



链滴

# Java 接口继承使用 (什么? 接口还能继承接口, 好吧第一次见)

作者: [ludengke95](#)

原文链接: <https://ld246.com/article/1595381016021>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## Java接口继承使用（多数据库源类似查询的封装）

### 场景描述

我是因为一个中间件的项目（由用户维护多个数据库链接信息，系统从多数据源中读取库中的全部表发现的这个东西。为了让扩展数据库类型比较方便，所以这个地方需要抽象出一个数据库基类，每次增数据库类型的时候，只是写新类继承基类就可以了，通过反射呀，或者简单的ifelse的方式调用就可以了。我就简单写一下实现需要哪几部分。

### 实现

#### Holder类

我比较懒，一个参数从controller传递到最内层是真的烦人呢，使用ThreadLocal就比较方便了，但也别乱用呀。这个地方就是保存了数据库类型。ThreadLocal是每个线程独享的，所以不会出现多线程互相影响的问题。

```
import org.springframework.core.NamedThreadLocal;
```

```
/**  
 * 描述:  
 * 数据库类型holder  
 *  
 * @author ludengke  
 * @create 2020-07-21 16:18  
 */  
public class DatabaseTypeHolder {  
    private static final ThreadLocal<String> DATABASE_TYPE HOLDER = new NamedThreadLocal<>("DATABASE_TYPE");
```

```

public static String getDatabaseType() {
    return DATABASE_TYPE_HOLDER.get();
}

public static void setDatabaseType(String databaseType) {
    DATABASE_TYPE_HOLDER.set(databaseType);
}

public static void clearUserInfo() {
    DATABASE_TYPE_HOLDER.remove();
}
}

```

## mapper层

mapper层分为mapper类和mapper xml

### mapper类

```

/**
 * 描述:
 * 数据库操作基类，不加Mapper注解，不加Mapper注解
 * @author ludengke
 * @create 2020-07-21 16:22
 */
public interface DatabaseOptMapper {

    /**
     * 查询数据库表信息
     * @return 返回表信息
     */
    List<TableInfo> queryTableInfo();
}

/**
 * 描述:
 * mysql操作类，加Mapper注解

 * @author ludengke
 * @create 2020-07-21 16:22
 */
@Mapper
public interface MysqlOptMapper extends DatabaseOptMapper{
}

/**
 * 描述:
 * PGSQL操作类
 * @author ludengke
 * @create 2020-07-21 16:22
 */
@Mapper

```

```
public interface PgsqloptMapper extends DatabaseOptMapper{
}
```

mysql操作类的mapper.xml用于查询，当前连接的所有表

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/ybatis-3-mapper.dtd">
<mapper namespace="com.MysqloptMapper">
  <select id="queryTableInfo" resultType="com.TableInfo">
    SELECT
      `table_name`,
      table_comment
    FROM
      information_schema.`TABLES`
    WHERE
      table_schema = (SELECT DATABASE ())
  </select>
</mapper>
```

PGSQL操作类的mapper.xml用于查询，当前连接的所有表

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/ybatis-3-mapper.dtd">
<mapper namespace="com.PgsqloptMapper">
  <select id="queryTableInfo" resultType="com.TableInfo">
    with tmp_tab as (
      select pc.oid as ooid,pn.nspname,pc.*
      from pg_class pc
      left join pg_namespace pn on pc.relnamespace = pn.oid
      where pc.relkind in ('r')
      and pc.relnamespace in (select pn.oid from pg_namespace pn where pn.nspname in
select CURRENT_SCHEMA))
      and pc.oid not in (select inhrelid from pg_inherits )
      order by pc.relname
    ),tmp_desc as (
      select pd.*
      from pg_description pd
      where pd.objsubid = 0
    )
    select tab.relname as table_name, de.description as table_commit
      from tmp_tab tab
      left join tmp_desc de on tab.ooid = de.objoid
  </select>
</mapper>
```

## service层

服务方：负责处理不同数据库操作

```
package com.wangdian.bigdata.biserver.dataset.service.impl;
```

```
import com.TableInfo;
```

```

import com.DatabaseTypeHolder;
import com.DatabaseOptMapper;
import com.MysqlOptMapper;
import com.PgsqloptMapper;
import com.IDatabaseOptService;
import org.springframework.stereotype.Service;

import javax.annotation.Resource;
import java.util.List;

/**
 * 描述:
 * 不同数据源操作调用
 *
 * @author ludengke
 * @create 2020-07-21 16:14
 */
@Service
public class DatabaseOptServiceImpl implements IDatabaseOptService {

    @Resource
    private MysqlOptMapper mysqlOptMapper;

    @Resource
    private PgsqloptMapper pgsqloptMapper;

    @Override
    public List<TableInfo> queryTableInfoFromOutSideLink(){
        return this.getDatabaseOptMapper().queryTableInfo();
    }

    private DatabaseOptMapper getDatabaseOptMapper() {
        DatabaseOptMapper result = null;
        switch (DatabaseTypeHolder.getDatabaseType()){
            case "mysql":
                result = mysqlOptMapper;break;
            case "pgsql":
                result = pgsqloptMapper;break;
            default:
                result = mysqlOptMapper;break;
        }
        return result;
    }
}

package com.service;

import com.TableInfo;

import java.util.List;

/**
 * 描述:
 *

```

```
* @author ludengke
* @create 2020-07-21 16:36
*/
public interface IDatabaseOptService {

    /**
     * 查询目标库的表的绑定情况
     * @return
     */
    List<TableInfo> queryTableInfoFromOutSideLink();
}
```

## 设计

这样的设计方便以后的扩展，扩展新的数据库类型的时候步骤比较清晰了。

1. 增加DatabaseOptMapper的实现类，
2. 增加实现类的mapper xml。
3. 在DatabaseOptServiceImpl中增加switch的内容。
4. 如果想做更多的数据库操作，在DatabaseOptMapper增加方法即可，xml记得实现哟。