



中华人民共和国国家标准

GB/T 32908—2016

非结构化数据访问接口规范

Unstructured data access interface specification

2016-08-29 发布

2017-03-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会 发布

目 次

前言	III
1 范围	1
2 术语和定义	1
3 缩略语	1
4 查询语言访问接口	1
4.1 概述	1
4.2 基本格式	1
4.3 数据库连接接口	2
4.4 执行查询语句接口	3
4.5 批量执行查询语句接口	4
4.6 结果集接口	7
5 应用程序访问接口	10
5.1 概述	10
5.2 基本格式	10
5.3 集群配置管理接口模块	10
5.3.1 概述	10
5.3.2 集群管理接口	10
5.3.3 单个节点管理接口	11
5.3.4 集群监控接口	12
5.3.5 数据空间管理接口	13
5.4 任务管理接口模块	17
5.4.1 概述	17
5.4.2 任务状态	17
5.4.3 任务接口	17
5.4.4 一致性控制接口	19
5.5 存储接口模块	20
5.5.1 概述	20
5.5.2 底层存储对象类型	20
5.5.3 存储对象定义接口	21
5.5.4 存储对象访问接口	22
5.5.5 非结构化数据访问接口	24
5.6 分析接口模块	33
5.6.1 概述	33
5.6.2 基本数据分析接口	33
5.6.3 时间单位	34
5.6.4 时序数据分析接口	35
5.7 索引接口模块	35

- 5.7.1 概述 35
- 5.7.2 通用向量接口 35
- 5.7.3 特征抽取接口 36
- 5.7.4 通用索引接口 39
- 5.8 搜索接口模块 42
 - 5.8.1 概述 42
 - 5.8.2 媒体数据类型 42
 - 5.8.3 高维数据查询接口 43
 - 5.8.4 关键字查询接口 43
- 6 Web 服务访问接口 44
 - 6.1 概述 44
 - 6.2 基本格式 44
 - 6.3 HTTP 请求格式 44
 - 6.4 HTTP 状态码 45
 - 6.5 服务接口描述 45
 - 6.5.1 数据增加接口 45
 - 6.5.2 数据删除接口 46
 - 6.5.3 数据更新接口 46
 - 6.5.4 数据查询接口 48



前 言

本标准按照 GB/T 1.1—2009 给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本标准起草单位：清华大学、中国电子技术标准化研究院、浙江大学、北京航空航天大学、中国人民大学。

本标准主要起草人：王朝坤、王建民、陈俊、高林、吴新松、刘贤刚、卫凤林、李莹、庄越挺、陈岭、邵健、孙建伶、马殿富、郎波、杜小勇、陈晋川。

非结构化数据访问接口规范

1 范围

本标准规定非结构化数据管理系统的访问接口要求,包括查询语言访问接口、应用程序访问接口和 Web 服务访问接口。

本标准适用于非结构化数据管理系统产品的研制、开发和测试。

2 术语和定义

下列术语和定义适用于本文件。

2.1

非结构化数据 unstructured data

没有明确结构约束的数据,如文本、图像、音频、视频等。

2.2

非结构化数据管理系统 unstructured data management system

对非结构化数据进行管理、操作的大型基础软件,提供非结构化数据存储、特征抽取、索引和查询等管理功能。

3 缩略语

下列缩略语适用于本文件。

HTTP:超文本传输协议(hypertext transfer protocol)

JSON:JavaScript 对象标记(javascript object notation)

XML:可扩展置标语言(extensible markup Language)

4 查询语言访问接口

4.1 概述

本章描述查询语言访问接口,包括数据库连接接口、执行查询语句接口、批量执行查询语句接口和结果集接口。

4.2 基本格式

查询语言访问接口包含以下 5 种元素:

- a) 接口名:接口的名称;
- b) 说明:对接口内容的描述;
- c) 继承:指明了该接口的父接口;
- d) 属性:该接口全部的属性变量;
- e) 方法:该接口提供的所有函数的声明。

4.3 数据库连接接口

接口名:Connection

说明:数据库连接的接口。

继承:无

属性:无

方法:

Statement createStatement()

说明:创建一个 Statement 对象。

参数:无

返回:Statement 对象

PreparedStatement prepareStatement(in String ql)

说明:根据声明语句 ql 创建一个 PreparedStatement 对象。

参数:ql 为查询语句声明,其中允许含多个“?”占位符

返回:PreparedStatement

void commit()

说明:将自上次提交或回滚之后的更新数据提交,并释放当前连接占用的资源。



参数:无

返回:无

void rollback()

说明:撤销自上次提交或回滚之后的更新操作,并释放当前连接占用的资源。

参数:无

返回:无

void rollback(in Savepoint sp)

说明:撤销自保存点 sp 之后的更新操作,并释放当前连接占用的资源。

参数:sp 为保存点实例对象

返回:无

bool isAutoCommit()

说明:获取当前连接是否为自动提交。

参数:无

返回:是否自动提交

void setAutoCommit(in bool auto)

说明:设置当前连接的自动提交属性。

参数:auto 为是否自动提交

返回:无

Savepoint setSavepoint()

说明:设置当前时刻的一个保存点,并获取它。

参数:无

返回:当前时刻的保存点

void releaseSavepoint(in Savepoint sp)

说明:释放保存点 sp 占用的资源。

参数:sp 为保存点实例对象

返回:无

void close()

说明:关闭当前连接。

参数:无

返回:无

bool isClosed()

说明:当前连接是否已关闭。

参数:无

返回:是否关闭

4.4 执行查询语句接口

接口名:Statement

说明:执行查询语句的接口。

继承:无

属性:无

方法:

ResultSet executeQuery(in String ql)

说明:执行数据查询语句。

参数:ql 为数据查询语句

返回:查询结果对象 ResultSet

int executeUpdate(in String ql)

说明:执行数据更新语句,如插入、删除、更新等。

参数:ql 为数据更新语句

返回:此次操作影响的行数

bool execute (in String ql)

说明:执行语句 ql。

参数:ql 为待执行的语句

返回:执行是否成功

Connection getConnection()

说明:获取当前的数据库连接。

参数:无

返回:数据库连接

ResultSet getResultSet()

说明:获取结果集对象 ResultSet,通常在 execute 方法执行后执行此方法。

参数:无

返回:结果集对象

int getFetchSize()

说明:获取从查询结果中提取的缓存的行数。

参数:无

返回:缓存行数

void setFetchSize(in int rows)

说明:设置查询结果缓存行数。

参数:rows 为缓存行数

返回:无

void close()

说明:关闭当前 Statement 对象。

参数:无

返回:无

bool isClosed()

说明:获取当前 Statement 是否已关闭。

参数:无

返回:是否已关闭

4.5 批量执行查询语句接口

接口名:PreparedStatement

说明:执行查询语句的接口,支持批量查询。

继承:Statement

属性:无

方法:

void clearParameters()

说明:清空所有 PreparedStatement 的参数值。

参数:无

返回:无

void setBinaryStream(in int index, in IStream is)

说明:设置语句中第 index 个占位符的值为输入流 is。

参数:index 为占位符的位置,is 为输入流(引用、指针等)

返回:无

void setBlob(in int index, in Blob blob)

说明:设置语句中第 index 个占位符的值为大对象 blob。

参数:index 为占位符的位置,blob 为大对象

返回:无

void setByteArray(in int index, in ByteArray ba)

说明:设置语句中第 index 个占位符的值为字节数组 ba。

参数:index 为占位符的位置,ba 为字节数组

返回:无

void setBool(in int index, in bool b)

说明:设置语句中第 index 个占位符的值为布尔值 b。

参数:index 为占位符的位置,b 为布尔值

返回:无

void setByte(in int index, in byte b)

说明:设置语句中第 index 个占位符的值为单字节数 b。

参数:index 为占位符的位置,b 为单字节数

返回:无

void setDateTime(in int index, in DateTime datetime)

说明:设置语句中第 index 个占位符的值为时间 datetime。

参数:index 为占位符的位置,datetime 为时间

返回:无

void setDate(in int index, in Date date)

说明:设置语句中第 index 个占位符的值为日期 date。

参数:index 为占位符的位置,date 为日期

返回:无

void setDecimal(in int index, in Decimal dec)

说明:设置语句中第 index 个占位符的值为 Decimal 数据。

参数:index 为占位符的位置,dec 为 Decimal 数据

返回:无

void setDouble(in int index, in double d)

说明:设置语句中第 index 个占位符的值为双精度浮点数 d。

参数:index 为占位符的位置,d 为双精度浮点数

返回:无



void setFloat(in int index, in float f)

说明:设置语句中第 index 个占位符的值为单精度浮点数 f。

参数:index 为占位符的位置,f 为单精度浮点数

返回:无

void setInt(in int index, in int i)

说明:设置语句中第 index 个占位符的值为整型数 i。

参数:index 为占位符的位置,i 为整型数

返回:无

void setLong(in int index, in long l)

