# iTree Permission Service

# 软件设计说明书

李晓欢

2015-11-04

### 版本历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 时间 | 版本 | 描述 | 作者 | 审核 |
| 1 | 2015-11-04 | 1.0 | 初步的软件设计说明书 | 李晓欢 |  |

目录

[1. 引言 4](#_Toc434485868)

[1.1 编写目的和范围 4](#_Toc434485869)

[1.2 术语表 4](#_Toc434485870)

[1.3 参考资料 4](#_Toc434485871)

[1.4 使用的工具 4](#_Toc434485872)

[1.5 使用的技术 4](#_Toc434485873)

[2. 模块设计 5](#_Toc434485874)

[2.1模块关系图 5](#_Toc434485875)

[2.2模块划分 5](#_Toc434485876)

[2.3模块描述 5](#_Toc434485877)

[3. 功能设计 6](#_Toc434485878)

[3.1 系统功能模块图 6](#_Toc434485879)

[3.2功能划分 6](#_Toc434485880)

[3.3功能描述 7](#_Toc434485881)

[4．接口设计 8](#_Toc434485882)

[4.1内部接口 8](#_Toc434485883)

[4.2外部接口 8](#_Toc434485884)

[5．数据库设计 8](#_Toc434485885)

[5.1 说明 8](#_Toc434485886)

[5.2 资源管理：配置数据库连接池 9](#_Toc434485887)

[5.3 数据表设计 9](#_Toc434485888)

[6. 系统安全保密设计 11](#_Toc434485889)

[6.1 说明 11](#_Toc434485890)

[6.2 设计 11](#_Toc434485891)

[7. 系统性能设计 11](#_Toc434485892)

[7.1精度 11](#_Toc434485893)

[7.2 时间特性要求 11](#_Toc434485894)

[8.系统出错处理 11](#_Toc434485895)

## 1. 引言

### 1.1 编写目的和范围

本详细设计说明书编写的目的是说明iTree Permission Service的设计，包括程序描述、输入/输出、算法和流程逻辑等，为软件编程和系统维护提供基础。本说明书的预期读者为系统设计人员、软件开发人员、软件测试人员和项目评审人员。

### 1.2 术语表

|  |  |  |
| --- | --- | --- |
| **序号** | **术语或缩略语** | **说明性定义** |
| 1 | uid | User id—— 用户ID |
| 2 | pid | Permission id ——权限ID |
| 3 | rid | Role id —— 角色ID |
| 4 | name | 权限内容 |

### 1.3 参考资料

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 资料名称 | 作者 | 文件编号、版本 |
|  |  |  |  |

### 1.4 使用的工具

数据库管理工具：MyS QL

编程工具： STS

文档编写工具：office 2013

### 1.5 使用的技术

本服务程序使用 java语言编写，主要使用：Hibernate 、 Spring 、 RESTful 等技术。

## 2. 模块设计

### 2.1模块关系图

用户

服务模块

业务逻辑模块

数据连接模块

数据库

图一：模块关系图

### 2.2模块划分

主要划分为数据连接模块、业务逻辑模块、服务模块三个模块。

### 2.3模块描述

##### 2.3.1模块一：数据连接模块

实现dao的接口，通过hibernate连接数据库，接收业务逻辑模块传递的参数，实现数据的增删改查。用事务对数据库操作进行并发操作，如果成功则提交事务，如果失败将回滚事务，保证数据库操作的一致性。

##### 2.3.2模块二：业务逻辑模块

实现engine的接口，调用数据连接层的方法，接收服务层传递的参数，具体实现一个完整的业务功能，具有空值、非法参数的判断，保证每一次提交给数据库的数据的有效性。

##### 2.3.3模块三：服务模块

为外部调用提供RESTful风格的url接口，方便用户进行操作。通过调用业务逻辑层的方法，实现具体的操作功能。

## 3. 功能设计

### 3.1 系统功能模块图

iTreePermissionService

基于角色的权限管理

基于用户的权限管理

图一：模块关系图

### 3.2功能划分

本权限服务程序下设基于角色的权限管理、基于用户的权限管理两大主功能，两种权限管理方式采用同一个权限表。每个功能下设增删改查及匹配等子功能，对外提供RESTful风格的url接口。

用户在使用本权限服务程序时，自由选择基于角色的权限管理或基于用户的权限管理方式。

### 3.3功能描述

#### 3.3.1功能一：基于角色的权限管理

##### 3.3.1.1 说明

使用这种权限管理方式时，当某用户需要某些权限，只需要给这个用户赋予某种角色即可。

##### 3.3.1.2 功能实现

通过对角色ID和角色所对应的权限ID的管理，控制角色的权限。提供用户角色和角色权限的增删改查及权限匹配的接口。

通过对用户ID和用户所对应的角色ID的管理，控制用户的权限，提供增删改查及角色匹配的接口。

#### 3.3.2功能二：基于用户的权限管理

##### 3.3.2.1 说明

这种权限管理方式更为灵活和自由，它采用直接给用户赋权的方式，来管理用户的权限。

##### 3.3.2.2 功能实现

通过对用户ID和用户所对应的权限ID的管理，控制用户的权限，提供增删改查及权限匹配的接口。

## 4．接口设计

### 4.1内部接口

com.itree.dao.api：数据访问层接口，实现类放在包com.itree.dao.impl下。

com.itree.engine.api：业务逻辑层接口，实现类放在包com.itree.engine.impl下。

### 4.2外部接口

#### 4.2.1 接口说明

详见【iTreePermissionAPI1.1】。

#### 4.2.2 调用方式

使用时调用URL :http://localhost:8080/iTreePermissionService+“接口地址”，传入请求参数后使用规定的请求方法即可。

