# 通用模板设计

## DB2模板

一般数据jdbc操作步骤如下：

1）加载驱动器，执行静态代码将DriverManager类文件加载到内存

2）获取连接

3）获取数据库传输器

4）执行SQL，处理结果集

5）释放资源

我们可以看见除了第4步需要我们自己定义，其它的代码可以放在一个通用模板内部。

本框架提供了通用模板，对sql进行接口式管理，面向注解编程简化sql和事务处理。

本框架提供功能：

1）通用模板

2）dao接口实例化

3）事务

### 配置连接信息和框架信息

读取配置信息，基本四大数据连接信息，是否支持池连接。将它们从properties文件读取封装在ConfigInfo对象（采用单例模式，懒汉模式）中。

### 获取数据库连接

数据库连接可以通过传统方式获取或者数据库连接池获取。通过Connect统一获取。获取方式通过配置文件db2中配置isPool

由于connection对象对应自己的请求处理线程，所以它的获取代码单独抽取出来。

### 通用模板&回调

通用模板是针对connection获得传输器，封装操作参数，释放资源prepareStatement资源（同时会释放ResultSet资源）

回调方法针对步骤4

### 结果集封装

结果集多样性，也是比较复杂的，本框架分类：

1）单个实体类

2）list集合

3）数量

## 功能配置

### 获取连接&释放连接

本框架不允许多个线程同时调用一个数据库连接对象，所以我专门写了一个**ConnectionStore**类来管理各线程本地的connection对象。提供了一套连接获取，存储，释放的API。

简化了传统数据库连接获取和创建的过程。

### Dao接口管理

本系统将sql信息封装在注解@SQL中，包含sql语句，实体类类型，操作类型（CRUD）。注解在方法上，代表一条语句在数据库执行的DAO层方法。

通过jdk动态代理模式，生成dao对象，解读dao层方法注解，执行通用模板代码和回调。

最为重要的是数据库连接的获取和资源释放，需要手动获取连接和手动是否资源。Dao接口提供了对应的方法。

#### 针对注解的定义

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| @SQL | |  |  | | --- | --- | | value | Sql语句 | | resultType | 实体类类型，针对查询有用 | | type | 操作类型，CRUD | |  |  | |  |  | |
|  |  |
|  |  |
|  |  |
|  |  |

#### 接口方法解析

Select类型：依据@SQL的resultType封装实体类，依据方法的返回类型判断是否返回集合或者实体类，或者数量。

Delete，Insert，Update：依据接口方法的返回类型：Number的子类或者对应的基本类型。或者void不返回任何值。

## 事务

每一个业务层方法可能会调用多次或者多个dao对象，因此我使用了ThreadLocal存取数据库连接对象，也就是上文所提到过的**ConnectionStore类**。

文档有待补全！尽情期待，思路尽在源码中！