## 数据库持久层设计

### 功能说明

持久化数据访问层提供如下功能：

* 提供POJO对象的简单持久化功能

1. 将POJO类映射为数据库表格
2. 支持动态持久化，开发者可以根据实际需要在程序运行过程中将POJO类映射为数据库表格

* 提供基于Hibernate 的POJO对象的CRUD以及数据库查询服务

1. 基于对象进行数据的CRUD和查询
2. 对Hibernate进行了封装
3. 使用便利，但是大数据量下效率会比较低

* 提供创建、删除表格服务

1. 主要用于在程序运行过程中创建表格、删除表格
2. 告警、性能等大数据量情况下分表处理

* 提供基于JDBC数据库访问服务

1. JDBC数据库访问效率比Hibernate更高
2. 需要开发者写更多的SQL语句

* 提供基于模块的可控制事务

1. 模块内部自己维护数据完整性
2. 不维护其他模块的数据完整性
3. 由开发者根据错误异常情况去维护其他模块数据完整性

### 设计思路

* 采用Hibernate作为持久化组件，在Hibernate基础上做简单包装，包装的对象包括SessionFactory、Session以及Transcation等对象。
* 为了方便进行数据创建、删除、修改以及查询操作，在Session基础上包装基于POJO访问的API。
* 基于模块的可控式事务

实现基本思想：

由开发者根据业务需要、异常情况实现事务的提交、回滚

一个模块提供一个Session，每个模块通过自己的锁控制用户的并发访问

如果调用其他模块的服务，则其他模块的服务不受当前模块事务影响

transcation.begin()

try{

……业务处理

otherModuleService.execute();//其他模块服务执行

…….业务处理

transcation.commit();

}catch(){

(实现自己业务回滚)}

//其他模块服务执行结果不回滚，由开发者自己实现数据完整性约束

transcation.rollback();//只回滚当前模块，当前线程的数据

基于上面接口描述，模块事务实现如下：

SessionFactory Manager sessionFactoryManager;

SessionFactory sessionFactory= sessionFactoryManager.getSessionFacatory (“dataSourceName”);

Session session=sessionFactory. openSession (“vlan”);表示是vlan模块的Session，一个模块需要定义一个Session名称。

Transaction transcation= session. beginTransaction();

try{

…….模块内部业务处理

otherModuleService.execute();//其他模块服务执行

…….模块内部业务处理

transcation.commit();

}catch(){

(实现自己业务回滚)

//只回滚当前模块，当前线程的数据

transcation.rollback();

//其他模块根据实际情况进行垃圾数据清理

otherModuleService.clear();

}

* 多数据源支持

网管系统通常情况下需要支持多个数据源，比如FM、PM单独使用数据库。

实现思想如下：

通过配置文件配置多个数据源，一个数据源对应一个数据库；数据库持久层模块启动的时候读取配置文件，根据数据源配置构建SessionFactory，每个数据源一个SessionFactory。

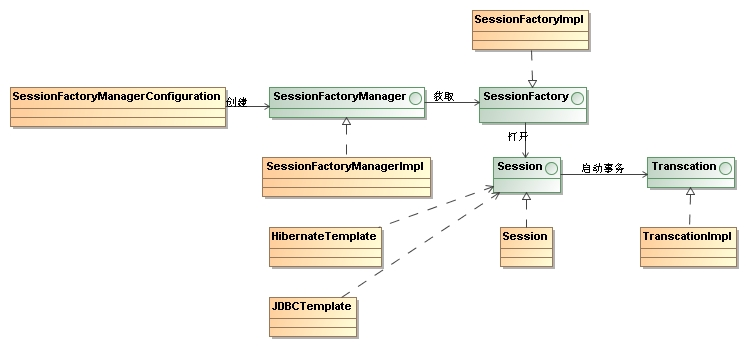
### 体系结构



数据库持久层体系结构

* Hibernate、JDBC：持久层依赖的基础组件；
* SessionFactoryManager：管理SessionFactory的工厂，可以根据数据源名称获取SessionFactory；
* SessionFactory：创建Session的工厂，一个数据源对应一个SessionFactory；
* Session：数据库会话，可以启动数据库事务、获取JDBC数据库连接并提供了数据库访问API
* Transcation：数据库事务，可以通过Session启动打开数据库事务
* HibernateTemplate ：基于POJO的数据库访问API，提供了很多数据库查询、删除、创建API；
* JDBCTemplate：基于JDBC的数据库访问API

### 类图



数据库持久层基础类图

|  |  |
| --- | --- |
| 类名 | 描述 |
| SessionFactoryManagerConfiguration | SessionManagerFactory配置管理器，启动的时候读取配置文件，然后构建SessionFactoryManagerImpl |
| SessionFactoryManager | SessionFactory管理器 |
| SessionFactory | Session工厂，用于创建数据库访问会话Session |
| Session | 数据库访问会话 |
| Transcation | 数据库事务 |
| HibernateTemplate | 基于Pojo对象的查询、创建、修改、删除 |
| JDBCTemplate | 基于JDBC的数据库查询、创建、修改、删除 |

### 时序图

无

### 业务规则和约束

* 多数据源配置文件格式



由于我们自定义了数据源配置文件，所以需要重写hibernate的ConnectionProvider实现类.

### 第三方库

Hibernate3.jar

Hibernate-validate.jar

Hibernate-entitymanager.jar

Hibernate-common-annotation.jar

Hibernate-annotation.jar