**KBMS XML接口**

**目录**

[1用户XML部分 3](#_Toc9808)

[1.1XML结构: 3](#_Toc6866)

[1.2 API接口 4](#_Toc26877)

[1.2.0根据用户ID，建立该用户的XML操作类 4](#_Toc5851)

[1.2.1为用户添加文件夹信息 4](#_Toc12755)

[1.2.2为用户添加文件信息 4](#_Toc27175)

[1.2.3修改用户标签 5](#_Toc163)

[1.2.4添加用户日志 5](#_Toc11574)

[1.2.5获得用户标签 6](#_Toc3101)

[1.2.6获得用户日志列表 6](#_Toc25464)

[1.3说明 6](#_Toc23397)

[2. 文档部分 6](#_Toc11028)

[2.1 XML结构 6](#_Toc16720)

[2.2 API接口 7](#_Toc11181)

[2.2.1根据文档Id建立docXml操作类 7](#_Toc28310)

[2.2.2建立新的Xml文件 7](#_Toc17487)

[2.2.3写入文档标签 7](#_Toc14966)

[2.2.4获得文档标签 8](#_Toc14846)

[2.2.5 写入相似文档节点 8](#_Toc32161)

[2.2.6获得该文件的相似文档内容 8](#_Toc29556)

[2.3说明 8](#_Toc31088)

# 

# 1用户XML部分

## 1.1XML结构:

<user-body> //XML根节点

<root-folder> //用户目录节点（包含文件夹以及文件节点）

<folder> //文件夹节点（包含文件夹以及文件节点）

<id></id>

<name></name>

</folder>

<file> //文件节点

<id></id>

<name></name>

<size></size>

<time></time>

</file>

</root-folder>

<user-log> //用户日志节点（包含多个日志信息节点）

<log> //日志信息节点

<time></time>

<value></value>

<parameter></parameter>

<note></note>

</log>

</user-log>

<user-tags> //用户标签节点（包含多个标签节点）

<tag></tag> //标签节点

</user-tags>

</user-body>

## 1.2 API接口

**包含的class：**

KbmsXml.class //XML文件操作子类，负责XML读取保存等基本操作

KbmsFolderXml.class //文件夹节点映射类

KbmsLogXml.class //日志节点映射类

KbmsTagsXml.class //XML标签映射容器

KbmsUserXml.class //用户XML操作类（重要），继承KbmsXml

**功能实现**

### 1.2.0根据用户ID，建立该用户的XML操作类

String userXmlPath = this.getUserXMLBasePath() + userId + “.xml”

KbmsUserXml kbmsUserXml = new KbmsUserXml(userXmlPath);

[说明]:这里的getUserXMLBasePath是手写一个获得用户XML绝对路径的函数，具体请见ControllerBasePath，实际上，所有Controller操作都是基于这个子类的。

### 1.2.1为用户添加文件夹信息

申明：public void addFolderById(List<String> buildRoute, KbmsFolderXml folder);

用法:kbmsUserXml.addFolderById(buildRoute, folder);

buildRoute为添加到的路径（这里是多个文件夹的ID）

Folder为添加的文件夹映射信息（具体见KbmsFolderXml的构造函数）

### 1.2.2为用户添加文件信息

申明：public void addDocumentById(List<String> saveRoute, Document doc);

用法：kbmsUserXml.addDocumentById(saveRoute, doc);

用法和(2)相似，这里不做叙述

### 1.2.3修改用户标签

申明：public boolean writeUserTags(KbmsTagsXml tagsXml)

用法：kbmsUserXml.writeUserTags(tagsXml);

tagsXml为Xml内的标签节点映射，具体见构造，里面的list为需要更新的标签

### 1.2.4添加用户日志

申明：

public boolean writeUserLog(KbmsLogXml logElement);

用法：kbmsUserXml.writeUserLog(logElement);

KbmsLogXml为日志节点映射：

public class KbmsLogXml {

private int value; //操作编号

private int parameter; //操作参数

private String note; //日志信息

private Long time; //日志创建时间

private String name; //暂无

}

其中，操作编号和参数的组合有以下几种：

0.用户创建

1.用户登录

2.用户签到

3.上传文件 + File ID

4.收藏文件 + File ID

5.关注用户 + User ID

6.查看文件 + File ID

7.下载文件 + File ID

### 1.2.5获得用户标签

申明：public KbmsTagsXml getUserTags();

用法：kbmsUserXml.getUserTags();

### 1.2.6获得用户日志列表

申明: public List<KbmsLogXml> getUserLog();

用法: kbmsUserXml.getUserLog();

## 1.3说明

用户Xml主要用于存储不好在数据库中操作以及存储的信息，比如用户的目录结构，用户的日志信息，用户的标签等信息，但在数据库中也设计了备份，如果在本地无法正常加载XML，那么在进行操作的时候会自动从数据库进行数据的恢复和备份，每次对Xml的修改都会将新的Xml备份更新到数据库，实现本地与数据库的信息统一。

# 文档部分

## 2.1 XML结构

<file-body>

<file-info></file-info> //文件信息(暂时没用到)

<file-comment></file-comment> //文件评论(暂时没用到)

<file-summary></file-summary> //文件摘要

<!-- 相似文件-->

<similar-files> //相似文件节点（包含多个相似文件元素）

<file> //相似文件元素节点

<id></id>

<name></name>

<hash></hash>

</file>

</similar-files>

<file-tags> //文件标签（包含多个标签节点）

<tag></tag> //标签节点

</file-tags>

</file-body>

## 2.2 API接口

**包含的类：**

KbmsXml.class //xml文件操作基类

KbmsDocXml.class //文档xml操作类，继承了KbmsXml.class

**功能实现：**

### 2.2.1根据文档Id建立docXml操作类

String docXmlPath = this.getFileXMLBasePath() + docId + “.xml”;

KbmsDocXml kbmsDocXml = new KbmsDocXml(docXmlPath);

### 2.2.2建立新的Xml文件

申明：public boolean buildFileXML(Document doc, User user);

用法：kbmsDocXml.buildFileXML(doc, user);

doc为需要建立对映文档的对象(不能为空)

User为用户操作的对象(可以为空)

### 2.2.3写入文档标签

申明：public boolean writeTagsInfo(List<String> tagsList);

用法：kbmsDocXml.writeTagsInfo(tagsList);

tagsList为传入的标签list

注意：这里的写入是覆盖式的写入，传入的数据会将原来的标签数据覆盖。

### 2.2.4获得文档标签

申明：public List<String> getTagsInfo()

用法：kbmsDocXml.getTagsInfo();

这里的返回值为一个list，直接用就可以

### 2.2.5 写入相似文档节点

申明：public boolean writeSimilarInfo(List<Document> doc, List<Integer> hashNum)

用法：kbmsDocXml.writeSimilarInfo(doc, hasNum);

Doc为需要写入的相似文件列表

hashNum为这几个doc和改文件的相似哈希值

注意：doc和hashNum是后台计算出的，需要一一对应

### 2.2.6获得该文件的相似文档内容

申明：public List<DocumentExtra> getSimilarInfo();

用法：kbmsDocXml.getSimilarInfo();

## 2.3说明

文档Xml主要用来存储2个数据，一个是和文档相似的文件映射关系，一个是文档的标签。

备份和同步办法和用户部分一样。