说明:设置语句中第 index 个占位符的值为长整型数 l。

参数:index 为占位符的位置,l 为长整型数

返回:无

void setString(in int index, in String str)

说明:设置语句中第 index 个占位符的值为字符串 str。

参数:index 为占位符的位置,str 为字符串

返回:无

void setURI(in int index, in URI uri)

说明:设置语句中第 index 个占位符的值为 URI 类型数据。

参数:index 为占位符的位置,uri 为 URI 类型数据

返回:无

void setVector(in int index, in IVectorObject vec)

说明:设置语句中第 index 个占位符的值为向量类型数据。

参数:index 为占位符的位置,vec 为向量类型数据

返回:无

ResultSet executeQuery()

说明:执行 PreparedStatement 对象包含的 ql 查询语句,此处的 PreparedStatement 执行的是查询操作,不涉及数据的更改。

参数:无

返回:数据查询执行结果的 ResultSet 对象

int executeUpdate()

说明:执行 PreparedStatement 对象包含的 ql 数据更新语句,此处的 PreparedStatement 执行的是数据的更新操作。

参数:无

返回:此次操作关联的行数

4.6 结果集接口

接口名:ResultSet

说明:查询结果集接口,是查询方法返回的结果的统一表示。

继承:无

属性:无

方法:

`bool absolute(in int rows)`

说明:将指针移到第 rows 行,从第 rows 行开始读取数据。

参数:rows 为开始读取的行序号

返回:指针移动是否成功

`bool beforeFirst()`

说明:将指针移到第 1 行数据之前,即 ResultSet 数据最起始的地方。

参数:无

返回:指针移动是否成功

`bool afterLast()`

说明:将指针移到第末行数据之后,即 ResultSet 数据最末尾的地方。

参数:无

返回:指针移动是否成功

`bool next()`

说明:移动指针到当前行的下一行。

参数:无

返回:移动是否成功

`bool previous()`

说明:移动指针到当前行的前一行。

参数:无

返回:移动是否成功

`IStream getBinaryStream(in String key)`

说明:从 key 对应的位置获取输入流对象。

参数:key 为属性的键

返回:输入流对象(引用、指针等)

`Blob getBlob(in String key)`

说明:从 key 对应的位置获取大对象。

参数:key 为属性的键

返回:大对象

ByteArray getByteArray(in String key)

说明:从 key 对应的位置获取字节数组。

参数:key 为属性的键

返回:字节数组

bool getBool(in String key)

说明:从 key 对应的位置获取布尔值。

参数:key 为属性的键

返回:布尔值

byte getByte(in String key)

说明:从 key 对应的位置获取单字节值。

参数:key 为属性的键

返回:单字节值

DateTime getDateTime(in String key)

说明:从 key 对应的位置获取时间数据。

参数:key 为属性的键

返回:时间数据



Date getDate(in String key)

说明:从 key 对应的位置获取日期数据。

参数:key 为属性的键

返回:日期数据

Decimal getDecimal(in String key)

说明:从 key 对应的位置获取 Decimal 数据。

参数:key 为属性的键

返回:Decimal 数据

double getDouble(in String key)

说明:从 key 对应的位置获取双精度浮点数值。

参数:key 为属性的键

返回:双精度浮点数值

float getFloat(in String key)

说明:从 key 对应的位置获取单精度浮点数值。

参数:key 为属性的键

返回:单精度浮点数值

int getInt(in String key)

说明:从 key 对应的位置获取整型数值。

参数:key 为属性的键

返回:整型数值

long getLong(in String key)

说明:从 key 对应的位置获取长整型值。

参数:key 为属性的键

返回:长整型值

String getString(in String key)

说明:从 key 对应的位置获取字符串。

参数:key 为属性的键

返回:字符串



URI getURI(in String key)

说明:从 key 对应的位置获取 URI 对象。

参数:key 为属性的键

返回:URI 对象

IVectorObject getVector(in String key)

说明:从 key 对应的位置获取向量对象。

参数:key 为属性的键

返回:向量对象

int getFetchSize()

说明:获取从查询结果中提取的缓存的行数。

参数:无

返回:缓存行数

void setFetchSize(in int rows)

说明:设置查询结果缓存行数。

参数:rows 为缓存行数

返回:无

Statement getStatement()

说明:获取当前 ResultSet 对象对应的 Statement 对象。

参数:无

返回:Statement 对象

void close()

说明:释放当前 ResultSet 对象的资源。

参数:无

返回:无

bool isClose()

说明:查看当前 ResultSet 对象的资源是否已经释放。

参数:无

返回:资源是否已释放

5 应用程序访问接口

5.1 概述

本章描述非结构化数据管理系统各个模块中的访问接口,本章包含集群配置管理接口模块、任务管理接口模块、存储接口模块、分析接口模块、索引接口模块和搜索接口模块。

5.2 基本格式

应用程序访问接口包含了以下 5 种元素:

- a) 接口名:接口的名称,符号“:.”之前的为模块名,模块名是按功能类型对接口进行分组后的组名;
- b) 说明:对接口内容的描述;
- c) 继承:指明了该接口的父接口;
- d) 属性:包含了该接口全部的属性变量;
- e) 方法:包含了该接口提供的全部功能函数的声明,是接口的核心内容。

5.3 集群配置管理接口模块

5.3.1 概述

模块名:udms_management

说明:与集群配置管理相关的模块,包括了访问、控制集群,及其配置信息的所有接口。该模块的使用对象为分布式非结构化数据访问系统,而对集中式系统而言,无需考虑该模块。

5.3.2 集群管理接口

接口名:udms_management::IClustersManagement

说明:控制整个集群的接口,包括集群的启动、关闭、查看等。

继承:无

属性:无

方法:

void load()

说明:导入集群节点信息,节点信息存储在缺省的路径下。

参数:无

返回:无

void startup()

说明:启动集群,开始工作。

参数:无

返回:无

void shutdown()

说明:关闭集群,停止服务。

参数:无

返回:无

MapList getClustersInfo()

说明:查看集群各节点的信息。

参数:无

返回:集群节点信息列表

Map getNodeInfo(in int nodeId)

说明:查看单个节点信息。

参数:nodeId 为节点的 id

返回:节点信息列表

int addNode(in Node node)

说明:向集群中动态添加新的节点。

参数:node 为待添加的节点对象

返回:新添加节点在集群中的 id

bool removeNode(in int nodeId)

说明:删除集群中的某个节点。

参数:nodeId 指定待删除的节点

返回:节点删除是否成功

bool pingAlive(in String hostaddr)

说明:查看某节点是否在正常工作。

参数:hostaddr 为待检测的节点的地址

返回:节点是否工作正常

5.3.3 单个节点管理接口

接口名:udms_management::INodeManagement

说明:控制集群中单个节点的接口,如单个节点的启动、关闭、修复等。

继承:无

属性:

名称:node

定义:public Node node

说明:当前节点的对象。

方法:

bool commission()

说明:启动或重新启动当前节点的对外服务。

参数:无

返回:是否成功

bool decommission()

说明:解除当前节点的对外服务。

参数:无

返回:是否成功

bool shutdown()

说明:关闭当前节点。

参数:无

返回:是否成功

bool repair()

说明:对节点上备份的数据进行错误恢复,保证节点持有最新版本的数据,以确保一致性。

参数:无

返回:是否成功

bool flush()

说明:停止接受写操作,将全部缓存的数据写入磁盘。

参数:无

返回:是否成功

5.3.4 集群监控接口

接口名:udms_management::IClustersMonitor

说明:监控集群运行情况的接口。

继承:无

属性:无

方法:

JSON monitorHardwareRes ()

说明:监控集群的硬件资源的使用情况。

参数:无

返回:以 JSON 封装的集群硬件资源信息

JSON monitorClustersConditions ()

说明:监控集群的运行时参数。

参数:无

返回:以 JSON 封装的集群运行时参数的信息

void writeToLog ()

说明:将监控数据写入日志。

参数:无

返回:无

void cleanup ()

说明:清理监控数据日志。

参数:无

返回:无

5.3.5 数据空间管理接口

5.3.5.1 表空间管理接口

接口名:udms_management;:IKeyspaceManagement

说明:管理表空间的接口。此处的表空间对应于关系型数据库中的数据库实例。

继承:无

属性:无

方法:

List getKeyspaceNameList()

说明:获取表空间的名称列表。

参数:无

返回:表空间的名称列表

Map getKeyspaceProperties(in String ksName)

说明:获取表空间的配置参数和运行参数。

参数:ksName 为表空间的名称

返回:表空间的配置参数和运行参数

bool createKeyspace(in String ksName, in int backup, in BackupStrategy bks, in String options)

说明:创建新的表空间。

参数:ksName 为新的表空间的名称, backup 为数据备份数, bks 为数据备份的策略, options 为备份选项

返回:新的表空间是否创建成功

bool dropKeyspace(in String ksName)

说明:删除已有的表空间。

参数:ksName 为待删除的表空间的名称

返回:表空间是否删除成功

bool updateKeyspace(in String ksName, in int backup, in BackupStrategy bks, in String options)

