'),  ('资源增加','resource:add','/admin/resource/add'),  ('资源删除','resource:delete','/admin/resource/delete'),  ('资源修改','resource:update','/admin/resource/update'),  ('资源查询','resource:list','/admin/resource/list');  # 创建角色资源关联表  CREATE TABLE t\_role\_resource(  id TINYINT PRIMARY KEY AUTO\_INCREMENT comment '角色资源关联 ID',  role\_id TINYINT not null comment '角色 id',  resource\_id TINYINT not null comment '资源 id'  )engine=innodb auto\_increment=1 charset=utf8 comment='角色资源关联表';  # 创建数据用于测试  INSERT INTO t\_role\_resource(role\_id,resource\_id)  VALUES(1,1),  (2,3),(2,5),(2,6),(2,7),(2,9),(2,11),(2,12),(2,13),(2,15),(2,17),(2,18),  (3,6),(3,7),(3,8),(3,14),  (4,6),(4,7),(4,12),(4,18); |

# 常见问题

[org.springframework.web.context.ContextLoader] - Context initialization failed

org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'shiroFilter' defined in file [C:\git\zydh\target\MavenDemo03\WEB-INF\classes\spring\spring-shiro.xml]: Cannot resolve reference to bean 'securityManager' while setting bean property 'securityManager'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'securityManager' defined in file [C:\git\zydh\target\MavenDemo03\WEB-INF\classes\spring\spring-shiro.xml]: Error setting property values; nested exception is org.springframework.beans.PropertyBatchUpdateException; nested PropertyAccessExceptions (1) are:

PropertyAccessException 1: org.springframework.beans.MethodInvocationException: Property 'cacheManager' threw exception; nested exception is org.apache.shiro.cache.CacheException: net.sf.ehcache.CacheException: Error configuring from input stream. Initial cause was 文档中根元素后面的标记必须格式正确。

四月 26, 2019 10:32:38 下午 org.apache.catalina.core.StandardContext startInternal

严重: Error listenerStart

at org.springframework.beans.factory.support.BeanDefinitionValueResolver.resolveReference(BeanDefinitionValueResolver.java:328)

at org.springframework.beans.factory.support.BeanDefinitionValueResolver.resolveValueIfNecessary(BeanDefinitionValueResolver.java:107)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.applyPropertyValues(AbstractAutowireCapableBeanFactory.java:1456)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:1197)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:537)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:475)

at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:304)

at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:228)

at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:300)

at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:200)

at org.springframework.context.support.PostProcessorRegistrationDelegate.registerBeanPostProcessors(PostProcessorRegistrationDelegate.java:232)

at org.springframework.context.support.AbstractApplicationContext.registerBeanPostProcessors(AbstractApplicationContext.java:618)

at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:467)

at org.springframework.web.context.ContextLoader.configureAndRefreshWebApplicationContext(ContextLoader.java:403)

at org.springframework.web.context.ContextLoader.initWebApplicationContext(ContextLoader.java:306)

at org.springframework.web.context.ContextLoaderListener.contextInitialized(ContextLoaderListener.java:106)

at org.apache.catalina.core.StandardContext.listenerStart(StandardContext.java:4973)

at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5467)

at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)

at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:901)

at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:877)

at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:632)

at org.apache.catalina.startup.HostConfig.manageApp(HostConfig.java:1740)

四月 26, 2019 10:32:39 下午 org.apache.catalina.core.StandardContext startInternal

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

严重: Context [] startup failed due to previous errors

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:498)

at org.apache.tomcat.util.modeler.BaseModelMBean.invoke(BaseModelMBean.java:301)

at com.sun.jmx.interceptor.DefaultMBeanServerInterceptor.invoke(DefaultMBeanServerInterceptor.java:819)

at com.sun.jmx.mbeanserver.JmxMBeanServer.invoke(JmxMBeanServer.java:801)

at org.apache.catalina.mbeans.MBeanFactory.createStandardContext(MBeanFactory.java:618)

at org.apache.catalina.mbeans.MBeanFactory.createStandardContext(MBeanFactory.java:565)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:498)

at org.apache.tomcat.util.modeler.BaseModelMBean.invoke(BaseModelMBean.java:301)

at com.sun.jmx.interceptor.DefaultMBeanServerInterceptor.invoke(DefaultMBeanServerInterceptor.java:819)

at com.sun.jmx.mbeanserver.JmxMBeanServer.invoke(JmxMBeanServer.java:801)

