**SSM的工具类项目**

# WebSocket使用

## Java类

路径：com.scy.ssm.websocket

## 前端页面

路径：utils\src\main\webapp\webSocket\webScoket.html

# Spring的Session同步

## 配置项

### Pom.xml

<!-- Jedis -->

<dependency>

<groupId>redis.clients</groupId>

<artifactId>jedis</artifactId>

<version>2.9.0</version>

</dependency>

<!-- Spring Data Redis -->

<dependency>

<groupId>org.springframework.data</groupId>

<artifactId>spring-data-redis</artifactId>

<version>1.7.3.RELEASE</version>

</dependency>

<!-- Spring Session -->

<dependency>

<groupId>org.springframework.session</groupId>

<artifactId>spring-session</artifactId>

<version>1.2.2.RELEASE</version>

</dependency>

<!-- Apache Commons Pool -->

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-pool2</artifactId>

<version>2.4.2</version>

</dependency>

### Web.xml

设置 springSessionRepositoryFilter

### redis.properties

#redis config

redis.pool.maxTotal=105

redis.pool.maxIdle=10

redis.pool.maxWaitMillis=5000

redis.pool.testOnBorrow=true

#redis \u5355\u8282\u70B9\u914D\u7F6E

redis.ip=192.168.91.1

redis.port=6379

### spring-mybatis.xml

<!-- Redis 配置 -->

<bean id="jedisPoolConfig" class="redis.clients.jedis.JedisPoolConfig">

<property name="maxTotal" value="${redis.pool.maxTotal}" />

<property name="maxIdle" value="${redis.pool.maxIdle}" />

<property name="maxWaitMillis" value="${redis.pool.maxWaitMillis}" />

<property name="testOnBorrow" value="${redis.pool.testOnBorrow}" />

</bean>

<bean class="org.springframework.session.data.redis.config.annotation.web.http.RedisHttpSessionConfiguration"/>

<!-- redis单节点数据库连接配置 -->

<bean class="org.springframework.data.redis.connection.jedis.JedisConnectionFactory">

<property name="hostName" value="${redis.ip}" />

<property name="port" value="${redis.port}" />

<property name="poolConfig" ref="jedisPoolConfig" />

</bean>

## 参考教程：

<https://blog.csdn.net/qq_25775675/article/details/94381197>

# POI的使用

## Excel

pom.xml 引入

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>4.1.0</version>

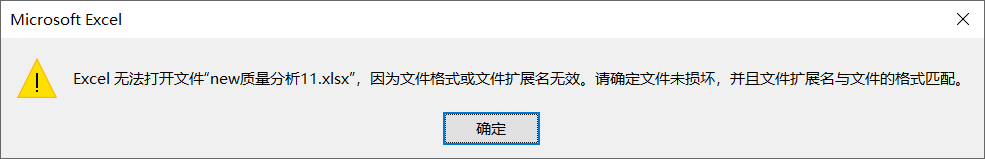
</dependency>

### 读取

### 写出

路径：com.scy.utils.poi.excel.WriteExcel

**问题：导出文件打开失败**



**原因：引入了batik-transcoder 导致java运行使用了jdk1.5的版本。**

<dependency>

<groupId>batik</groupId>

<artifactId>batik-transcoder</artifactId>

<version>1.6-1</version>

</dependency>

**解决：将这个jar换掉**

<dependency>

<groupId>org.apache.xmlgraphics</groupId>

<artifactId>batik-transcoder</artifactId>

<version>1.12</version>

</dependency>

# PageHelper使用

## 引入jar

<dependency>

<groupId>com.github.pagehelper</groupId>

<artifactId>pagehelper</artifactId>

<version>5.1.2</version>

</dependency>

## 方法一

设置Page page = PageHelper.startPage(2, 3);即可使用，无需设置sqlSessionFactory的插件。否则会报错。

Page page = PageHelper.startPage(2, 3);

List<Map<String, Object>> resultList = sqlSessionTemplate.selectList(map.get("sqlMapId").toString(), map);

System.out.println(page.toString());

## 方法二

在spring-mybatis.xml 文件中设置sqlSessionFactory的插件。或者在sqlMapConfig.xml 文件中设置。不可以同时设置，只能设置一次，否则会报错的。

spring-mybatis.xml文件

<property name="plugins">

<array>

<bean class="com.github.pagehelper.PageInterceptor">

<property name="properties">

<!--使用下面的方式配置参数，一行配置一个 -->

<value>

params=value1

</value>

</property>

</bean>

</array>

</property>

sqlMapConfig.xml文件

<plugins>

<!-- com.github.pagehelper为PageHelper类所在包名 -->

<plugin interceptor="com.github.pagehelper.PageInterceptor">

<!-- 使用下面的方式配置参数，后面会有所有的参数介绍 -->

<property name="param1" value="value1"/>

</plugin>

</plugins>

List<Map<String, Object>> resultList = sqlSessionTemplate.selectList(map.get("sqlMapId").toString(), map, new RowBounds(0, 3));

PageInfo pageInfo = new PageInfo<>(resultList, 3);

System.out.println(pageInfo.toString());