说明:更新表空间的信息。

参数:ksName 为表空间的名称, backup 为数据备份数, bks 为数据备份的策略, options

为备份选项

返回:表空间是否更新成功

`bool useKeyspace(in String ksName)`

说明:使用指定的表空间。

参数:ksName 为待使用的表空间的名称

返回:表空间是否使用成功

5.3.5.2 表管理接口

接口名:udms_management::ITableManagement

说明:管理表各项操作的接口,如创建、更新、删除、查询表等。

继承:无

属性:无

名称:kspace

定义:public KeySpace kspace

说明:当前所处的表空间的名称。

方法:

List getTableNameList()

说明:获取当前表空间内所有表的名称列表。

参数:无

返回:表的名称列表

Map getTableProperties(in String tableName)

说明:获取表的配置参数、运行参数和索引信息。

参数:tableName 为表的名称

返回:表的配置参数、运行参数和索引信息

`bool createTable(in String tableName, optional in TableType ttype,
optional in ComparatorType ctype,
optional in float rowCache,
optional in int rowTTL,
optional in float keyCache,
optional in int keyTTL,
optional in float readRepairFreq,
optional in int recycleFreq,
optional in ComparatorType vtype,
optional in int minCompactTime,
optional in int maxCompactTime)`


说明:创建新的表。

参数:tableName 为新的表的名称,其他可选参数包括:

TableType ttype 为表的类型;

ComparatorType ctype 为比较器类型;

float rowCache 为行缓存大小；
 int rowTTL 为行缓存保存时间；
 float keyCache 为关键字缓存大小；
 int keyTTL 为关键字缓存保存时间；
 float readRepairFreq 为读修复几率；
 int recycleFreq 为垃圾收集频率；
 ComparatorType vtype 为验证器类型；
 int minCompactTime 为最小压缩时间；
 int maxCompactTime 为最大压缩时间

 返回:新的表是否创建成功

```
bool updateTable(in String tableName, optional in TableType ttype,
                 optional in ComparatorType ctype,
                 optional in float rowCache,
                 optional in int rowTTL,
                 optional in float keyCache,
                 optional in int keyTTL,
                 optional in float readRepairFreq,
                 optional in int recycleFreq,
                 optional in ComparatorType vtype,
                 optional in int minCompactTime,
                 optional in int maxCompactTime)
```

说明:更新表参数。

参数:tableName 为表的名称,其他可选参数包括:

TableType ttype 为表的类型；
 ComparatorType ctype 为比较器类型；
 float rowCache 为行缓存大小；
 int rowTTL 为行缓存保存时间；
 float keyCache 为关键字缓存大小；
 int keyTTL 为关键字缓存保存时间；
 float readRepairFreq 为读修复几率；
 int recycleFreq 为垃圾收集频率；
 ComparatorType vtype 为验证器类型；
 int minCompactTime 为最小压缩时间；
 int maxCompactTime 为最大压缩时间

返回:表是否更新成功

```
bool dropTable(in String tableName)
```

说明:删除已有的表。

参数:tableName 为待删除的表的名称

返回:表是否删除成功

bool emptyTable(in String tableName)

说明:清空指定的表。

参数:tableName 为待清空的表的名称

返回:表是否清空成功

5.3.5.3 列管理接口

接口名: udms_management::IColumnManagement

说明:管理列各项操作的接口,如增加、更新、删除、查询等。

继承:无

属性:

名称:table

定义:public Table table

说明:当前列所在的表的对象。

方法:

List getColumnNameList()

说明:获取当前表内所有列的名称列表。

参数:无

返回:列的名称列表

Map getColumnProperties(in String colName)

说明:获取列的信息。

参数:colName 为列的名称

返回:列的信息

bool createColumn(in String colName, in ComparatorType ctype)

说明:添加新的列。

参数:colName 为新的列的名称, ctype 为比较器类型

返回:新的列是否添加成功

bool dropColumn(in String colName)

说明:删除一个列。

参数:colName 为待删除的列的名称

返回:列是否删除成功

bool updateColumn(in String colName, in ComparatorType ctype)

说明:更新列。

参数:colName 为待更新的列的名称, ctype 为比较器类型

返回:是否更新成功

5.4 任务管理接口模块

5.4.1 概述

模块名:udms_tasks

说明:定义任务管理基本要素及操作的模块。

5.4.2 任务状态

任务状态名:udms_tasks::TaskStatus

类型:枚举类型,取值可为:

Preparing,任务准备中,尚不能被执行;

Prepared,任务准备完毕,待执行;

InAction,正在执行中的任务;

Completed,任务已执行完毕;

Aborted,任务被终止。

5.4.3 任务接口

接口名:udms_tasks::ITask

说明:任务在系统中的统一表示。

继承:无

属性:

名称:tid

定义:public String tid

说明:任务的标示符。

名称:title

定义:public String title

说明:任务的名称。

名称:desc

定义:public String desc

说明:任务的描述信息。

名称:timestamp

定义:public long timestamp

说明:任务最近一次被更新的时间戳。

名称:status

定义:public TaskStatus status

说明:任务的状态。

名称:priority

定义:public int priority

说明:任务的优先级。

名称:parent

定义:public ITask parent

说明:当前任务的父级任务。

名称:children

定义:protected ITaskList children

说明:当前任务的子级任务列表。

方法:

void schedule()

说明:调度该任务,并递归调度其子任务。

参数:无

返回:无

void execute()

说明:执行当前任务,并递归执行其子任务。

参数:无

返回:无

void abort()

说明:终止并放弃当前任务。

参数:无

返回:无

void cleanup()

说明:清空子任务列表中所有状态为 Completed 的子任务。

参数:无

返回:无

void addChild(in ITask child)

说明:添加一个子任务。

参数:child 为子任务

返回:无



void addChild(in ITask child, int position)

说明:在指定位置添加一个子任务。

参数:child 为子任务,position 为待添加的任务在子任务列表中的位置

返回:无

void removeChild(in ITask child)

说明:删除一个子任务。

参数:child 为子任务

返回:无

void removeChild(int position)

说明:删除指定位置的一个子任务。

参数:position 为待删除的任务在子任务列表中的位置

返回:无

5.4.4 一致性控制接口

5.4.4.1 协调器接口

接口名:udms_tasks::ICoordination

说明:提供统一的对数据和日志的读写操作。

继承:无

属性:

名称:controller

定义:public IVersionControl controller

说明:版本控制器对象。

方法:

bool write(in IStorageID id, in any data)

说明:对指定的对象执行写操作。

参数:id 为存储对象的唯一标示符,data 为待写入的数据

返回:写操作是否成功

any read(in IStorageID id)

说明:对指定的对象执行读操作。

参数:id 为存储对象的唯一标示符

返回:最新版本的数据

List getLogNodeList(in IStorageID id)

说明:根据数据备份规则计算出给定标示符对应的数据的多个日志存储器。

参数:id 为存储对象的唯一标示符

返回:日志存储器列表

List getDataNodeList(in IStorageID id)

说明:根据数据备份规则计算出给定标示符对应的数据的多个数据存储器。

参数:id 为存储对象的唯一标示符

返回:数据存储器列表

void writeToLogNode(in IStorageID id, in any data, in List nodelist)

说明:将日志和数据写入多个日志存储器中。

参数: id 为存储对象的唯一标示符, data 为待写入的数据, nodelist 为日志存储器列表
返回: 无

void writeToDataNode(in IStorageID id, in any data, in List nodelist,
in LogPosition lp)

说明: 将数据和日志位置写入到 nodelist 指定的多个数据存储器中。

参数: id 为存储对象的唯一标示符, data 为待写入的数据, nodelist 为数据存储器列表, lp
为日志写入位置

返回: 无

5.4.4.2 版本控制器接口

接口名: udms_tasks::IVersionControl

说明: 版本控制器接口主要负责读写数据的版本, 以位置 Position 表示数据的版本, Position 越高
表示版本越新, Position 的实现策略不限。

继承: 无

属性: 无

方法:

Position get(in IStorageID id)

说明: 获取指定对象的最新的日志位置。

参数: id 为存储对象的唯一标示符

返回: 最新的日志位置

void update(in IStorageID id, Position pos)

说明: 更新指定的日志写入位置。

参数: id 为存储对象的唯一标示符, pos 为用以更新的日志位置

返回: 无

5.5 存储接口模块

5.5.1 概述

模块名: udms_store

说明: 定义存储对象类型及存储操作的模块。



5.5.2 底层存储对象类型

对象名: udms_store::StorageType

类型: 枚举类型, 取值可为:

FileID, 唯一标识存储系统中的一个文件对象;

FolderID, 唯一标识存储系统中的一个文件夹对象;

TableID, 唯一标识存储系统中得一个表对象;

RowID, 唯一标识存储系统中表的一行;

ColumnID, 唯一标识存储系统中表的一列。

5.5.3 存储对象定义接口

接口名:udms_store::IStorageID

说明:存储对象在系统中的统一表示。

继承:无

属性:无

方法:

IStorageID constructFileID(in String path)

说明:定义一个 FileID 对象。

参数:path 为文件的存储路径

返回:底层存储对象 ID

IStorageID constructFolderID (in String path)

说明:定义一个 FolderID 对象。

参数:path 为文件夹的存储路径

返回:底层存储对象 ID

IStorageID constructTableID(in String tableName, in String keySpace)

说明:定义一个 TableID 对象,标识一个表。

参数:tableName 为表的名称,keySpace 表示表空间名称

返回:底层存储对象 ID

IStorageID constructRowID(in String rowKey, in String keySpace)

说明:定义一个 RowID 对象,标识表的一行。

参数:rowKey 为行键,keySpace 为表空间的名称

返回:底层存储对象 ID

IStorageID constructColumnID(in String colName, in String keySpace, in String rowKey)

说明:构造一个 ColumnID 对象,标识表空间中的一列。

参数:colName 为列的名称,keySpace 为表空间的名称,rowKey 为行键

返回:底层存储对象 ID

StorageType getIDType()

说明:获得当前 IStorageID 的类型。

参数:无

返回:底层存储对象的类型

String getFilePath()

说明:获得文件路径,只有 FileID 类型的 IStorageID 才能够访问,其他类型的 IStorageID 访问会抛出异常。

参数:无

返回:文件路径

String getFolderPath()

说明:获得文件夹路径,只有 FolderID 类型的 IStorageID 才能够访问,其他类型的 IStorageID 访问会抛出异常。

参数:无

返回:文件夹路径

String getTableName()

说明:返回表的名称,只有 TableID,ColumnFamilyID,RowID,ColumnID 类型的 IStorageID 才能够访问,其他类型的 IStorageID 访问会抛出异常。

参数:无

返回:表名称

String getColumnName()

说明:返回列的名称,只有 ColumnID 类型的 IStorageID 才能够访问,其他类型的 IStorageID 访问会抛出异常。

参数:无

返回:列名称

String getRowKey()

说明:返回一个行的 ID,只有 ColumnFamilyID,ColumnID,RowID 类型的 IStorageID 才能够访问,其他类型的 IStorageID 访问会抛出异常。

参数:无

返回:行 ID

5.5.4 存储对象访问接口

接口名:udms_store::IStorageAccess

说明:定义了访问存储对象的基本操作。

继承:无

属性:无

方法:

bool create(in IStorageID id)

说明:创建一个存储对象,该方法仅支持对文件夹 FolderID 和表 TableID 的创建。

参数:id 为存储对象定义的 ID

返回:创建是否成功

bool remove(in IStorageID id)

说明:删除一个存储对象。

参数:id 为存储对象定义的 ID

返回:删除是否成功

bool remove(in IStorageIDList idList)

说明:批量删除文件。

参数: idList 表示存储对象的列表

返回: 无

void put(in IStorageID id, in String localpath)

说明: 上传本地文件或文件夹。

参数: localpath 为本地文件或文件夹的路径, id 为存储对象

返回: 无

void put(in IStorageID id, in ByteArray values, in bool append)

说明: 使用字节数组上传文件或列, 如果 id 表示上传的列, 则应保证其对应的 keySpace、Column Family 存在, 否则会抛出 StorageException 异常, 上传数据大小不能超过文件块的上限, 文件块大小在配置文件中指定。

参数: value 为上传的列的内容, 或文件内容, id 为使用 StorageID 包装的远程文件系统的路径, append 为布尔类型, 当 append 为 true 时, 数据添加在已有的列或文件的末尾, 不删除已有的列或文件的数据; 当 append 为 false 时, 添加的数据将覆盖已有的列或文件

返回: 无

void put(in IStorageIDList idList, in ByteArrayList list, in bool append)

说明: 批量使用字节数组上传文件或列, 如果 idList 表示上传的列的列表, 则应保证其各自对应的 keySpace, Column Family 存在, 否则会抛出 StorageException 异常, 上传数据大小不能超过文件块的上限, 文件块大小在配置文件中指定。

参数: idList 表示远程文件系统对应的对象的列表, list 为文件字节数组列表, append 为布尔类型, 当 append 为 true 时, 数据添加在已有的列或文件的末尾, 不删除已有的列或文件的数据; 当 append 为 false 时, 添加的数据将覆盖已有的列或文件

返回: 无

void get(in IStorageID id, in String localpath)

说明: 下载文件或文件夹到本地。

参数: id 表示文件或文件夹的存储对象, localpath 表示本地路径

返回: 无

ByteArray get(in IStorageID id)

说明: 获取列内容的字节数组。

参数: id 表示列的存储对象

返回: 列的字节数组

ByteArrayList get(in IStorageIDList idList)

说明: 批量获取列内容的字节数组。

参数: idList 表示列存储对象的列表

返回: 含多个列对象字节数组的列表

UploadStream putStream(in IStorageID id, in bool append)

说明:以数据流的方式上传列或文件,UploadStream 是数据输出流的子类,提供以流的形式上传列和文件的方法。如果 id 为列对象,则调用该方法时,需保证 id 对应的 keySpace 和 Column Family 存在,否则会抛出 StorageException 异常。文件或列在上传结束后执行 flush 方法将本地缓冲中的数据发送到服务器,如果上传的文件占用的内存比较大,在上传结束后需执行 close 方法,释放占用的内存资源。

参数:id 为待上传的列或文件对象,append 为布尔类型,当 append 为 true 时,数据添加在已有的列或文件的末尾,不删除已有的列或文件的数据;当 append 为 false 时,添加的数据将覆盖已有的列或文件

返回:上传的输出流对象

DownloadStream getStream(in IStorageID id)

说明:以流的方式下载列或文件,若下载的是列数据,则需保证 id 对应的 keySpace, Column Family 存在,否则会抛出 StorageException 异常。DownloadStream 为数据输入流的子类,提供以流的形式下载列和文件的方法。

参数:id 表示待下载的列或文件对象

返回:下载的输出流对象

5.5.5 非结构化数据访问接口

5.5.5.1 图像存储与访问接口

接口名:udms_store::IImageAccess

说明:存储及访问图像的接口。在系统中,图像对应于 Image 类。

继承:无

属性:无

方法:

void put (in IStorageID id, in Image img)

说明:上传图像数据。

参数:id 表示图像的存储对象,img 为图像的实例对象

返回:无

void putAttribute (in IStorageID id, in String key, in any data)

说明:上传图像的元数据。

参数:id 表示图像的存储对象,key 为与 id 关联的图像的属性键,data 为属性值

返回:无

void putFeature (in IStorageID id, in String key, in any data)

说明:上传图像的特征数据。

参数:id 表示图像的存储对象,key 为与 id 关联的图像的特征的键,data 为特征数据

返回:无

Image get (in IStorageID id)

说明:获取指定的图像的实例对象。

参数:id 表示图像的存储对象

返回:图像实例

any getAttribute (in IStorageID id, in String key)

说明:获取指定的图像的属性值。

参数:id 表示图像的存储对象,key 为与 id 关联的图像的属性键

返回:属性值

any getFeature (in IStorageID id, in String key)

说明:获取指定的图像的特征数据。

参数:id 表示图像的存储对象,key 为与 id 关联的图像的特征的键

返回:特征数据

bool remove (in IStorageID id)

说明:删除指定的图像数据。

参数:id 表示图像的存储对象

返回:是否成功

bool removeAttribute (in IStorageID id, in String key)

说明:删除指定的图像的属性。

参数:id 表示图像的存储对象,key 为与 id 关联的图像的属性键

返回:是否成功

bool removeFeature (in IStorageID id, in String key)

说明:删除指定的图像的特征。

参数:id 表示图像的存储对象,key 为与 id 关联的图像的特征的键

返回:是否成功

5.5.5.2 视频存储与访问接口

接口名:udms_store::IVideoAccess

说明:存储及访问视频的接口。在系统中,视频对应于 Video 类。

继承:无

属性:无

方法:

void put (in IStorageID id, in Video vdo)

说明:上传视频数据。

参数:id 表示视频的存储对象,vdo 为视频的实例对象

返回:无

void putAttribute (in IStorageID id, in String key, in any data)

说明:上传视频的元数据。

参数:id 表示视频的存储对象,key 为与 id 关联的视频的属性键,data 为属性值

返回:无

void putFeature (in IStorageID id, in String key, in any data)

说明:上传视频的特征数据。

参数:id 表示视频的存储对象,key 为与 id 关联的视频的特征的键,data 为特征数据

返回:无

Video get (in IStorageID id)

说明:获取指定的视频的实例对象。

参数:id 表示视频的存储对象

返回:视频实例

any getAttribute (in IStorageID id, in String key)

说明:获取指定的视频的属性值。

参数:id 表示视频的存储对象,key 为与 id 关联的视频的属性键

返回:属性值

any getFeature (in IStorageID id, in String key)