at javax.management.remote.rmi.RMIConnectionImpl.doOperation(RMIConnectionImpl.java:1468)

at javax.management.remote.rmi.RMIConnectionImpl.access$300(RMIConnectionImpl.java:76)

at javax.management.remote.rmi.RMIConnectionImpl$PrivilegedOperation.run(RMIConnectionImpl.java:1309)

at javax.management.remote.rmi.RMIConnectionImpl.doPrivilegedOperation(RMIConnectionImpl.java:1401)

at javax.management.remote.rmi.RMIConnectionImpl.invoke(RMIConnectionImpl.java:829)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:498)

at sun.rmi.server.UnicastServerRef.dispatch(UnicastServerRef.java:357)

at sun.rmi.transport.Transport$1.run(Transport.java:200)

at sun.rmi.transport.Transport$1.run(Transport.java:197)

at java.security.AccessController.doPrivileged(Native Method)

at sun.rmi.transport.Transport.serviceCall(Transport.java:196)

at sun.rmi.transport.tcp.TCPTransport.handleMessages(TCPTransport.java:568)

at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run0(TCPTransport.java:826)

at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.lambda$run$0(TCPTransport.java:683)

at java.security.AccessController.doPrivileged(Native Method)

at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run(TCPTransport.java:682)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)

at java.lang.Thread.run(Thread.java:748)

Caused by: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'securityManager' defined in file [C:\git\zydh\target\MavenDemo03\WEB-INF\classes\spring\spring-shiro.xml]: Error setting property values; nested exception is org.springframework.beans.PropertyBatchUpdateException; nested PropertyAccessExceptions (1) are:

PropertyAccessException 1: org.springframework.beans.MethodInvocationException: Property 'cacheManager' threw exception; nested exception is org.apache.shiro.cache.CacheException: net.sf.ehcache.CacheException: Error configuring from input stream. Initial cause was 文档中根元素后面的标记必须格式正确。

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.applyPropertyValues(AbstractAutowireCapableBeanFactory.java:1493)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:1197)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:537)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:475)

at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:304)

at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:228)

at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:300)

at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:195)

[2019-04-26 10:32:39,073] Artifact zydh:war exploded: Error during artifact deployment. See server log for details.

at org.springframework.beans.factory.support.BeanDefinitionValueResolver.resolveReference(BeanDefinitionValueResolver.java:320)

... 60 more

Caused by: org.springframework.beans.PropertyBatchUpdateException; nested PropertyAccessExceptions (1) are:

PropertyAccessException 1: org.springframework.beans.MethodInvocationException: Property 'cacheManager' threw exception; nested exception is org.apache.shiro.cache.CacheException: net.sf.ehcache.CacheException: Error configuring from input stream. Initial cause was 文档中根元素后面的标记必须格式正确。

at org.springframework.beans.AbstractPropertyAccessor.setPropertyValues(AbstractPropertyAccessor.java:108)

at org.springframework.beans.AbstractPropertyAccessor.setPropertyValues(AbstractPropertyAccessor.java:62)

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.applyPropertyValues(AbstractAutowireCapableBeanFactory.java:1489)

... 68 more

错误的xml文件

<?xml version="1.0" encoding="UTF-8"?>

<ehcache name="es">

<diskStore path="java.io.tmpdir"/>

<!-- 密码输入错误 锁定1小时 -->

<!-- timeToIdleSeconds:设置对象在失效前的允许闲置时间（单位：秒） -->

<!-- timeToLiveSeconds:设置对象在失效前允许存活时间（单位：秒）。最大时间介于创建时间和失效时间之间。仅当eternal=false对象不是永久有效时使用，默认是0.，也就是对象存活时间无穷大。 -->

<cache name="passwordRetryCache"

maxEntriesLocalHeap="2000"

eternal="false"

timeToIdleSeconds="3600"

timeToLiveSeconds="0"

overflowToDisk="false"

statistics="true">

</cache>

<!-- 权限记录缓存 锁定1小时 -->

<cache name="authorizationCache"

maxEntriesLocalHeap="2000"

eternal="false"

timeToIdleSeconds="3600"

timeToLiveSeconds="0"

overflowToDisk="false"

statistics="true">

</cache>

<!-- 登录认证记录缓存 锁定10分钟 -->

<cache name="authenticationCache"

maxEntriesLocalHeap="2000"

eternal="false"

timeToIdleSeconds="3600"

timeToLiveSeconds="0"

overflowToDisk="false"

statistics="true">

</cache>

<!-- 会话次数缓存 -->

<cache name="shiro-activeSessionCache"

maxEntriesLocalHeap="10000"

overflowToDisk="false"

eternal="false"

diskPersistent="false"

timeToLiveSeconds="0"

timeToIdleSeconds="0"

statistics="true"/>

</ehcache>

<ehcache xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="http://ehcache.xsd">

