开源力量
Open Source Force
www.osforce.cn

HBase客户端API操作

主讲人： Louis

Java类与数据模型

开源力量 Open Source Force
www.osforce.cn

Java类	HBase数据模型
HBaseAdmin	数据库（DataBase)
HBaseConfiguration	
HTable	表（Table)
HTableDescriptor	列族（Column Family)
Put	列修饰符（Column Qualifier）
Get	
Scanner	

HBaseConfiguration

包名：org.apache.hadoop.hbase.HBaseConfiguration
作用：对HBase进行配置
用法示例：

```
HBaseConfiguration hconfig = new HBaseConfiguration();
hconfig.set("hbase.zookeeper.property.clientPort","2181");
```

返回值	函数	描述
void	addResource(Path file)	通过给定的路径所指的文件来添加资源
void	clear()	清空所有已设置的属性
string	get(String name)	获取属性名对应的值
String	getBoolean(String name, boolean defaultValue)	获取为boolean类型的属性值，如果其属性值类型部位boolean,则返回默认属性值
void	set(String name, String value)	通过属性名来设置值
void	setBoolean(String name, boolean value)	设置boolean类型的属性值

HBaseAdmin

包名：org.apache.hadoop.hbase.client.HBaseAdmin
作用：提供了一个接口来管理HBase数据库的表信息。它提供的方法包括：创建表，删除表，列出表项，使表有效或无效，以及添加或删除表列族成员等。
用法示例：

```
HBaseAdmin admin = new HBaseAdmin(config);
admin.disableTable("tablename")
```

返回值	函数	描述
void	addColumn(String tableName, HColumnDescriptor column)	向一个已经存在的表添加咧
	checkHBaseAvailable(HBaseConfiguration conf)	静态函数，查看HBase是否处于运行状态
	createTable(HTableDescriptor desc)	创建一个表，同步操作
	deleteTable(byte[] tableName)	删除一个已经存在的表
	enableTable(byte[] tableName)	使表处于有效状态
	disableTable(byte[] tableName)	使表处于无效状态
HTableDescriptor[]	listTables()	列出所有用户控件表项
void	modifyTable(byte[] tableName, HTableDescriptor htd)	修改表的模式，是异步的操作，可能需要花费一定的时间
boolean	tableExists(String tableName)	检查表是否存在

HTableDescriptor

开源力量 Open Source Force
www.osforce.cn

包名：org.apache.hadoop.hbase.HTableDescriptor
作用：包含了表的名字极其对应表的列族
用法示例：

```
HTableDescriptor htd = new HTableDescriptor(table);
htd.addFamily(new HcolumnDescriptor("family"));
```

返回值	函数	描述
void	addFamily(HColumnDescriptor)	添加一个列族
HColumnDescriptor	removeFamily(byte[] column)	移除一个列族
byte[]	getName()	获取表的名字
byte[]	getValue(byte[] key)	获取属性的值
void	setValue(String key, String value)	设置属性的值

HColumnDescriptor

包名：org.apache.hadoop.hbase.HColumnDescriptor
作用：维护着关于列族的信息，例如版本号，压缩设置等。它通常在创建表或者为表添加列族的时候使用。列族被创建后不能直接修改，只能通过删除然后重新创建的方式。列族被删除的时候，列族里面的数据也会同时被删除。
用法示例：

```
HTableDescriptor htd = new HTableDescriptor(tablename);
HColumnDescriptor col = new HColumnDescriptor("content:");
htd.addFamily(col);
```

返回值	函数	描述
byte[]	getName()	获取列族的名字
byte[]	getValue(byte[] key)	获取对应的属性的值
void	setValue(String key, String value)	设置对应属性的值

HTable

包名：org.apache.hadoop.hbase.client.HTable
作用：可以用来和HBase表直接通信。此方法对于更新操作来说是非线程安全的
用法示例：

```
HTable table = new HTable(conf, Bytes.toBytes(tablename));
ResultScanner scanner =  table.getScanner(family);
```

返回值	函数	描述
void	checkAdnPut(byte[] row, byte[] family, byte[] qualifier, byte[] value, Put put	自动的检查row/family/qualifier是否与给定的值匹配
void	close()	释放所有的资源或挂起内部缓冲区中的更新
Boolean	exists(Get get)	检查Get实例所指定的值是否存在于HTable的列中
Result	get(Get get)	获取指定行的某些单元格所对应的值
byte[][]	getEndKeys()	获取当前一打开的表每个区域的结束键值
ResultScanner	getScanner(byte[] family)	获取当前给定列族的scanner实例
HTableDescriptor	getTableDescriptor()	获取当前表的HTableDescriptor实例
byte[]	getTableName()	获取表名
static boolean	isTableEnabled(HBaseConfiguration conf, String tableName)	检查表是否有效
void	put(Put put)	向表中添加值

