**新版IPTV数据模块概要设计方案  
V1.1**

## 修订记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修订时间 | 修订内容 | 作者 | 说明 |
| V1.0 | 2017/01/17 | 初稿 | 文腙 | 待评审 |
| V1.1 | 2017/01/24 | 初稿 | 文腙 | 新增备份数据设计 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 说明

新版本IPTV数据模块主要基于老版本基础上扩展，增强数据的流向管理，来源管理，增加数据备份，权限控制等安全措施，旨在提高数据的安全性，可扩展性以及代码的可维护性。

# 总体设计

## 需求规定

### 扩展性需求

IPTV内部存储数据，应有私有文件以存储数据，保证数据操作的安全性，其他进程不可随意操作私有数据文件。私有文件可以是基于SharedPerference的轻量级配置文件，也可以是基于SQLite的轻量级数据库。两者根据实际需要二选一，工程里两者都实现，提供接口以选择。

对于IPTV与系统共用数据，以保存系统数据库为宜，尽量减小系统对接的工作量。在存储系统数据库时，同一字段根据不同厂家、不同芯片可能会有不同的字段名或表名，所以数据模块应该提供一个方便修改，扩展性高的接口，灵活地读写系统数据库。

### 安全性需求

为了保证数据的安全性，对外提供的数据接口时应进行权限控制，以及备份数据的管理。

在ContentProvider以及Aidl中都可以通过自定义权限以达到权限控制的目的，其中Aidl还可以拿到调用进程的包名，还可以进行包名权限控制。

备份管理应有一个备份数据库为宜，备份数据应该有多维度的特性，尽可能包含多的信息，比如上次修改的进程包名，上次修改前的值，上次修改后的值，修改时间等等。

### 其他需求

模块应该提供一个数据修改的回调通知功能，某些场景下，其他进程需要实时监听某些业务数据的修改状态，比如网管需要监听部分业务的数据修改，设置APK需要监听部分数据的修改以实时更新UI等。

模块的对外接口应该支持多种，比如aidl，contentProvider等,并需要支持现网市场的所有方式，以达到能直接对接现网盒子的要求。

## 模块结构设计

### 总体模块结构



数据模块有一个总的入口文件AmtDataManager，入口文件管理数据的去向，来源，备份以及数据更新的通知，对内提供操作接口。

* **数据去向管理：**

通过DataFactory提供多种不同的数据存储方式的实现类，比如SharedPreference文件，数据库方式等。

部分数据，如PPPOE/IPOE账号密码等网络数据，需要与系统共用，则直接明文存储系统数据库，可灵活配置存储在secure表或者system表，以及数据的字段名，并向备份数据库备份加密后的数据。

* **数据来源管理：**

通过记录修改者包名的方式记录数据来源。以AIDl方式对接的数据模块，可以获取调用者包名，并作为数据来源的依据。以ContentProvider方式对接的数据模块，不提供修改接口。

若以数据库方式存储数据，则可以存储数据的修改源。

* **数据更新通知**

提供注册监听者接口，管理监听者，并且过滤掉数据更新的发起者，对其他进程进行通知。此功能需要保证数据修改的唯一入口为AmtDataManager。但某些情况下，某些数据的通知并不准确。如PPPOE/IPOE账号密码等与系统共用的数据，若系统自行更新了数据，则IPTV检测不到，便无法通知。设置APK、网管零配置等流程，应该都通过AmtDataManager来保存数据。

### 数据备份

IPTV创建一个单独的私有备份数据库，用于存放备份数据。备份数据库表名为iptv\_bake。备份数据库存放本次更新前的值，更新后的值，本次更新的日期，上次更新的日期，发起更新的进程包名等。当IPTV业务数据被污染时，可根据备份数据获取到被修改前的值，以恢复数据，提高数据的安全性。

*IPTV备份数据表结构：*



数据字典：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| id | Int | M | 主键 |
| key | String | M | 数据节点名。 |
| value | String | O | 本次更新的值 |
| modify\_date | String | O | 本次更新日期 |
| mender | String | O | 本次更新的进程包名 |
| last\_value | String | O | 上次更新的值（本次更新前的值） |
| last\_modify\_date | String | O | 上次更新的日期 |
| last\_mender | String | O | 上次更新的包名 |

注：若连续重复更新同一字段，则备份模块不更新对应数据的last\_value、last\_modify\_date、last\_mender字段，只更新modify\_date、mender字段，保证在有效更新数据时记录有效备份信息。