<!--diskStore：缓存数据持久化的目录 地址 -->

<diskStore path="E:\copyrightPlatformehCache" />

<defaultCache maxElementsInMemory="1000"

maxElementsOnDisk="10000000" eternal="false" overflowToDisk="false"

diskPersistent="false" timeToIdleSeconds="120" timeToLiveSeconds="120"

diskExpiryThreadIntervalSeconds="120" memoryStoreEvictionPolicy="LRU">

</defaultCache>

</ehcache>

更改后能正常运行的文件

<?xml version="1.0" encoding="UTF-8"?>

<ehcache updateCheck="false" name="shiroCache">

<diskStore path="C:\shiro\ehcache" />

<!-- <diskStore path="java.io.tmpdir"/> -->

<!--

eternal：缓存中对象是否为永久的，如果是，超时设置将被忽略，对象从不过期。

maxElementsInMemory：缓存中允许创建的最大对象数

overflowToDisk：内存不足时，是否启用磁盘缓存。

timeToIdleSeconds：缓存数据的钝化时间，也就是在一个元素消亡之前， 两次访问时间的最大时间间隔值，这只能在元素不是永久驻留时有效，如果该值是 0 就意味着元素可以停顿无穷长的时间。

timeToLiveSeconds：缓存数据的生存时间，也就是一个元素从构建到消亡的最大时间间隔值，这只能在元素不是永久驻留时有效，如果该值是0就意味着元素可以停顿无穷长的时间。

memoryStoreEvictionPolicy：缓存满了之后的淘汰算法。

diskPersistent:设定在虚拟机重启时是否进行磁盘存储，默认为false

diskExpiryThreadIntervalSeconds: 属性可以设置该线程执行的间隔时间(默认是120秒，不能太小

1 FIFO，先进先出

2 LFU，最少被使用，缓存的元素有一个hit属性，hit值最小的将会被清出缓存。

3 LRU，最近最少使用的，缓存的元素有一个时间戳，当缓存容量满了，而又需要腾出地方来缓存新的元素的时候，那么现有缓存元素中时间戳离当前时间最远的元素将被清出缓存。

-->

<defaultCache

maxElementsInMemory="10000"

eternal="false"

timeToIdleSeconds="120"

timeToLiveSeconds="120"

overflowToDisk="false"

diskPersistent="false"

diskExpiryThreadIntervalSeconds="120"

/>

<cache name="activeSessionCache"

maxElementsInMemory="10000"

eternal="true"

overflowToDisk="false"

diskPersistent="true"

diskExpiryThreadIntervalSeconds="600"/>

<cache name="shiro.authorizationCache"

maxElementsInMemory="100"

eternal="false"

timeToLiveSeconds="600"

