



链滴

大数据学习笔记 (9) -- hbase 的 java 客户端的基本使用

作者: [kevinBobo](#)

原文链接: <https://ld246.com/article/1547707139163>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

直接上代码，具体参考代码里面的注释

```
package com.bobo.hbase;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.Cell;
import org.apache.hadoop.hbase.CellScanner;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.TableName;
import org.apache.hadoop.hbase.client.Admin;
import org.apache.hadoop.hbase.client.ColumnFamilyDescriptor;
import org.apache.hadoop.hbase.client.ColumnFamilyDescriptorBuilder;
import org.apache.hadoop.hbase.client.Connection;
import org.apache.hadoop.hbase.client.ConnectionFactory;
import org.apache.hadoop.hbase.client.Delete;
import org.apache.hadoop.hbase.client.Get;
import org.apache.hadoop.hbase.client.Put;
import org.apache.hadoop.hbase.client.Result;
import org.apache.hadoop.hbase.client.Table;
import org.apache.hadoop.hbase.client.TableDescriptor;
import org.apache.hadoop.hbase.client.TableDescriptorBuilder;
import org.apache.hadoop.hbase.regionserver.BloomType;
import org.apache.hadoop.hbase.util.Bytes;
import org.junit.Before;
import org.junit.Test;

import java.io.IOException;
import java.util.Arrays;

/**
 * @author bobo
 * @Description:
 * @date 2019-01-15 17:08
 */
public class HbaseMain {

    private Connection conn;

    private TableName userInfo = TableName.valueOf("user_info1");

    /**
     * 初始化客户端
     */
    @Before
    public void init() throws IOException {
        Configuration conf = HBaseConfiguration.create();
        conf.set("hbase.zookeeper.quorum", "hdp-master:2181,hdp-node:2181,hdp-node2:2181")

        conn = ConnectionFactory.createConnection(conf);
    }

    /**
     * 创建表
     */
}
```

```

* @throws IOException
*/
@Test
public void createTable() throws IOException {
    Admin admin = conn.getAdmin();
    TableDescriptor build = TableDescriptorBuilder.newBuilder(userInfo)
        .setColumnFamily(ColumnFamilyDescriptorBuilder.of("base_info"))
        .setColumnFamily(ColumnFamilyDescriptorBuilder.of("extra_info"))
        .build();
    admin.createTable(build);
    admin.close();
    conn.close();
}

/**
 * 修改表
 *
 * @throws IOException
 */
@Test
public void alterTable() throws IOException {
    Admin admin = conn.getAdmin();
    ColumnFamilyDescriptor baseInfo = ColumnFamilyDescriptorBuilder.newBuilder(Column
amilyDescriptorBuilder.of("base_info"))
        .setBloomFilterType(BloomType.ROWCOL)
        .build();
    ColumnFamilyDescriptor extraInfo = ColumnFamilyDescriptorBuilder.newBuilder(Column
amilyDescriptorBuilder.of("extra_info"))
        .setBloomFilterType(BloomType.ROWCOL)
        .build();
    TableDescriptor descriptor = TableDescriptorBuilder.newBuilder(userInfo)
        .setColumnFamily(baseInfo)
        .setColumnFamily(extraInfo)
        .build();
    admin.modifyTable(descriptor);
    admin.close();
    conn.close();
}

/**
 * 删除表
 *
 * @throws IOException
 */
@Test
public void delTable() throws IOException {
    Admin admin = conn.getAdmin();
    admin.disableTable(userInfo);
    admin.deleteTable(userInfo);
    admin.close();
    conn.close();
}

/**

```

```

* 添加表数据/修改表数据
*
* @throws IOException
*/
@Test
public void putData() throws IOException {
    Table table = conn.getTable(userInfo);
    Put put = new Put(Bytes.toBytes("001"));
    put.addColumn(Bytes.toBytes("base_info"), Bytes.toBytes("username"), Bytes.toBytes("zh
ngsan"));
    put.addColumn(Bytes.toBytes("base_info"), Bytes.toBytes("age"), Bytes.toBytes("18"));
    put.addColumn(Bytes.toBytes("base_info"), Bytes.toBytes("sex"), Bytes.toBytes("male"));
    put.addColumn(Bytes.toBytes("extra_info"), Bytes.toBytes("addr"), Bytes.toBytes("beijing"
));
    table.put(put);
    table.close();
    conn.close();
}

/**
* 删除表数据
*/
@Test
public void delData() throws IOException {
    Table table = conn.getTable(userInfo);
    Delete delete = new Delete(Bytes.toBytes("001"));
    table.delete(delete);
    table.close();
    conn.close();
}

/**
* 获取表数据
*
* @throws IOException
*/
@Test
public void getData() throws IOException {
    Table table = conn.getTable(userInfo);
    Get get = new Get(Bytes.toBytes("001"));
    Result result = table.get(get);
    byte[] value1 = result.value();
    CellScanner cellScanner = result.cellScanner();
    while (cellScanner.advance()){
        Cell cell = cellScanner.current();
        //行
        byte[] rowArray = cell.getRowArray();
        //value
        byte[] valueArray = cell.getValueArray();
        //列簇名数组
        byte[] familyArray = cell.getFamilyArray();
        //列名数组
        byte[] qualifierArray = cell.getQualifierArray();
        String family = new String(familyArray,"utf-8");
    }
}

```

```
        String qualifier = new String(qualifierArray,"utf-8");
        String row = new String(rowArray,"utf-8");
        String value = new String(valueArray,"utf-8");
        System.out.println(row+":"+family+":"+qualifier+":"+value);
    }
    table.close();
    conn.close();
}
}
```

关于具体代码可以参考[study-hadoop](#)