说明:获取指定的视频的特征数据。

参数:id 表示视频的存储对象,key 为与 id 关联的视频的特征的键

返回:特征数据

bool remove (in IStorageID id)

说明:删除指定的视频数据。

参数:id 表示视频的存储对象

返回:是否成功

bool removeAttribute (in IStorageID id, in String key)

说明:删除指定的视频的属性。



参数:id 表示视频的存储对象,key 为与 id 关联的视频的属性键

返回:是否成功

bool removeFeature (in IStorageID id, in String key)

说明:删除指定的视频的特征。

参数:id 表示视频的存储对象,key 为与 id 关联的视频的特征的键

返回:是否成功

5.5.5.3 音频存储与访问接口

接口名:udms_store::IAudioAccess

说明:存储及访问音频的接口。在系统中,音频对应于 Audio 类。

继承:无

属性:无

方法:

void put (in IStorageID id, in Audio ado)

说明:上传音频数据。

参数:id 表示音频的存储对象,ado 为音频的实例对象

返回:无

void putAttribute (in IStorageID id, in String key, in any data)

说明:上传音频的元数据。

参数:id 表示音频的存储对象,key 为与 id 关联的音频的属性键,data 为属性值

返回:无

void putFeature (in IStorageID id, in String key, in any data)

说明:上传音频的特征数据。

参数:id 表示音频的存储对象,key 为与 id 关联的音频的特征的键,data 为特征数据

返回:无

Audio get (in IStorageID id)

说明:获取指定的音频的实例对象。

参数:id 表示音频的存储对象

返回:音频实例

any getAttribute (in IStorageID id, in String key)

说明:获取指定的音频的属性值。

参数:id 表示音频的存储对象,key 为与 id 关联的音频的属性键

返回:属性值

any getFeature (in IStorageID id, in String key)

说明:获取指定的音频的特征数据。

参数:id 表示音频的存储对象,key 为与 id 关联的音频的特征的键

返回:特征数据

bool remove (in IStorageID id)

说明:删除指定的音频数据。

参数:id 表示音频的存储对象

返回:是否成功

bool removeAttribute (in IStorageID id, in String key)

说明:删除指定的音频的属性。

参数:id 表示音频的存储对象,key 为与 id 关联的音频的属性键

返回:是否成功

bool removeFeature (in IStorageID id, in String key)

说明:删除指定的音频的特征。

参数:id 表示音频的存储对象,key 为与 id 关联的音频的特征的键

返回:是否成功

5.5.5.4 文本存储与访问接口

接口名:udms_store::IDocumentAccess

说明:存储及访问文本的接口。在系统中,文本对应于 Document 类。

继承:无

属性:无

方法:

void put (in IStorageID id, in Document doc)

说明:上传文本数据。

参数:id 表示文本的存储对象,doc 为文本的实例对象

返回:无

void putAttribute (in IStorageID id, in String key, in any data)

说明:上传文本的元数据。

参数:id 表示文本的存储对象,key 为与 id 关联的文本的属性键,data 为属性值

返回:无

void putFeature (in IStorageID id, in String key, in any data)

说明:上传文本的特征数据。

参数:id 表示文本的存储对象,key 为与 id 关联的文本的特征的键,data 为特征数据

返回:无

Document get (in IStorageID id)

说明:获取指定的文本的实例对象。

参数:id 表示文本的存储对象

返回:文本实例

any getAttribute (in IStorageID id, in String key)

说明:获取指定的文本的属性值。

参数:id 表示文本的存储对象,key 为与 id 关联的文本的属性键

返回:属性值

any getFeature (in IStorageID id, in String key)

说明:获取指定的文本的特征数据。

参数:id 表示文本的存储对象,key 为与 id 关联的文本的特征的键

返回:特征数据

bool remove (in IStorageID id)

说明:删除指定的文本数据。

参数:id 表示文本的存储对象

返回:是否成功

bool removeAttribute (in IStorageID id, in String key)

说明:删除指定的文本的属性。

参数:id 表示文本的存储对象,key 为与 id 关联的文本的属性键

返回:是否成功

bool removeFeature (in IStorageID id, in String key)

说明:删除指定的文本的特征。

参数:id 表示文本的存储对象,key 为与 id 关联的文本的特征的键

返回:是否成功

5.5.5.5 图数据存储与访问接口

接口名:udms_store::IGraphAccess

说明:存储及访问图数据的接口。在系统中,图对应于 Graph 类。

继承:无

属性:无

方法:

void put (in IStorageID id, in Graph graph)

说明:上传图数据。

参数:id 表示图的存储对象,graph 为图的实例对象

返回:无

void putAttribute (in IStorageID id, in String key, in any data)

说明:上传图数据元数据。

参数:id 表示图的存储对象,key 为与 id 关联的图的属性键,data 为属性值

返回:无

void putFeature (in IStorageID id, in String key, in any data)

说明:上传图数据特征数据。

参数:id 表示图的存储对象,key 为与 id 关联的图的特征的键,data 为特征数据

返回:无

Graph get (in IStorageID id)

说明:获取指定的图的实例对象。

参数:id 表示图的存储对象

返回:图实例

any getAttribute (in IStorageID id, in String key)

说明:获取指定的图的属性值。

参数:id 表示图的存储对象,key 为与 id 关联的图的属性键

返回:属性值

any getFeature (in IStorageID id, in String key)

说明:获取指定的图的特征数据。

参数:id 表示图的存储对象,key 为与 id 关联的图的特征的键

返回:特征数据



bool remove (in IStorageID id)

说明:删除指定的图数据。

参数:id 表示图的存储对象

返回:是否成功

bool removeAttribute (in IStorageID id, in String key)

说明:删除指定的图的属性。

参数:id 表示图的存储对象,key 为与 id 关联的图的属性键

返回:是否成功

bool removeFeature (in IStorageID id, in String key)

说明:删除指定的图的特征。

参数:id 表示图的存储对象,key 为与 id 关联的图的特征的键

返回:是否成功

5.5.5.6 过程数据存储与访问接口

接口名:udms_store::IWorkflowAccess

说明:存储及访问过程数据的接口。在系统中,过程对应于 Workflow 类,过程的存储格式通常为 XML。

继承:无

属性:无

方法:

void put (in IStorageID id, in Workflow wf)

说明:上传过程数据。

参数:id 表示过程的存储对象,wf 为过程的实例对象

返回:无

void putAttribute (in IStorageID id, in String key, in any data)

说明:上传过程的元数据。

参数:id 表示过程的存储对象,key 为与 id 关联的过程的属性键,data 为属性值

返回:无

void putFeature (in IStorageID id, in String key, in any data)

说明:上传过程的特征数据。

参数:id 表示过程的存储对象,key 为与 id 关联的过程的特征的键,data 为特征数据

返回:无

Workflow get (in IStorageID id)

说明:获取指定的过程的实例对象。

参数:id 表示过程的存储对象

返回:过程实例

any getAttribute (in IStorageID id, in String key)

说明:获取指定的过程的属性值。

参数:id 表示过程的存储对象,key 为与 id 关联的过程的属性键

返回:属性值

any getFeature (in IStorageID id, in String key)

说明:获取指定的过程的特征数据。

参数:id 表示过程的存储对象,key 为与 id 关联的过程的特征的键

返回:特征数据

bool remove (in IStorageID id)

说明:删除指定的过程数据。

参数:id 表示过程的存储对象

返回:是否成功

bool removeAttribute (in IStorageID id, in String key)

说明:删除指定的过程的属性。

参数:id 表示过程的存储对象,key 为与 id 关联的过程的属性键

返回:是否成功

bool removeFeature (in IStorageID id, in String key)

说明:删除指定的过程的特征。

参数:id 表示过程的存储对象,key 为与 id 关联的过程的特征的键

返回:是否成功

5.5.5.7 复杂类型数据存储与访问接口

接口名:udms_store::IObjectAccess

说明:本接口定义了除以上数据类型以外其他非结构化数据的访问接口,后文中所有其他的非结构化数据将统称为“复杂类型数据”。

继承:无

属性:无

方法:

void put (in IStorageID id, in Object obj)

说明:上传复杂类型数据。

参数:id 表示复杂类型数据的存储对象,obj 为复杂类型数据的实例对象

返回:无

void putAttribute (in IStorageID id, in String key, in any data)

说明:上传复杂类型数据的属性信息。

参数:id 表示复杂类型数据存储对象,key 为与 id 关联的属性键,data 为属性值

返回:无

void putFeature (in IStorageID id, in String key, in any data)

说明:上传复杂类型数据的特征数据。

参数:id 表示复杂类型数据的存储对象,key 为与 id 关联的特征的键,data 为特征数据

返回:无

Object get (in IStorageID id)

说明:获取指定的复杂类型数据的实例对象。

参数:id 表示复杂类型数据的存储对象

返回:实例对象

any getAttribute (in IStorageID id, in String key)

