

昵称：[扎心了老铁](#)
园龄：2年11个月
粉丝：87
关注：0
[+加关注](#)

<

2019年10月

>

日	一	二	三	四	五	六
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

搜索

找找看

谷歌搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

随笔分类

[Django\(8\)](#)
[Docker\(1\)](#)
[ElasticSearch\(13\)](#)
[Hadoop\(4\)](#)
[Hive\(9\)](#)
[Java\(30\)](#)
[Javascript\(2\)](#)
[Kafka\(2\)](#)
[Linux\(6\)](#)
[MapReduce\(2\)](#)
[MongoDB\(1\)](#)
[MySQL\(2\)](#)
[Nginx\(1\)](#)
[Python\(33\)](#)
[Redis\(13\)](#)
[Spark\(8\)](#)
[Spring\(15\)](#)
[thrift\(2\)](#)
[ZooKeeper\(5\)](#)
[大数据\(16\)](#)
[消息队列\(4\)](#)

随笔档案

[2019年4月\(1\)](#)
[2019年3月\(3\)](#)
[2018年3月\(2\)](#)
[2018年2月\(3\)](#)
[2018年1月\(3\)](#)
[2017年12月\(6\)](#)
[2017年11月\(2\)](#)
[2017年10月\(14\)](#)
[2017年9月\(9\)](#)
[2017年8月\(15\)](#)
[2017年7月\(21\)](#)
[2017年6月\(2\)](#)
[2017年5月\(1\)](#)
[2017年4月\(3\)](#)
[2017年1月\(2\)](#)
[2016年12月\(1\)](#)
[2016年11月\(4\)](#)

最新评论

1. Re:Django扩展自定义manage命令

@ 老李博客嗯嗯 谢谢ThanksJ(·ω·)/大佬，给大佬递茶、捏肩、揉腿、捶背(*ゝ▽ゝ！*)...

Mybatis分页插件PageHelper的配置和使用方法

• 前言

在web开发过程中涉及到表格时，例如dataTable，就会产生分页的需求，通常我们将分页方式分为两种：前端分页和后端分页。

前端分页

一次性请求数据表格中的所有记录(ajax)，然后在前端缓存并且计算count和分页逻辑，一般前端组件(例如dataTable)会提供分页动作。

特点是：简单，很适合小规模的web平台；当数据量大的时候会产生性能问题，在查询和网络传输的时间会很长。

后端分页

在ajax请求中指定页码(pageNum)和每页的大小(pageSize)，后端查询出当页的数据返回，前端只负责渲染。

特点是：复杂一些；性能瓶颈在MySQL的查询性能，这个当然可以调优解决。一般来说，web开发使用的是这种方式。

我们说的也是后端分页。

• MySQL对分页的支持

简单来说MySQL对分页的支持是通过limit子句。请看下面的例子。

limit关键字的用法是
LIMIT [offset,] rows
offset是相对于首行的偏移量(首行是0)，rows是返回条数。

每页10条记录，取第一页，返回的是前10条记录
select * from tableA limit 0,10;
每页10条记录，取第二页，返回的是第11条记录，到第20条记录，
select * from tableA limit 10,10;

这里提一嘴的是，MySQL在处理分页的时候是这样的：

limit 1000,10 - 过滤出1010条数据，然后丢弃前1000条，保留10条。当偏移量大的时候，性能会有所下降。

limit 100000,10 - 会过滤10w+10条数据，然后丢弃前10w条。如果在分页中发现了性能问题，可以根据这个思路调优。

• Mybatis分页插件PageHelper

在使用Java Spring开发的时候，Mybatis算是对数据库操作的利器了。不过在处理分页的时候，Mybatis并没有什么特别的方法，一般需要自己去写limit子句实现，成本较高。好在有个PageHelper插件。

1、POM依赖

Mybatis的配置就不多提了。PageHelper的依赖如下。需要新的版本可以去maven上自行选择

```
1 <dependency>
2     <groupId>com.github.pagehelper</groupId>
3     <artifactId>pagehelper</artifactId>
4     <version>4.1.4</version>
5 </dependency>
```