例如要调用增加角色权限的功能，URL为：http://localhost:8080/iTreePermissionService[/rolepermission/post](http://localhost:8080/iTreePermissionService/role/post)，传参（rid，pid）后调用 post 方法。

## 5．数据库设计

### 5.1 说明

本权限服务用hibernate对数据库的jdbc进行了封装，用户只需在本地mysql数据库下建有一个名为itreepermission的数据库，无需自行建表。

为为用户提供更加强大功能的权限管理功能，本数据库数据结构简单，所涉及到的多表操作均封装在了程序源代码中，故数据库中不存在多表关系。

本数据库采用C3p0数据库连接池，实现了连接复用。它通过建立一个数据库连接池以及一套连接使用管理策略，使得一个数据库连接可以得到高效、安全的复用，避免了数据库连接频繁建立、关闭的开销。

### 5.2 资源管理：配置数据库连接池

<!-- 最大连接数 -->

<property name=*"hibernate.c3p0.max\_size"*>20</property>

<!-- 最小连接数 -->

<property name=*"hibernate.c3p0.min\_size"*>5</property>

<!-- 获得连接的超时时间,如果超过这个时间,会抛出异常，单位毫秒 -->

<property name=*"hibernate.c3p0.timeout"*>120</property>

<!-- 最大的PreparedStatement的数量 -->

<property name=*"hibernate.c3p0.max\_statements"*>100</property>

<!-- 每隔120秒检查连接池里的空闲连接 ，单位是秒 -->

<property name=*"hibernate.c3p0.idle\_test\_period"*>120</property>

<!-- 当连接池里面的连接用完的时候，C3P0一下获取的新的连接数 -->

<property name=*"hibernate.c3p0.acquire\_increment"*>2</property>

<!-- 每次都验证连接是否可用 -->

<property name=*"hibernate.c3p0.validate"*>true</property>

### 5.3 数据表设计

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用途 | 存储用户权限 | | | | | |
| 表名 | tb\_user\_permission | | | | | |
| 主键 | id | | | | | |
| 序号 | 字段名 | 字段说明 | 类型 | 位数 | 属性 | 备注 |
| 1 | id | ID | INT | 11 | 非空 | 主键、自增 |
| 2 | uid | 用户ID | INT | 11 | 非空 |  |
| 3 | pid | 权限ID | INT | 11 | 非空 |  |
| 4 | status | 标识是否可用 | INT | 2 | 非空 | 用以标识该实体是否可用。规定：1为可用；0为不可用。默认值为1 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 用途 | 存储角色权限 | | | | | | | | | |
| 表名 | tb\_role\_permission | | | | | | | | | |
| 主键 | id | | | | | | | | | |
| 序号 | 字段名 | 字段说明 | | 类型 | | 位数 | | 属性 | | 备注 |
| 1 | id | ID | INT | | 11 | | 非空 | | 主键、自增 | |
| 2 | rid | 角色ID | INT | | 11 | | 非空 | |  | |
| 3 | pid | 权限ID | INT | | 11 | | 非空 | |  | |
| 4 | status | 标识是否可用 | INT | | 2 | | 非空 | | 用以标识该实体是否可用。规定：1为可用；0为不可用。默认值为1 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用途 | 存储用户角色 | | | | | |
| 表名 | tb\_user\_role | | | | | |
| 主键 | id | | | | | |
| 序号 | 字段名 | 字段说明 | 类型 | 位数 | 属性 | 备注 |
| 1 | id | ID | INT | 11 | 非空 | 主键、自增 |
| 2 | uid | 用户ID | INT | 11 | 非空 |  |
| 3 | rid | 角色ID | INT | 11 | 非空 |  |
| 4 | status | 是否可用标识 | INT | 2 | 非空 | 用以标识该实体是否可用。规定：1为可用；0为不可用。默认值为1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用途 | 存储权限信息 | | | | | |
| 表名 | tb\_permission | | | | | |
| 主键 | id | | | | | |
| 序号 | 字段名 | 字段说明 | 类型 | 位数 | 属性 | 备注 |
| 1 | id | ID | INT | 11 | 非空 | 主键、自增 |
| 2 | name | 用户ID | INT | 11 | 非空 |  |
| 3 | status | 是否可用标识 | INT | 2 | 非空 | 用以标识该实体是否可用。规定：1为可用；0为不可用。默认值为1 |

## 6. 系统安全保密设计

### 6.1 说明

针对本服务要求能够防止各类错误操作可能造成的数据丢失破坏，做出数据安全性设计。

### 6.2 设计

1. 对数据库的操作加入事务管理，在事务运行的过程中发生了某种故障，事务不能继续进行，系统将事务中对数据库的所有以完成的操作全部撤消，回滚到事务开始的状态。
2. 利用mysql数据库的安全机制，只有具有操作数据库权限的用户才可以对数据库进行直接操作。

## 7. 系统性能设计

### 7.1精度

1. 所有的ID均为11位以内的int类型。
2. 所有的字符串长度限制在50位内。

### 7.2 时间特性要求

1. 对数据库进行操作时，返回结果时间应尽量小，速度尽可能快。

（具体限制日后补充）

## 8.系统出错处理

|  |  |  |
| --- | --- | --- |
| 序号 | 错误原因 | 处理方法 |
| 1 | 空值、数据库连接失败及其他 | 根据接口的功能返回null 或者false。  在log4j中记录下出错信息。 |