

Mybatis分页插件PageHelper接入指南

一、二方库依赖配置

```
<dependency>
  <groupId>com.github.pagehelper</groupId>
  <artifactId>pagehelper</artifactId>
  <version>5.1.3</version>
</dependency>
```

二、spring配置

```
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
  <!-- 注意其他配置 -->
  <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>
</bean>
```

params和value说明：

1. helperDialect：分页插件会自动检测当前的数据库链接，自动选择合适的分页方式。
你可以配置helperDialect属性来指定分页插件使用哪种方言。配置时，可以使用下面的缩写值：oracle,mysql,mariadb,sqlite,hsqldb,postgresql,db2,sqlserver,informix,h2,sqlserver2012,derby
特别注意：使用 SqlServer2012 数据库时，需要手动指定为 sqlserver2012，否则会使用 SqlServer2005 的方式进行分页。
你也可以实现 AbstractHelperDialect，然后配置该属性为实现类的全限定名称即可使用自定义的实现方法。
2. offsetAsPageNum：默认值为 false，该参数对使用 RowBounds 作为分页参数时有效。当该参数设置为 true 时，会将 RowBounds 中的 offset 参数当成 pageNum 使用，可以用页码和页面大小两个参数进行分页。
3. rowBoundsWithCount：默认值为false，该参数对使用 RowBounds 作为分页参数时有效。当该参数设置为true时，使用 RowBounds 分页会进行 count 查询。
4. pageSizeZero：默认值为 false，当该参数设置为 true 时，如果 pageSize=0 或者 RowBounds.limit = 0 就会查询出全部的结果（相当于没有执行分页查询，但是返回结果仍然是 Page 类型）。
5. reasonable：分页合理化参数，默认值为false。当该参数设置为 true 时，pageNum<=0 时会查询第一页，pageNum>pages（超过总数时），会查询最后一页。默认false 时，直接根据参数进行查询。
6. params：为了支持startPage(Object params)方法，增加了该参数来配置参数映射，用于从对象中根据属性名取值，可以配置pageNum,pagesize,count,pagesizeZero,reasonable，不配置映射的用默认值，默认值为pageNum=pageNum;pageSize=pageSize;count=countSql;reasonable=reasonable;pageSizeZero=pageSizeZero。
7. supportMethodsArguments：支持通过 Mapper 接口参数来传递分页参数，默认值false，分页插件会从查询方法的参数值中，自动根据上面 params 配置的字段中取值，查找到合适的值时就会自动分页。使用方法可以参考测试代码中的 com.github.pagehelper.test.basic 包下的 ArgumentsMapTest 和 ArgumentsObjTest。
8. autoRuntimeDialect：默认值为 false。设置为 true 时，允许在运行时根据多数数据源自动识别对应方言的分页（不支持自动选择sqlserver2012，只能使用sqlserver），用法和注意事项参考下面的场景五。
9. closeConn：默认值为 true。当使用运行时动态数据源或没有设置 helperDialect 属性自动获取数据库类型时，会自动获取一个数据库连接，通过该属性来设置是否关闭获取的这个连接，默认true关闭，设置为 false 后，不会关闭获取的连接，这个参数的设置要根据自己选择的数据源来决定。
10. aggregateFunctions(5.1.5+)：默认为所有常见数据库的聚合函数，允许手动添加聚合函数（影响行数），所有以聚合函数开头的函数，在进行 count 转换时，会套一层。其他函数和列会被替换为 count(0)，其中count列可以自己配置。

三、springboot配置

1. boot-starter依赖

```
<dependency>
<groupId>com.github.pagehelper</groupId>
<artifactId>pagehelper-spring-boot-starter</artifactId>
<version>1.2.4</version>
</dependency>
```

2. yaml配置

```
#pagehelper分页插件
pagehelper:
  helperDialect: mysql
  reasonable: true
  supportMethodsArguments: true
  params: count=countSql
```

四、代码中使用

```
//第一种，RowBounds方式的调用
List<Country> list = sqlSession.selectList("x.y.selectIf", null, new RowBounds(0, 10));

//第二种，Mapper接口方式的调用，推荐这种使用方式。
PageHelper.startPage(1, 10);
List<Country> list = countryMapper.selectIf(1);

//第三种，Mapper接口方式的调用，推荐这种使用方式。
PageHelper.offsetPage(1, 10);
List<Country> list = countryMapper.selectIf(1);

//第四种，参数方法调用
//存在以下 Mapper 接口方法，你不需要在 xml 处理后两个参数
public interface CountryMapper {
    List<Country> selectByPageNumSize(
        @Param("user") User user,
        @Param("pageNum") int pageNum,
        @Param("pageSize") int pageSize);
}
//配置supportMethodsArguments=true
//在代码中直接调用：
List<Country> list = countryMapper.selectByPageNumSize(user, 1, 10);

//第五种，参数对象
//如果 pageNum 和 pageSize 存在于 User 对象中，只要参数有值，也会被分页
//有如下 User 对象
public class User {
    //其他fields
    //下面两个参数名和 params 配置的名字一致
    private Integer pageNum;
    private Integer pageSize;
}
//存在以下 Mapper 接口方法，你不需要在 xml 处理后两个参数
public interface CountryMapper {
    List<Country> selectByPageNumSize(User user);
}
//当 user 中的 pageNum!= null && pageSize!= null 时，会自动分页
List<Country> list = countryMapper.selectByPageNumSize(user);
```

```
//第六种, ISelect 接口方式
//jdk6,7用法, 创建接口
Page<Country> page = PageHelper.startPage(1, 10).doSelectPage(new ISelect() {
    @Override
    public void doSelect() {
        countryMapper.selectGroupBy();
    }
});
//jdk8 lambda用法
Page<Country> page = PageHelper.startPage(1, 10).doSelectPage(()-> countryMapper.selectGroupBy());

//也可以直接返回PageInfo, 注意doSelectPageInfo方法和doSelectPage
pageInfo = PageHelper.startPage(1, 10).doSelectPageInfo(new ISelect() {
    @Override
    public void doSelect() {
        countryMapper.selectGroupBy();
    }
});
//对应的lambda用法
pageInfo = PageHelper.startPage(1, 10).doSelectPageInfo(() -> countryMapper.selectGroupBy());

//count查询, 返回一个查询语句的count数
long total = PageHelper.count(new ISelect() {
    @Override
    public void doSelect() {
        countryMapper.selectLike(country);
    }
});
```

```
});  
//lambda  
total = PageHelper.count(()->countryMapper.selectLike(country));
```