overflowToDisk="false"/>

</ehcache>

## 异常信息

```

信息: Initializing Spring root WebApplicationContext

四月 26, 2019 11:45:46 下午 org.apache.catalina.core.StandardContext listenerStart

严重: Exception sending context initialized event to listener instance of class org.springframework.web.context.ContextLoaderListener

org.springframework.beans.factory.xml.XmlBeanDefinitionStoreException: Line 2 in XML document from file [C:\git\zydh\target\MavenDemo03\WEB-INF\classes\spring\spring-shiro.xml] is invalid; nested exception is org.xml.sax.SAXParseException; lineNumber: 2; columnNumber: 48; cvc-elt.1: 找不到元素 'ehcache' 的声明。

at org.springframework.beans.factory.xml.XmlBeanDefinitionReader.doLoadBeanDefinitions(XmlBeanDefinitionReader.java:398)

at org.springframework.beans.factory.xml.XmlBeanDefinitionReader.loadBeanDefinitions(XmlBeanDefinitionReader.java:335)

at org.springframework.beans.factory.xml.XmlBeanDefinitionReader.loadBeanDefinitions(XmlBeanDefinitionReader.java:303)

at org.springframework.beans.factory.support.AbstractBeanDefinitionReader.loadBeanDefinitions(AbstractBeanDefinitionReader.java:180)

at org.springframework.beans.factory.support.AbstractBeanDefinitionReader.loadBeanDefinitions(AbstractBeanDefinitionReader.java:216)

at org.springframework.beans.factory.support.AbstractBeanDefinitionReader.loadBeanDefinitions(AbstractBeanDefinitionReader.java:187)

at org.springframework.web.context.support.XmlWebApplicationContext.loadBeanDefinitions(XmlWebApplicationContext.java:125)

at org.springframework.web.context.support.XmlWebApplicationContext.loadBeanDefinitions(XmlWebApplicationContext.java:94)

at org.springframework.context.support.AbstractRefreshableApplicationContext.refreshBeanFactory(AbstractRefreshableApplicationContext.java:129)

at org.springframework.context.support.AbstractApplicationContext.obtainFreshBeanFactory(AbstractApplicationContext.java:540)

at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:454)

at org.springframework.web.context.ContextLoader.configureAndRefreshWebApplicationContext(ContextLoader.java:403)

at org.springframework.web.context.ContextLoader.initWebApplicationContext(ContextLoader.java:306)

at org.springframework.web.context.ContextLoaderListener.contextInitialized(ContextLoaderListener.java:106)

at org.apache.catalina.core.StandardContext.listenerStart(StandardContext.java:4973)

at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5467)

at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)

at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:901)

at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:877)

at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:632)

at org.apache.catalina.startup.HostConfig.manageApp(HostConfig.java:1740)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:498)

at org.apache.tomcat.util.modeler.BaseModelMBean.invoke(BaseModelMBean.java:301)

at com.sun.jmx.interceptor.DefaultMBeanServerInterceptor.invoke(DefaultMBeanServerInterceptor.java:819)

at com.sun.jmx.mbeanserver.JmxMBeanServer.invoke(JmxMBeanServer.java:801)

at org.apache.catalina.mbeans.MBeanFactory.createStandardContext(MBeanFactory.java:618)

at org.apache.catalina.mbeans.MBeanFactory.createStandardContext(MBeanFactory.java:565)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:498)

at org.apache.tomcat.util.modeler.BaseModelMBean.invoke(BaseModelMBean.java:301)

at com.sun.jmx.interceptor.DefaultMBeanServerInterceptor.invoke(DefaultMBeanServerInterceptor.java:819)

at com.sun.jmx.mbeanserver.JmxMBeanServer.invoke(JmxMBeanServer.java:801)

at javax.management.remote.rmi.RMIConnectionImpl.doOperation(RMIConnectionImpl.java:1468)

at javax.management.remote.rmi.RMIConnectionImpl.access$300(RMIConnectionImpl.java:76)

at javax.management.remote.rmi.RMIConnectionImpl$PrivilegedOperation.run(RMIConnectionImpl.java:1309)

at javax.management.remote.rmi.RMIConnectionImpl.doPrivilegedOperation(RMIConnectionImpl.java:1401)

at javax.management.remote.rmi.RMIConnectionImpl.invoke(RMIConnectionImpl.java:829)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:498)

at sun.rmi.server.UnicastServerRef.dispatch(UnicastServerRef.java:357)

at sun.rmi.transport.Transport$1.run(Transport.java:200)

at sun.rmi.transport.Transport$1.run(Transport.java:197)

at java.security.AccessController.doPrivileged(Native Method)

at sun.rmi.transport.Transport.serviceCall(Transport.java:196)

at sun.rmi.transport.tcp.TCPTransport.handleMessages(TCPTransport.java:568)

at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run0(TCPTransport.java:826)

at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.lambda$run$0(TCPTransport.java:683)

at java.security.AccessController.doPrivileged(Native Method)

at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run(TCPTransport.java:682)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)

at java.lang.Thread.run(Thread.java:748)

Caused by: org.xml.sax.SAXParseException; lineNumber: 2; columnNumber: 48; cvc-elt.1: 找不到元素 'ehcache' 的声明。

at com.sun.org.apache.xerces.internal.util.ErrorHandlerWrapper.createSAXParseException(ErrorHandlerWrapper.java:203)

at com.sun.org.apache.xerces.internal.util.ErrorHandlerWrapper.error(ErrorHandlerWrapper.java:134)

at com.sun.org.apache.xerces.internal.impl.XMLErrorReporter.reportError(XMLErrorReporter.java:396)

at com.sun.org.apache.xerces.internal.impl.XMLErrorReporter.reportError(XMLErrorReporter.java:327)

at com.sun.org.apache.xerces.internal.impl.XMLErrorReporter.reportError(XMLErrorReporter.java:284)

at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator.handleStartElement(XMLSchemaValidator.java:1900)

at com.sun.org.apache.xerces.internal.impl.xs.XMLSchemaValidator.startElement(XMLSchemaValidator.java:740)

at com.sun.org.apache.xerces.internal.impl.XMLNSDocumentScannerImpl.scanStartElement(XMLNSDocumentScannerImpl.java:374)