2、Mybatis对PageHelper的配置

打开Mybatis配置文件，一般在Resource路径下。我这里叫mybatis-config.xml。

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org
4
```

@ 小小灰def load_command_class(app_name, name): """ Given a command name and an application name, retur...

--老李博客

3. Re:Kazoo Python Zookeeper 选主

我测试了代码，但当网络断开时，程序陷入了死循环

self.create_instance() 中 create 方法不断的抛出 SessionExpiredError

--龙泉居士

4. Re:Elasticsearch JAVA api轻松搞定groupBy聚合

可以声明一下环境吗？

谢谢

--Merlin_ee

5. Re:数据库表反向生成(二) Django ORM inspectdb

楼主 这个反向生成是生成一个model.py吗，我是需求是建立了多个子应用，有多个model.py 这要怎么——对应啊，还是说是可以cd到子应用的目录下然后 python manage.py ins...

--绪方

阅读排行榜

1. Mybatis分页插件PageHelper的配置和使用方法(102668)
2. springboot kafka集成(实现producer和consumer)(80908)
3. Elasticsearch-sql 用SQL查询Elasticsearch(44597)
4. Linux Redis集群搭建与集群客户端实现(32478)
5. springboot mybatis 事务管理(31001)

评论排行榜

1. Elasticsearch-sql 用SQL查询Elasticsearch(7)
2. Mybatis分页插件PageHelper的配置和使用方法(5)
3. 使用spark与MySQL进行数据交互的方法(3)
4. Django扩展自定义manage命令(3)
5. 数据库表反向生成(二) Django ORM in inspectdb(3)

推荐排行榜

1. Mybatis分页插件PageHelper的配置和使用方法(6)
2. Linux Redis集群搭建与集群客户端实现(5)
3. springboot kafka集成(实现producer和consumer)(5)
4. python concurrent.futures(4)
5. spark三种连接join(3)

```
7 <settings>
8 <!-- 使全局的映射器启用或禁用缓存。 -->
9 <setting name="cacheEnabled" value="true"/>
10 <!-- 全局启用或禁用延迟加载。当禁用时，所有关联对象都会即时加载。 -->
11 <setting name="lazyLoadingEnabled" value="true"/>
12 <!-- 当启用时，有延迟加载属性的对象在被调用时将会完全加载任意属性。否则，每种属性将会按需要
13 <setting name="aggressiveLazyLoading" value="true"/>
14 <!-- 是否允许单条sql 返回多个数据集 （取决于驱动的兼容性） default:true -->
15 <setting name="multipleResultSetsEnabled" value="true"/>
16 <!-- 是否可以使用列的别名（取决于驱动的兼容性） default:true -->
17 <setting name="useColumnLabel" value="true"/>
18 <!-- 允许JDBC 生成主键。需要驱动器支持。如果设为true，这个设置将强制使用被生成的主键，有
19 <setting name="useGeneratedKeys" value="true"/>
20 <!-- 指定 MyBatis 如何自动映射 数据基表的列 NONE：不隐射 PARTIAL：部分 FULL：全部 -->
21 <setting name="autoMappingBehavior" value="PARTIAL"/>
22 <!-- 这是默认的执行类型 （SIMPLE：简单； REUSE：执行器可能重复使用prepared statements；
23 <setting name="defaultExecutorType" value="SIMPLE"/>
24 <!-- 使用驼峰命名法转换字段。 -->
25 <setting name="mapUnderscoreToCamelCase" value="true"/>
26 <!-- 设置本地缓存范围 session：就会有数据的共享 statement：语句范围 （这样就不会有数据的共
27 <setting name="localCacheScope" value="SESSION"/>
28 <!-- 设置但JDBC类型为空时，某些驱动程序 要指定值，default:OTHER，插入空值时不需要指定类型
29 <setting name="jdbcTypeForNull" value="NULL"/>
30 </settings>
31
32 <plugins>
33 <plugin interceptor="com.github.pagehelper.PageHelper">
34 <property name="dialect" value="mysql"/>
35 <property name="offsetAsPageNum" value="false"/>
36 <property name="rowBoundsWithCount" value="false"/>
37 <property name="pageSizeZero" value="true"/>
38 <property name="reasonable" value="false"/>
39 <property name="supportMethodsArguments" value="false"/>
40 <property name="returnPageInfo" value="none"/>
41 </plugin>
42 </plugins>
43 </configuration>
```

这里要注意的是PageHelper相关的配置。

如果你没有加载Mybatis配置文件，那么使用的是Mybatis默认的配置。如何加载Mybatis配置文件呢？

到你的dataSouce配置中。

在配置sqlSessionFactory的时候，指定Mybatis核心配置文件和mapper的路径，代码如下

```
1 @Bean(name = "moonlightSqlSessionFactory")
2 @Primary
3 public SqlSessionFactory moonlightSqlSessionFactory(@Qualifier("moonlightData") DataSource
4 SqlSessionFactoryBean bean = new SqlSessionFactoryBean();
5 bean.setDataSource(dataSource);
6 bean.setMapperLocations(new PathMatchingResourcePatternResolver().getResources("clas
7 bean.setConfigLocation(new ClassPathResource("mybatis-config.xml"));
8 return bean.getObject();
9 }
```

说明：

这里配置的mapper.xml存放路径，在Resource/mybatis-mapper文件夹下

这里配置的mybatis-config.xml文件，在Resource/下

3、分页

准备一个mapper.xml，测试就随便写一个吧，干脆就用工程里的一个。

这里这个查询，是一个典型的多条件查询，我们要做的是对多条件匹配到的记录进行分页。