### 扩展性设计

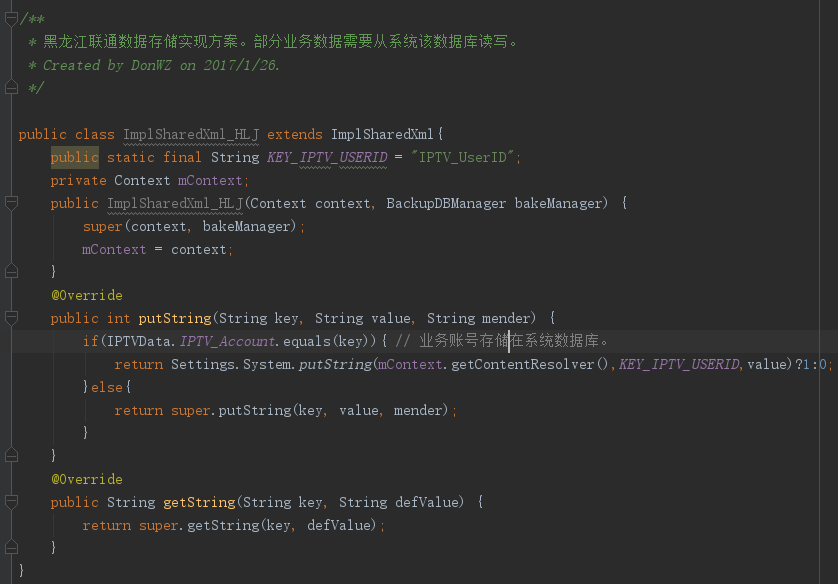
数据模块业务耦合度高，某些项目下数据存储有特殊需求，约定制定的数据存储到指定的数据库或文件等，就需要提高到本模块的可扩展性。

本模块默认实现两套数据存储方案，分别为 ImplSharedXml和ImplDataDB两套，第一个将数据全部存储在SharedPreference文件中，另一个将数据全部存储在IPTV私有数据库中。若后期项目中需要一小部分数据从约定的第三方文件中读写，可新建数据存储方案，继承ImplSharedXml或者ImplDataDB其中一个方案，重写数据读写接口，针对指定数据做特殊业务处理。

本文以黑龙江联通作为示例。

黑龙江局点约定IPTV业务账号、密码、PPPOE账号密码、IPOE账号密码和主备认证地址都从系统数据库system表读写。那么IPTV其他的私有数据没有做约定，则其他数据可以按照IPTV自己的规则读写，目前黑龙江是以SharedPreference方式存储私有数据的。

首先新建黑龙江数据存储方案，继承ImplSharedXml，重写putString和getString接口，根据入参key判断是否是需要特殊处理的数据，否则调用super.putString 和super.getString 执行父类的默认方案。如图：



## 数据字典

* SharedPreference方式

由于该方式的局限性，数据只能以Key-Value的形式存储。此处不再详细说明。数据类型和key应该和老版本的节点保持一致。

* SQLIte数据库方式

数据全部以String类型存储。数据库表包含四个字段，详情见表：

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| id | Integer | 数据id，主键。 |
| ikey | String | 数据的key值。不可重复，不可为空 |
| ivalue | String | 数据的value值。 |
| description | String | 数据描述。 |

# 接口设计

## IPTV内部接口

内部接口提供者为AmtDataManger，提供所有基本类型数据的读写接口。

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 返回值 | 参数 | 说明 |
| getString | String | Key, defValue | 获取String数据 |
| setString | Int 0:失败 1:成功 | key | 保存数据 |
| value |
| from:数据来源 |
| setStringBatch | Int 0:失败 1:成功 | ContentValues | 批量导入String数据。此接口只能IPTV调用。不对外提供 |
| setDataCallback | void | callingPck：注册者包名 | 注册数据变化监听器 |
| callback:监听器 |
| removeDataCallBack | void | callingPck：注册者包名 | 移除数据监听器 |
| … |  |  |  |
|  |  |  |  |

## 外部接口

外部接口有三个实现方式，分别为新版aidl接口，老版本aidl接口，符合联通规范的ContentProvider接口。

* **老版本aidl接口：**

新版本数据模块了为了能兼容老版本设置、网管、升级等其他模块，故需要兼容老版本aidl接口。此处不再详细说明。

* **新版本aidl接口：**

老版本的aidl接口功能有限，为了实现数据更新回调通知的功能，新增了一个新版本的aidl接口，详情如图：

package com.amt.amtdata;  
  
import com.amt.amtdata.IDataCallBack;  
  
interface IAmtDataAidl {  
 int putString(String key,String value);  
 String getString(String key,String defValue);  
 void registDataCallBack(IDataCallBack callback);  
 void unRegistDataCallBack();  
}

主aidl文件为IAmtDataAidl，其中包含一个回调接口IDataCallBack,如图：

package com.amt.amtdata;  
  
interface IDataCallBack {  
 void dataChanged(String key,String oldValue,String newValue);  
}

新版aidl接口精简为String的读写操作，若对现网已有的其他类型进行读写，本模块将转成String类型传递。

新版本aidl封装了一个快速对接的jar包，集成方引入jar包即可完成快速对接。

* **ContentProvider接口：**

此接口为符合联通集团的ContentProvider接口，此处不再详细说明。