说明:获取指定的复杂类型数据的属性值。

参数:id 表示复杂类型数据的存储对象,key 为与 id 关联的复杂类型数据的属性键

返回:属性值

any getFeature (in IStorageID id, in String key)

说明:获取指定的复杂类型数据的特征数据。

参数:id 表示复杂类型数据的存储对象,key 为与 id 关联的复杂类型数据的特征的键

返回:特征数据

bool remove (in IStorageID id)

说明:删除指定的复杂类型数据。

参数:id 表示复杂类型数据的存储对象

返回:是否成功

bool removeAttribute (in IStorageID id, in String key)

说明:删除指定的过程的属性。

参数:id 表示过程的存储对象,key 为与 id 关联的过程的属性键

返回:是否成功

bool removeFeature (in IStorageID id, in String key)

说明:删除指定的复杂类型数据的特征。

参数:id 表示复杂类型数据的存储对象,key 为与 id 关联的复杂类型数据的特征的键

返回:是否成功

5.6 分析接口模块

5.6.1 概述

模块名:udms_analysis

说明:数据分析类型及操作定义的模块。

5.6.2 基本数据分析接口

接口名:udms_analysis::IBasicMetrics

说明:提供基本的数据分析操作。

继承:无

属性:无

方法:

int count (in List dataList, optional in FilterList filters)

说明:计数函数,可通过 filters 过滤器过滤掉不需要考虑的对象。

参数:dataList 为待计数的数据列表,filters 为过滤器列表,每个过滤器单独作用于数据列表,filters 为可选项

返回:元素个数

int countUnique (in List dataList, optional in FilterList filters)

说明:无重复的元素的计数函数,可通过 filters 过滤器过滤掉不需要考虑的对象。

参数:dataList 为待计数的数据列表,filters 为过滤器列表,每个过滤器单独作用于数据列表,filters 为可选项

返回:无重复的元素个数

any minimum (in List dataList, optional in FilterList filters, optional in Sort sortable)

说明:获取列表中的最小值,可通过 filters 过滤器过滤掉不需要考虑的对象,也可通过 sortable 指定排序函数。

参数:dataList 为待考虑的对象列表,filters 为过滤器列表,每个过滤器单独作用于数据列表,sortable 为排序函数,指定了元素之间的大小关系,filters 和 sortable 为可选项

返回:列表中的最小值

any maximum (in List dataList, optional in FilterList filters, optional in Sort sortable)

说明:获取列表中的最大值,可通过 filters 过滤器过滤掉不需要考虑的对象,也可通过 sortable 指定排序函数。

参数:dataList 为待考虑的对象列表,filters 为过滤器列表,每个过滤器单独作用于数据列表,sortable 为排序函数,指定了元素之间的大小关系,filters 和 sortable 为可选项

返回:列表中的最大值

any average (in List dataList, optional in FilterList filters)

说明:获取列表中元素的平均值,可通过 filters 过滤器过滤掉不需要考虑的对象。

参数:dataList 为待考虑的对象列表,filters 为过滤器列表,每个过滤器单独作用于数据列表,filters 为可选项

返回:列表中元素的平均值

any variance (in List dataList, optional in FilterList filters)

说明:获取列表中元素的方差,可通过 filters 过滤器过滤掉不需要考虑的对象。

参数:dataList 为待考虑的对象列表,filters 为过滤器列表,每个过滤器单独作用于数据列表,filters 为可选项

返回:列表中元素的方差

any quantile (in List dataList, in float quantile, optional in FilterList filters, optional Sort sortable)

说明:获取列表中元素的 quantile 分位数,可通过 filters 过滤器过滤掉不需要考虑的对象,也可通过 sortable 指定排序函数,若 quantile 为 0.5,则返回值为列表的中位数。

参数:dataList 为待考虑的对象列表,quantile 为分位数值,filters 为过滤器列表,每个过滤器单独作用于数据列表,sortable 为排序函数,指定了元素之间的大小关系,filters 和 sortable 为可选项

返回:列表中元素的 quantile 分位数值

any sum (in List dataList, optional in FilterList filters)

说明:获取列表中所有元素值的和,可通过 filters 过滤器过滤掉不需要考虑的对象。

参数:dataList 为待考虑的对象列表,filters 为过滤器列表,每个过滤器单独作用于数据列表,filters 为可选项

返回:列表中元素值的和



5.6.3 时间单位

时间单位名:PeriodType

类型:枚举类型,取值可为:

Minutely,每分钟;

Hourly,每小时;

Daily,每天;

Weekly,每星期;

Monthly, 每月;
 Quarterly, 每季度;
 Yearly, 每年。

5.6.4 时序数据分析接口

接口名: udms_analysis::ISeries

说明: 提供获取时序数据的基本操作。

继承: 无

属性: 无

方法:

List getSeries (in IStorageID id, in PeriodType type, in long startTSP, in int span)

说明: 获取 id 指定的数据项上, 从 startTSP 时间开始, 跨度为 span 个时间单位为 type 的区间的序列数据。

参数: id 为数据存储的对象, type 为时间单位, startTSP 为起始时间的时间戳, span 为时间跨度, 其单位为 type

返回: 指定时间段内的数据序列

List getSeries (in List list, in PeriodType type, in long startTSP, in int span)

说明: 获取数据列表 list 上, 从 startTSP 时间开始, 跨度为 span 个时间单位为 type 的区间的序列数据。

参数: list 为待筛选的数据列表, type 为时间单位, startTSP 为起始时间的时间戳, span 为时间跨度, 其单位为 type

返回: 指定时间段内的数据序列

5.7 索引接口模块

5.7.1 概述

模块名: udms_index

说明: 索引类型及操作定义的模块。

5.7.2 通用向量接口

接口名: udms_index::IVectorObject

说明: 向量在系统中的统一表示。

继承: 无

属性:

名称: id

定义: public IStorageID id

说明: 向量代表的对象的 ID。

名称: data

定义: public FloatArray data

说明: 向量表示的浮点数数组。

方法:

float getDistance (in IVectorObject vec)

说明:计算向量之间的距离。

参数:vec 为比较的向量

返回:向量之间的距离

FloatArray getData ()

说明:获取向量的浮点数数组。

参数:无

返回:浮点数数组

5.7.3 特征抽取接口

5.7.3.1 特征抽取基本接口

接口名:udms_index::IFeatureExtraction

说明:提供特征抽取的基本方法。

继承:无

属性:无

说明:未直接在本标准中说明的其他非结构化数据类型的特征抽取,可继承或实现本接口以扩展。

方法:

any extract()

说明:特征抽取通用接口。

参数:无

返回:特征数据

5.7.3.2 图像特征抽取接口



接口名:IImageFExtraction

说明:提供图像特征提取的基本方法。

继承:udms_index::IFeatureExtraction

属性:无

方法:

any extractColorFeature (in Image img)

说明:抽取图像的颜色特征,颜色特征种类任意。

参数:img 为图像的实例对象,img 参数可以替换为图像的字节数组 ByteArray、输入流 IStream 或文件存储地址 String 等

返回:图像的颜色特征

any extractTextureFeature(in Image img)

说明:抽取图像的纹理特征,纹理特征种类任意。

参数:img 为图像的实例对象,img 参数可以替换为图像的字节数组 ByteArray、输入流 IStream 或文件存储地址 String 等

返回:图像的纹理特征

any extracShapeFeature(in Image img)

说明:抽取图像的形状特征,形状特征种类任意。

参数:img 为图像的实例对象,img 参数可以替换为图像的字节数组 ByteArray、输入流
IStream 或文件存储地址 String 等

返回:图像的形状特征

any extracLocalFeature(in Image img)

说明:抽取图像的局部特征,局部特征种类任意。

参数:img 为图像的实例对象,img 参数可以替换为图像的字节数组 ByteArray、输入流
IStream 或文件存储地址 String 等

返回:图像的局部特征

5.7.3.3 视频特征抽取接口

接口名:IVideoFExtraction

说明:提供视频特征提取的基本方法。

继承:udms_index;IFeatureExtraction

属性:无

方法:

any extractTimeFeature (in Video vdo)

说明:抽取视频的时间域特征,特征种类任意。

参数:vdo 为视频的实例对象,vdo 参数可以替换为视频的字节数组 ByteArray、输入流
IStream 或文件存储地址 String 等

返回:视频的时间域特征

any extractMotionFeature (in Video vdo)

说明:抽取视频的运动特征,特征种类任意。

参数:vdo 为视频的实例对象,vdo 参数可以替换为视频的字节数组 ByteArray、输入流
IStream 或文件存储地址 String 等

返回:视频的运动特征

any extractLocalFeature (in Video vdo, in int winSize, in int hopSize)

说明:抽取视频局部特征,特征种类任意。

参数:vdo 为视频的实例对象,vdo 参数可以替换为视频的字节数组 ByteArray、输入流
IStream 或文件存储地址 String 等

返回:视频局部特征

5.7.3.4 音频特征抽取接口

接口名:IAudioFExtraction

说明:提供音频特征提取的基本方法。

继承:udms_index;IFeatureExtraction

属性:无

方法:

any extractTemporalFeature(in Audio ado)