```

3  <mapper namespace="com.kangaroo.studio.moonlight.dao.mapper.MoonlightMapper">
4    <resultMap id="geoFenceList" type="com.kangaroo.studio.moonlight.dao.model.GeoFence">
5      <constructor>
6        <idArg column="id" javaType="java.lang.Integer" jdbcType="INTEGER" />
7        <arg column="name" javaType="java.lang.String" jdbcType="VARCHAR" />
8        <arg column="type" javaType="java.lang.Integer" jdbcType="INTEGER" />
9        <arg column="group" javaType="java.lang.String" jdbcType="VARCHAR" />
10       <arg column="geo" javaType="java.lang.String" jdbcType="VARCHAR" />
11       <arg column="createTime" javaType="java.lang.String" jdbcType="VARCHAR" />
12       <arg column="updateTime" javaType="java.lang.String" jdbcType="VARCHAR" />
13     </constructor>
14   </resultMap>
15
16   <sql id="base_column">id, name, type, `group`, geo, createTime, updateTime </sql>
17
18   <select id="queryGeoFence" parameterType="com.kangaroo.studio.moonlight.dao.model.Geo
19     select <include refid="base_column"/> from geoFence where 1=1
20     <if test="type != null">
21       and type = #{type}
22     </if>
23     <if test="name != null">
24       and name like concat('%', #{name}, '%')
25     </if>
26     <if test="group != null">
27       and `group` like concat('%', #{group}, '%')
28     </if>
29     <if test="startTime != null">
30       and createTime >= #{startTime}
31     </if>
32     <if test="endTime != null">
33       and createTime <= #{endTime}
34     </if>
35   </select>
36 </mapper>

```

在Mapper.java接口中编写对应的方法

```
1 List<GeoFence> queryGeoFence(GeoFenceQueryParam geoFenceQueryParam);
```

先上分页代码，后面再说明

```

1  @RequestMapping(value = "/fence/query", method = RequestMethod.POST)
2    @ResponseBody
3    public ResponseEntity<Response> queryFence(@RequestBody GeoFenceQueryParam geoFence
4      try {
5        Map<String, Object> data = new HashMap<>();
6        Integer pageNum = geoFenceQueryParam.getPageNum() != null ? geoFenceQueryParam.
7        Integer pageSize = geoFenceQueryParam.getPageSize() != null ? geoFenceQueryPara
8        Page page = PageHelper.startPage(pageNum, pageSize, true);
9        List<GeoFence> list = moonlightMapper.queryGeoFence(geoFenceQueryParam);
10       data.put("total", page.getTotal());
11       data.put("nowPage", pageNum);
12       data.put("data", list);
13       return new ResponseEntity<> (
14         new Response(ResultCode.SUCCESS, "查询geoFence成功", data),
15         HttpStatus.OK);
16     } catch (Exception e) {
17       logger.error("查询geoFence失败", e);
18       return new ResponseEntity<> (
19         new Response(ResultCode.EXCEPTION, "查询geoFence失败", null),
20         HttpStatus.INTERNAL_SERVER_ERROR);
21     }
22   }

```

说明：

1、PageHelper的优点是，分页和Mapper.xml完全解耦。实现方式是以插件的形式，对Mybatis执行的流程进行了强化，添加了总数count和limit查询。属于物理分页。

2、Page page = PageHelper.startPage(pageNum, pageSize, true); - true表示需要统计总数，这样会多进行一次请求select count(0); 省略掉true参数只返回分页数据。

1)统计总数，（将SQL语句变为 select count(0) from xxx, 只对简单SQL语句其效果，复杂SQL语句需要自己写）

```
Page<?> page = PageHelper.startPage(1,-1);
```

```
long count = page.getTotal();
```

2)分页，pageNum - 第N页， pageSize - 每页M条数

A、只分页不统计(每次只执行分页语句)

```
PageHelper.startPage([pageNum],[pageSize]);
```

```
List<?> pagelist = queryForList( xxx.class, "queryAll" , param);
```

//pagelist就是分页之后的结果

B、分页并统计（每次执行2条语句，一条select count语句，一条分页语句）适用于查询分页时数据发生变动，需要将实时的变动信息反映到分页结果上

```
Page<?> page = PageHelper.startPage([pageNum],[pageSize],[iscount]);
```

```
List<?> pagelist = queryForList( xxx.class , "queryAll" , param);
```

```
long count = page.getTotal();
```

//也可以 List<?> pagelist = page.getList(); 获取分页后的结果集

3)使用PageHelper查全部（不分页）