at com.sun.org.apache.xerces.internal.impl.XMLNSDocumentScannerImpl$NSContentDriver.scanRootElementHook(XMLNSDocumentScannerImpl.java:613)

at com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl$FragmentContentDriver.next(XMLDocumentFragmentScannerImpl.java:3132)

at com.sun.org.apache.xerces.internal.impl.XMLDocumentScannerImpl$PrologDriver.next(XMLDocumentScannerImpl.java:852)

at com.sun.org.apache.xerces.internal.impl.XMLDocumentScannerImpl.next(XMLDocumentScannerImpl.java:602)

at com.sun.org.apache.xerces.internal.impl.XMLNSDocumentScannerImpl.next(XMLNSDocumentScannerImpl.java:112)

at com.sun.org.apache.xerces.internal.impl.XMLDocumentFragmentScannerImpl.scanDocument(XMLDocumentFragmentScannerImpl.java:505)

at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(XML11Configuration.java:841)

at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(XML11Configuration.java:770)

at com.sun.org.apache.xerces.internal.parsers.XMLParser.parse(XMLParser.java:141)

at com.sun.org.apache.xerces.internal.parsers.DOMParser.parse(DOMParser.java:243)

at com.sun.org.apache.xerces.internal.jaxp.DocumentBuilderImpl.parse(DocumentBuilderImpl.java:339)

at org.springframework.beans.factory.xml.DefaultDocumentLoader.loadDocument(DefaultDocumentLoader.java:76)

at org.springframework.beans.factory.xml.XmlBeanDefinitionReader.doLoadDocument(XmlBeanDefinitionReader.java:428)

at org.springframework.beans.factory.xml.XmlBeanDefinitionReader.doLoadBeanDefinitions(XmlBeanDefinitionReader.java:390)

... 58 more

四月 26, 2019 11:45:46 下午 org.apache.catalina.core.ApplicationContext log

信息: Closing Spring root WebApplicationContext

四月 26, 2019 11:45:46 下午 org.apache.catalina.core.StandardContext listenerStop

严重: Exception sending context destroyed event to listener instance of class org.springframework.web.context.ContextLoaderListener

java.lang.IllegalStateException: BeanFactory not initialized or already closed - call 'refresh' before accessing beans via the ApplicationContext

at org.springframework.context.support.AbstractRefreshableApplicationContext.getBeanFactory(AbstractRefreshableApplicationContext.java:170)

at org.springframework.context.support.AbstractApplicationContext.destroyBeans(AbstractApplicationContext.java:921)

at org.springframework.context.support.AbstractApplicationContext.doClose(AbstractApplicationContext.java:895)

at org.springframework.context.support.AbstractApplicationContext.close(AbstractApplicationContext.java:841)

at org.springframework.web.context.ContextLoader.closeWebApplicationContext(ContextLoader.java:579)

at org.springframework.web.context.ContextLoaderListener.contextDestroyed(ContextLoaderListener.java:115)

at org.apache.catalina.core.StandardContext.listenerStop(StandardContext.java:5014)

at org.apache.catalina.core.StandardContext.stopInternal(StandardContext.java:5659)

at org.apache.catalina.util.LifecycleBase.stop(LifecycleBase.java:232)

at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:160)

at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:901)

at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:877)

at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:632)

at org.apache.catalina.startup.HostConfig.manageApp(HostConfig.java:1740)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:498)

at org.apache.tomcat.util.modeler.BaseModelMBean.invoke(BaseModelMBean.java:301)

at com.sun.jmx.interceptor.DefaultMBeanServerInterceptor.invoke(DefaultMBeanServerInterceptor.java:819)

at com.sun.jmx.mbeanserver.JmxMBeanServer.invoke(JmxMBeanServer.java:801)

at org.apache.catalina.mbeans.MBeanFactory.createStandardContext(MBeanFactory.java:618)

at org.apache.catalina.mbeans.MBeanFactory.createStandardContext(MBeanFactory.java:565)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:498)

at org.apache.tomcat.util.modeler.BaseModelMBean.invoke(BaseModelMBean.java:301)

at com.sun.jmx.interceptor.DefaultMBeanServerInterceptor.invoke(DefaultMBeanServerInterceptor.java:819)

at com.sun.jmx.mbeanserver.JmxMBeanServer.invoke(JmxMBeanServer.java:801)

at javax.management.remote.rmi.RMIConnectionImpl.doOperation(RMIConnectionImpl.java:1468)

at javax.management.remote.rmi.RMIConnectionImpl.access$300(RMIConnectionImpl.java:76)

at javax.management.remote.rmi.RMIConnectionImpl$PrivilegedOperation.run(RMIConnectionImpl.java:1309)

at javax.management.remote.rmi.RMIConnectionImpl.doPrivilegedOperation(RMIConnectionImpl.java:1401)

at javax.management.remote.rmi.RMIConnectionImpl.invoke(RMIConnectionImpl.java:829)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:498)

at sun.rmi.server.UnicastServerRef.dispatch(UnicastServerRef.java:357)

at sun.rmi.transport.Transport$1.run(Transport.java:200)

at sun.rmi.transport.Transport$1.run(Transport.java:197)

at java.security.AccessController.doPrivileged(Native Method)

at sun.rmi.transport.Transport.serviceCall(Transport.java:196)

at sun.rmi.transport.tcp.TCPTransport.handleMessages(TCPTransport.java:568)

at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run0(TCPTransport.java:826)

at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.lambda$run$0(TCPTransport.java:683)

at java.security.AccessController.doPrivileged(Native Method)

at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run(TCPTransport.java:682)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)

at java.lang.Thread.run(Thread.java:748)

四月 26, 2019 11:45:52 下午 org.apache.catalina.core.ApplicationContext log

信息: ContextListener: contextInitialized()

四月 26, 2019 11:45:52 下午 org.apache.catalina.core.ApplicationContext log

信息: SessionListener: contextInitialized()

四月 26, 2019 11:45:52 下午 org.apache.catalina.core.ApplicationContext log

信息: ContextListener: attributeAdded('org.apache.jasper.compiler.TldLocationsCache', 'org.apache.jasper.compiler.TldLocationsCache@326c5d95')