说明:抽取音频的时序特征。

参数:ado 为音频的实例对象,ado 参数可以替换为音频的字节数组 ByteArray、输入流
IStream 或文件存储地址 String 等

返回:音频的时序特征

any extractFrequencyFeature(in Audio ado)

说明:抽取音频的频率特征。

参数:ado 为音频的实例对象,ado 参数可以替换为音频的字节数组 ByteArray、输入流
IStream 或文件存储地址 String 等

返回:音频的频率特征

any extractPerceptualFeature(in Audio ado)

说明:抽取音频的感知特征。

参数:ado 为音频的实例对象,ado 参数可以替换为音频的字节数组 ByteArray、输入流
IStream 或文件存储地址 String 等

返回:音频的感知特征

any extractCepstralFeature(in Audio ado)

说明:抽取音频的倒谱特征。

参数:ado 为音频的实例对象,ado 参数可以替换为音频的字节数组 ByteArray、输入流
IStream 或文件存储地址 String 等

返回:音频的倒谱特征

any extractLocalFeature (in Audio ado, in int winSize, in int hopSize)

说明:抽取音频局部特征,局部特征种类任意。

参数:ado 为音频的实例对象,ado 参数可以替换为音频的字节数组 ByteArray、输入流
IStream 或文件存储地址 String 等

返回:音频局部特征

5.7.3.5 文本特征抽取接口

接口名:IDocumentFExtraction

说明:提供文本特征提取的基本方法。

继承:udms_index:IFeatureExtraction

属性:无

方法:

any extractFrequencyFeature(in String doc)

说明:抽取文本词频相关的特征,特征种类任意。

参数:待抽取特征的文本字符串 doc

返回:词频特征

5.7.3.6 图数据特征抽取接口

接口名: IGraphFExtraction

说明: 提供图结构数据特征提取的基本方法。

继承: udms_index; IFeatureExtraction

属性: 无

方法:

any extractLocal(in Graph graph, optional in NodeList seeds)

说明: 抽取子图特征。

参数: graph 为原图的实例对象, seeds 为种子顶点, seeds 参数为可选项

返回: 子图的特征

5.7.3.7 过程数据特征抽取接口

接口名: IWorkflowFExtraction

说明: 提供过程数据特征提取的基本方法。

继承: udms_index; IFeatureExtraction

属性: 无

方法:

any extractLocal(in Workflow wf, optional in NodeList seeds)

说明: 抽取过程子网特征。

参数: wf 为原过程网的实例对象, seeds 为由库所和变迁组成的种子顶点, seeds 参数为可选项

返回: 过程子网的特征

5.7.3.8 复杂类型数据的特征抽取接口

接口名: IObjectFExtraction

说明: 提供复杂类型数据特征提取的基本方法。

继承: udms_index; IFeatureExtraction

属性: 无

方法:

any extractLocal(in Object obj)

说明: 抽取复杂类型数据的局部特征。

参数: obj 为复杂类型数据的实例对象, 或字节数组, 或输入流, 或存储地址

返回: 局部特征



5.7.4 通用索引接口

5.7.4.1 通用索引基本接口

接口名: udms_index; IBaseIndex

说明: 提供基本的索引方法。

继承:无

属性:

名称: type

定义: public IndexType type

说明: 索引的类型

方法:

void create(in IndexType type)

说明: 创建新索引。

参数: type 为索引类型

返回: 无

void drop()

说明: 删除全部索引记录。

参数: 无

返回: 无

bool exist(in any indexable)

说明: 查看指定的索引是否存在。

参数: indexable 为索引

返回: 索引是否存在

bool merge(in IBaseIndex anotherindex)

说明: 合并现有的索引。

参数: anotherindex 为待合并的索引

返回: 索引合并是否成功



void index(in any indexable)

说明: 在当前索引中添加一个索引项。

参数: indexable 为待索引的对象

返回: 无

void remove (in any indexable)

说明: 在当前索引中删除一个索引项。

参数: indexable 为待删除的对象

返回: 无

List list()

说明: 按索引顺序获取所有的索引对象的列表。

参数: 无

返回: 索引对象的列表

5.7.4.2 通用高维索引接口

接口名:udms_index::IHDIndex

说明:提供高维索引的基本操作。

继承:udms_index::IBaseIndex

属性:无

方法:

void index(IVectorObject vec)

说明:将单个向量添加到索引中。

参数:vec 为待添加的向量

返回:无

void index(IVectorList veclist)

说明:将 veclist 向量列表中的每个向量添加到索引中。

参数:veclist 为待添加的向量列表

返回:无

IVectorObject getNearest(IVectorObject vec)

说明:获取向量 vec 的最近邻的向量。

参数:vec 为待查询的向量

返回:最近邻的向量

IVectorList getKNearest(IVectorObject vec, int k)

说明:获取向量 vec 的 k 近邻向量。

参数:vec 为待查询的向量,k 为将要获取的近邻个数

返回:k 近邻的向量列表

5.7.4.3 文本索引接口

接口名:udms_index::ITextIndex

说明:提供文本索引的基本操作。

继承:udms_index::IBaseIndex

属性:无

方法:

IStorageID index(IStorageID inputId)

说明:索引单个文本文件。



参数:inputId 为待索引文件的存储对象

返回:索引存储的对象

IStorageIDList index(IStorageIDList inputIdList)

说明:批量索引文本文件。

参数:inputIdList 为待索引的多个文件的存储对象列表

返回:多个索引的对象的列表

5.7.4.4 通用键值索引接口

接口名:udms_index::IKVIndex

说明:提供键值索引的基本操作。

继承:udms_index::IBaseIndex

属性:无

方法:

IStorageIDList list()

说明:按索引顺序获取对象列表。

参数:无

返回:对象列表

void index(in any key, IStorageID id)

说明:将 id 代表的对象按键 key 添加到索引中。

参数:id 为待索引的对象,key 为该对象的键

返回:无

IStorageID get(in any key)

说明:查询键为 key 的索引对象,若有多个对象,则返回第一个。

参数:key 为待查询的键

返回:键为 key 的第一个索引的对象

IStorageIDList getAll(in any key, optional in int limit)

说明:查询键为 key 的所有索引对象,若设置 limit 的值,则最多返回 limit 条记录。

参数:key 为待查询的键,limit 为最多返回的记录数(可选)

返回:键为 key 的所有索引的对象

IStorageIDList between(in any keyLower, in any keyUpper)

说明:查询键介于 keyLower 和 keyUpper 之间的所有索引对象。

参数:keyLower 为待查范围的键的下界,keyUpper 为待查范围的键的上界

返回:范围内的所有索引对象列表

5.8 搜索接口模块

5.8.1 概述

模块名:udms_search

说明:搜索类型及操作定义的模块。

5.8.2 媒体数据类型

媒体数据类型名:MediaType

类型:枚举类型,取值可为:

MDocument, 文本类型;

MImage, 图像类型;

MVideo, 视频类型;

MAudio, 音频类型;
 MGraph, 图类型;
 MWorkflow, 过程类型;
 MAny, 任意类型。

5.8.3 高维数据查询接口

接口名: udms_search::IHDSearch

说明: 提供基本高维数据查询操作的接口。

继承: 无

属性: 无

方法:

IVectorList search(in IVectorObject vec, in int k, in MediaType media)

说明: 查询指定类型的高维向量的 k 近邻。

参数: vec 为待查询的高维向量, k 为近邻个数, media 为指定的媒体数据类型

返回: k 近邻的向量列表

IVectorList search(in IVectorObject vec, in int k, in Sortable sort, in MediaType media)

说明: 根据给定的排序方法 sort, 查询指定类型的高维向量的 k 近邻。

参数: vec 为待查询的高维向量, k 为近邻个数, media 为指定的媒体数据类型

返回: k 近邻的向量列表

IVectorList search(in IStorageID id, in int k, in MediaType media)

说明: 查询指定类型的存储对象的 k 近邻。

参数: id 为待查询的对象, k 为近邻个数

返回: k 近邻的向量列表

IVectorList search(in IStorageID id, in int k, in Sortable sort, in MediaType media)

说明: 根据给定的排序方法 sort, 查询指定类型的存储对象的 k 近邻。

参数: id 为待查询的对象, k 为近邻个数, media 为指定的媒体数据类型

返回: k 近邻的向量列表

5.8.4 关键字查询接口

接口名: udms_search::IKeywordSearch

说明: 提供基本关键字查询操作的接口。

继承: 无

属性: 无

方法:

IStorageIDList search(in String term, in int n, in MediaType media)

说明: 查询指定类型的, 与关键词 term 最相关的 n 个对象。

参数: term 为待查询的关键词, n 为返回结果的数量, media 为指定的媒体数据类型

返回: 相关性最高的 n 个对象的列表

IStorageIDList search(in String term, in int n, in Sortable sort, in MediaType media)