```
PageHelper.startPage(1,0);
```

```
List<?> alllist = queryForList( xxx.class , "queryAll" , param);
```

4)PageHelper的其他API

```
String orderBy = PageHelper.getOrderBy(); //获取orderBy语句
```

```
Page<?> page = PageHelper.startPage(Object params);
```

```
Page<?> page = PageHelper.startPage(int pageNum, int pageSize);
```

```
Page<?> page = PageHelper.startPage(int pageNum, int pageSize, boolean isCount);
```

```
Page<?> page = PageHelper.startPage(pageNum, pageSize, orderBy);
```

Page<?> page = PageHelper.startPage(pageNum, pageSize, isCount, isReasonable); //isReasonable 分页合理化,null时用默认配置

Page<?> page = PageHelper.startPage(pageNum, pageSize, isCount, isReasonable, isPageSizeZero);
//isPageSizeZero是否支持PageSize为0，true且pageSize=0时返回全部结果，false时分页,null时用默认配置

5)、默认值

//RowBounds参数offset作为PageNum使用 - 默认不使用

```
private boolean offsetAsPageNum = false;
```

//RowBounds是否进行count查询 - 默认不查询

```
private boolean rowBoundsWithCount = false;
```

//当设置为true的时候，如果pagesize设置为0（或RowBounds的limit=0），就不执行分页，返回全部结果

```
private boolean pageSizeZero = false;
```

//分页合理化

```
private boolean reasonable = false;
```

//是否支持接口参数来传递分页参数，默认false

```
private boolean supportMethodsArguments = false;
```

3、有一个安全性问题，需要注意一下，不然可能导致分页错乱。我这里直接粘贴了[这篇博客](#)里的一段话。

1 4. 什么时候会导致不安全的分页？

2

3 PageHelper 方法使用了静态的 ThreadLocal 参数，分页参数和线程是绑定的。

```
6
7  如果代码在进入 Executor 前发生异常，就会导致线程不可用，这属于人为的 Bug（例如接口方法和 XML
8
9  但是如果你写出下面这样的代码，就是不安全的用法：
10
11 PageHelper.startPage(1, 10);
12 List<Country> list;
13 if(param1 != null){
14     list = countryMapper.selectIf(param1);
15 } else {
16     list = new ArrayList<Country>();
17 }
18 这种情况下由于 param1 存在 null 的情况，就会导致 PageHelper 生产了一个分页参数，但是没有被消
19
20 上面这个代码，应该写成下面这个样子：
21
22 List<Country> list;
23 if(param1 != null){
24     PageHelper.startPage(1, 10);
25     list = countryMapper.selectIf(param1);
26 } else {
27     list = new ArrayList<Country>();
28 }
29 这种写法就能保证安全。
30
31 如果你对此不放心，你可以手动清理 ThreadLocal 存储的分页参数，可以像下面这样使用：
32
33 List<Country> list;
34 if(param1 != null){
35     PageHelper.startPage(1, 10);
36     try{
37         list = countryMapper.selectAll();
38     } finally {
39         PageHelper.clearPage();
40     }
41 } else {
42     list = new ArrayList<Country>();
43 }
44 这么写很不好看，而且没有必要。
```


官方文档，给你参考：

<https://github.com/pagehelper/Mybatis-PageHelper/blob/master/wikis/zh/HowToUse.md>

分类: [Java](#), [Spring](#)

好文要顶 关注我 收藏该文





[扎心了老铁](#)
[关注 - 0](#)
[粉丝 - 87](#)
[+加关注](#)

61

« 上一篇: [CentOS环境下Docker私有仓库搭建](#)
» 下一篇: [springboot mybatis redis 二级缓存](#)

posted @ 2017-12-07 12:19 扎心了老铁 阅读(102667) 评论(5) 编辑 收藏

评论列表

- #1楼 2018-03-20 11:01 optor 回复 引用
了解一点点
支持(0) 反对(0)
- #2楼 2018-08-01 15:00 猴子童鞋 回复 引用

回复 引用

支持(0) 反对(0)

回复 引用

支持(0) 反对(0)

回复 引用

支持(0) 反对(0)

发表评论

评论内容：

[退出](#) [订阅评论](#)

【福利】学AI有奖：博客园&华为云 Modelarts 有奖训练营

- Mybatis框架插件PageHelper的使用
- Mybatis 的分页插件PageHelper-4.1.1的使用
- Mybatis 的分页插件PageHelper-4.1.1的使用
- Mybatis 的分页插件 PageHelper
- Mybatis分页插件PageHelper的配置和使用方法

- **因美国制裁：Oracle 终止与「委内瑞拉」的业务合作**

- [“水氢发动机”已经凉了，日本氢能源车却越活越好](#)
- [首个5G智能电网切片集采：华为中标，报价403万](#)
- » [更多新闻...](#)