```

---

## 场景描述

ssm整合shiro，shiro添加ehcache配置的时候。

---

## 异常原因

### 我的

BeanFactory not initialized or already closed

大概猜到是bean注入错误被，看了下springmvc，shiro的配置文件，发现是我把ehcache的配置 复制错到shiro里面了，导致不能加载正确的bean。

### 还有可能是

启用注解时，

applicationContext.xml文件头部需要加入

xmlns:context="http://www.springframework.org/schema/context"

但是接着会报错误`在这里插入代码片`

BeanFactory not initialized or already closed - call 'refresh' before access

原因是还要在bean头文件中加入http://www.springframework.org/schema/contexthttp://www.springframework.org/schema/context/spring-context-2.5.xsd

```

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:aop="http://www.springframework.org/schema/aop"

xmlns:tx="http://www.springframework.org/schema/tx"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="

http://www.springframework.org/schema/beanshttp://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/txhttp://www.springframework.org/schema/tx/spring-tx-3.0.xsd

http://www.springframework.org/schema/aophttp://www.springframework.org/schema/aop/spring-aop-3.0.xsdx

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd">

<context:annotation-config/>

```

bean头文件配置

https://blog.csdn.net/u010417178/article/details/52681232

```

1.我创建的是web工程，在web.xml中将contextConfigLocation改成<param-value>classpath:\*\*\*.xml</param-value>即可；（\*\*\*是你Bean的配置文件名)；

2.如果创建的不是web工程，则可能是如下产生的错误：

ApplicationContext ctx = new ClassPathXmlApplicationContext();

没有指定Bean配置文件，Spring实例化BeanFactory的时候默认到classPath下面查找名为applicationContext.xml的文件的，如果没有指定配置文件，则会报错。

改成：ApplicationContext ctx = new ClassPathXmlApplicationContext("\*\*\*.xml"); (\*\*\*是你指定的配置文件的名字）。

然后可以成功运行程序。

```

https://blog.csdn.net/zr\_1877/article/details/78653849

## 解决方案

---

## 参考资料

> 这里是引用