说明:根据给定的排序方法 sort,查询指定类型的,与关键词 term 最相关的 n 个对象。

参数:term 为待查询的关键词,n 为返回结果的数量,sort 为排序函数,media 为指定的媒体数据类型

返回:相关性最高的 n 个对象的列表

IStorageIDList search(in StringList termlist, in int n, in MediaType media)

说明:查询指定类型的,与关键词列表 termlist 中每个词的综合相关性最高的 n 个对象。

参数:termlist 为待查询的关键词列表,n 为返回结果的数量,media 为指定的媒体数据类型

返回:相关性最高的 n 个对象的列表

IStorageIDList search(in StringList termlist, in int n, int Sortable sor, in MediaType media)

说明:根据给定的排序方法 sort,查询指定类型的,与关键词列表 termlist 中每个词的综合相关性最高的 n 个对象。

参数:termlist 为待查询的关键词列表,n 为返回结果的数量,sort 为排序函数,media 为指定的媒体数据类型

返回:相关性最高的 n 个对象的列表

6 Web 服务访问接口

6.1 概述

本章描述 Web 服务访问接口,包括 HTTP 请求格式、HTTP 状态码和服务接口描述。

6.2 基本格式

Web 服务访问接口包含了以下 5 种元素:

- a) 接口名:接口的名称;
- b) 说明:对接口内容的描述;
- c) 调用格式:调用 Web 服务时的语法格式;
- d) 语义:对调用格式的解释;
- e) 调用结果:Web 服务调用结果,包括成功状态和失败状态下的响应信息。

6.3 HTTP 请求格式

服务的调用由 HTTP 请求发起,HTTP 请求具有如下的基本格式:

WebService ::= <METHOD><ENTRY>[<PARAM>][<HEADER>][<STREAM>]

在上述基本调用格式中,各个字段的含义如表 1 所示。HTTP 请求的返回结果包含两部分,一部分为 HTTP 消息的状态码(StatusCode),表示响应的状态,另一部分为 HTTP 请求的消息体,消息体默认采用 JSON 格式进行封装,可通过参数设置返回 XML 格式的消息。

表 1 Web 服务接口的字段含义

字段	含 义
〈METHOD〉	HTTP 请求的方法头,本接口中使用到的该字段取值有 POST, PUT 和 DELETE,各取值的含义请参考 HTTP 协议
〈ENTRY〉	Web 服务接口的入口地址
〈PARAM〉	可选参数,用以传递少量的参数信息
〈HEADER〉	HTTP 请求的消息头部
〈STREAM〉	可选的附加流信息,用以传递大量的参数信息

6.4 HTTP 状态码

HTTP 状态码及其含义如表 2 所示。

表 2 HTTP 状态码的表示及含义

状态码编号	状态码名称	含 义
200	OK	请求已成功
201	CREATED	对象或资源已成功创建
202	ACCEPTED	操作已被成功接受
400	BADREQUEST	请求的内容丢失或不合法
404	NOTFOUND	请求的内容不存在
409	CONFLICT	访问冲突,访问资源已被上锁或可能导致服务器内部状态出错
500	INTERNAL_ERROR	服务器出现内部错误,无法正确响应请求

6.5 服务接口描述

6.5.1 数据增加接口

接口名: Create

说明:提供增加数据的功能,包括源数据、元数据和特征数据等。

调用格式:

PUT/create [id,class,type,subtype1[,subtype2,...]] 〈STREAM〉

语义:

上传数据到指定的 id 位置,若未指定 id,则系统自动生成 id;参数 class 指定待增加的数据的类型,例如图片、视频、音频、文本等;参数 type 指定待增加的数据的基本类别,例如源数据、元数据或特征数据;参数 subtype 指定待增加的数据的子类别,可含多级子类别,例如图片特征中的颜色特征或纹理特征,又如颜色特征中的 RGB 特征或 HSV 特征等。上述调用格式中,〈STREAM〉为上传的数据流。

调用结果:

成功状态：

返回的 HTTP 状态码为 201。若未指定 id,则返回新生成的文件的 id,格式为 JSON,如：
{“id”: “123456789”}。

失败状态：

若 id 格式错误,则返回的 HTTP 状态码为 400;若 id 已经存在(发生冲突),返回的 HTTP 状态码为 409;其他内部错误,返回的 HTTP 状态码为 500。

6.5.2 数据删除接口

接口名: Delete

说明:提供删除数据的功能,包括源数据、元数据和特征数据等。

调用格式:

DELETE/delete [id,class,type,subtype1[,subtype2,...]]

语义:

删除 id 指定的数据;参数 class 指定待删除的数据的类型,例如图片、视频、音频、文本等;参数 type 指定待删除的数据的基本类别,例如源数据、元数据或特征数据;参数 subtype 指定删除的数据的子类别,可含多级子类别;若不指定 class 和 type,则删除与 id 关联的所有数据。

调用结果:

成功状态:

返回的 HTTP 状态码为 202。

失败状态:

若 id 格式错误,返回的 HTTP 状态码为 400;若 id 不存在,返回的 HTTP 状态码为 404;其他内部错误,返回的 HTTP 状态码为 500。

6.5.3 数据更新接口

接口名: Update

说明:提供数据的更新功能,包括源数据、元数据、特征数据等。

调用格式:

POST /update [id] <STREAM>

语义:

更新 id 指定的数据;<STREAM>的附加流指定结构化的子句信息,例如需要更新的数据的类型、基本类别、属性值等;<STREAM>的结构由具体的实现方案而定。附加流<STREAM>支持 XML 和 JSON 格式,需在 HTTP 请求的 Content-Type 中指定。

示例:

一个合法的 XML 格式的请求如下:

```
<properties>
  <property>
    <class>image</class>
    <type>feature</type>
```

```

    <subtype>
      <name>SURF</name>
      <value>[1,2,3,4,5]</value>
    </subtype>
    <subtype>
      <name>GIST</name>
      <value>[2,3,5,6,8]</value>
    </subtype>
  </property>
</property>
  <class>image</class>
  <type>meta</type>
  <subtype>
    <name>title</name>
    <value>a_new_title</value>
  </subtype>
  <subtype>
    <name>location</name>
    <value>Beijing</value>
  </subtype>
</property>
</properties>

```

与此对应的 JSON 格式的示例如下：

```

{
  "properties":
  [
    {
      "class": "image",
      "type": "feature",
      "subtype":
      [
        {
          "name": "SURF", "value": "[1,2,3,4,5]",
          "name": "GIST", "value": "[2,3,5,6,8]"
        }
      ]
    },
    {
      "class": "image",
      "type": "meta",
      "subtype":
      [
        {
          "name": "title", "value": "a_new_title",
          "name": "location", "value": "Beijing"
        }
      ]
    }
  ]
}

```

调用结果：

成功状态：

返回的 HTTP 状态码为 202。

失败状态：

若 id 格式错误，返回的 HTTP 状态码为 400；若 id 不存在，返回的 HTTP 状态码为 404；若结构化子句格式不正确，返回的 HTTP 状态码为 400；其他内部错误，返回的 HTTP 状态码为 500。

6.5.4 数据查询接口

接口名: Select

说明:按条件查询特定的数据。

调用格式:

POST /select <STREAM>

语义:

获取满足查询条件的数据;<STREAM>的附加流指定结构化的子句信息,例如选择的属性以及需要满足的查询条件(where子句);<STREAM>的结构由具体的实现方案而定;附加流<STREAM>支持XML和JSON格式,需在HTTP请求的Content-Type中指定。

示例 1:

一个合法的XML格式的请求如下:

```
<select>
  <properties>
    <field>* </field>
  </properties>
  <where-clause>
    Feature1=2 and feature2>feature3 or feature8 LIKE '%abc%'
  </where-clause>
  <top>5</top >
</select>
```

与此对应的JSON格式如下:

```
{
  "properties": {"field": "*"},
  "where-clause": "Feature1=2 and feature2>feature3 or feature8 LIKE '%abc%'",
  "top": "5"
}
```

调用结果:

成功状态:

返回的HTTP状态码为202,返回的结果用XML或JSON封装。

示例 2:

一个合法的XML格式的响应示例如下,

```
<Results>
  <Result>
    <id>1234567</id>
    <values>
      <field>
        <name>* </name>
        <value>123</value>
      </field>
    </values>
  <Result>
  <Result>
```

```

    <id>2345678</id>
    <values>
      <field>
        <name> * </name>
        <value>99</value>
      </field>
    </values>
  </Result>
</Results>

```

与此对应的 JSON 格式的示例如下：

```

{
  "Results": [
    {
      "id": "1234567",
      "values": [
        {
          "field": {
            "name": " * ",
            "value": "123"
          }
        }
      ]
    },
    {
      "id": "2345678",
      "values": [
        {
          "field": {
            "name": " * ",
            "value": "99"
          }
        }
      ]
    }
  ]
}

```

调用结果：

失败状态：

若结构化子句格式不正确，返回的 HTTP 状态码为 400；其他内部错误，返回的 HTTP 状态码为 500。