Put

包名：org.apache.hadoop.hbase.client.Put
作用：用来对单个行执行添加操作
用法示例：

```
HTable table = new HTable(conf,Bytes.toBytes(tablename));
Put p = new Put(brow);//为指定行创建一个Put操作
p.add(family,qualifier,value);
table.put(p);
```

返回值	函数	描述
void	checkAdnPut(byte[] row, byte[] family, byte[] qualifier, byte[] value, Put put	自动的检查row/family/qualifier是否与给定的值匹配
void	close()	释放所有的资源或挂起内部缓冲区中的更新
Boolean	exists(Get get)	检查Get实例所指定的值是否存在于HTable的列中
Result	get(Get get)	获取指定行的某些单元格所对应的值
byte[][]	getEndKeys()	获取当前一打开的表每个区域的结束键值
ResultScanner	getScanner(byte[] family)	获取当前给定列族的scanner实例
HTableDescriptor	getTableDescriptor()	获取当前表的HTableDescriptor实例
byte[]	getTableName()	获取表名
static boolean	isTableEnabled(HBaseConfiguration conf, String tableName)	检查表是否有效
void	put(Put put)	向表中添加值

Get

包名：org.apache.hadoop.hbase.client.Get
作用：用来获取单个行的相关信息
用法示例：

```
HTable table = new HTable(conf, Bytes.toBytes(tablename));
Get g = new Get(Bytes.toBytes(row));
table.get(g);
```

返回值	函数	描述
Get	addColumn(byte[] family, byte[] qualifier)	获取指定列族和列修饰符对应的列
Get	addFamily(byte[] family)	通过指定的列族获取其对应列的所有列
Get	setTimeRange(long minStamp,long maxStamp)	获取指定取件的列的版本号
Get	setFilter(Filter filter)	当执行Get操作时设置服务器端的过滤器

Result

包名：org.apache.hadoop.hbase.client.Result

作用：存储Get或者Scan操作后获取表的单行值。使用此类提供的方法可以直接获取值或者各种Map结构（key-value对）

返回值	函数	描述
boolean	containsColumn(byte[] family, byte[] qualifier)	检查指定的列是否存在
NavigableMap<byte[],byte[]>	getFamilyMap(byte[] family)	获取对应列族所包含的修饰符与值的键值对
byte[]	getValue(byte[] family, byte[] qualifier)	获取对应列的最新值

ResultScanner

包名：org.apache.hadoop.hbase.client.ResultScanner

作用：存储Get或者Scan操作后获取表的单行值。使用此类提供的方法可以直接获取值或者各种Map结构（key-value对）

返回值	函数	描述
void	close()	关闭scanner并释放分配给它的资源
Result	next()	获取下一行的值

HTablePool

包名：org.apache.hadoop.hbase.client.HTablePool
作用：可以解决HTable存在的线程不安全问题，同时通过维护固定数量的HTable对象，能够在程序运行期间复用这些HTable资源对象
说明：
1. HTablePool可以自动创建HTable对象，而且对客户端来说使用上是完全透明的，可以避免多线程间数据并发修改问题。
2. HTablePool中的HTable对象之间是公用Configuration连接的，能够可以减少网络开销。

HTablePool的使用很简单：每次进行操作前，通过HTablePool的getTable方法取得一个HTable对象，然后进行put/get/scan/delete等操作，最后通过HTablePool的putTable方法将HTable对象放回到HTablePool中。

```
@InterfaceAudience.Private
@Deprecated
public class HTablePool
extends Object
implements Closeable
```

A simple pool of HTable instances. Each HTablePool acts as a pool for all tables. To use, instantiate an HTablePool and use `getTable(String)` to get an HTable from the pool. This method is not needed anymore, clients should call HTableInterface.close() rather than returning the tables to the pool Once you are done with it, close your instance of `HTableInterface` by calling `HTableInterface.close()` rather than returning the tables to the pool with (deprecated) `putTable(HTableInterface)`.

A pool can be created with a *maxSize* which defines the most HTable references that will ever be retained for each table. Otherwise the default is `Integer.MAX_VALUE`.

Pool will manage its own connections to the cluster. See `HConnectionManager`.