HBaseDAO 使用手册

1. 环境设定
2. 创建HBase实例名称

HTablePoolManager.createHBaseConfiguration("myinstance", "127.0.0.1", "2181"); // 设置名称是myinstance的HBase数据库

1. 使用HBase实例名称生成HBase Dao实例

HDao dao = new DefaultHDao("myinstance"); // 创建名称是myinstace的HDao实例。

1. 标注的使用
2. @HBaseTable，在类名上定义。指定类对应到表的相关信息。

包括以下设置：

* 1. name : 类型String, 表名
  2. autoCreate： 类型boolean, 是否在表不存在时，自动创建表。默认false，即不自动创建。
  3. families：类型String[], 列族数组。默认{"f"}。在autoCreate=true时，列族数组中必须存在至少一个列族。

eg：

@HBaseTable(name = "simple\_bean", autoCreate = **true**) **public** **class** SimpleBean {

1. @HRowkey，在类属性上定义。指定对应到HBase表rowkey的类属性。 **注意：**一个类中，最多只能有一个@HRowkey标注的属性。

eg：

@HRowkey **private** String rowkey;

1. @HColumn，在类属性上定义。指定对应到HBase表固定列名的列。包括以下设置：
   1. family, 指定固定列对应到的列族名称。不指定或者指定为空字符串时，使用@HBaseTable定义的列族数组families的第一个列族名称。
   2. key, 自定义列名。不指定或者指定为空字符串时，使用属性名称做为列名。

eg：

@HColumn **private** **int** age;

1. @HRowkeyPart，在类属性上定义。指定被标注的属性是rowkey的一部分。**注意：**不能与@HRowkey同时使用。包括以下设置：
   1. bytesLen, 指定组成占用rowkey的多少个字节。默认-1，表示由所标注的@HRowkeyPart类型决定或者不定长度。JAVA基本类型都不需要指定bytesLen。

注1: JAVA基本类型是指**boolean/byte/char/short/int/long/float/double**以及上述类型对应的对象类型。

注2: @HRowkeyPart所标注的属性，除了基本类型之外，最多只能有一个**不**定义bytesLen。

注3: rowkey由@HRowkeyPart所标注的属性值组成最终的rowkey字节数组。

eg:

@HRowkeyPart **private** String code;

@HRowkeyPart **private** **int** id;

1. @HDynamic，高级标注。在类属性上定义。指定被标注的属性对应到表的动态列。**注意：**只有标注在Map类型的属性上。包括以下设置：  
   a) family, 与@HColumn含义相同。

b) mapping。这个设置有点复杂。它指定一个@HTypePair数组。第个数组元素包括一个keyType和一个valueType。当向HBase存JAVABEAN时，对于Map中的每个entry，会依次使用数组元素的keyType去将entry.key转换成byte[], 转换不成功（返回null）表示转换失败，那继续尝试下一个，直到成功为止。若全部失败，则该entry不被存入HBase。若某个keyType转换成功，则使用对应的valueType转换entry.value值到byte[].

从HBase读取时，进行类似的尝试转换原则。

eg:

@HDynamic(mapping = {

@HTypePair(keyType = SignKey.**class**, valueType = String.**class**),

@HTypePair(keyType = ParamKey.**class**, valueType = ParamKeyValue.**class**),

@HTypePair(keyType = String.**class**, valueType = String.**class**)})  
**private** Map<Object, Object> dynamicProperties /\*= new HashMap<Object, Object>()\*/;

1. @HCascade，高级标注。在类属性上定义。表关联处理用。HBase表关联分两种情况：one-one以及one-many。下面例子里面@HCascade标注在List<T>上的表示one-many, 其它表示one-one。one-one时，使用相同的rowkey，one-many时，many使用one的rowkey作为前缀。包括以下设置：

a) clazz, 关联表对应的JAVA CLASS。默认不设置时，表示DAO会根据标注自动推导出关联表对应的JAVA CLASS。

b) rowkeyBytesLen, one-many时有效。表示子表rowkey的字节长度。DAO在处理关联查询时，需要它用来生成startRowkey和stopRowkey。默认不设置时，DAO会自动从clazz中推导。

eg:

@HCascade **private** List<SubItemBean> subItems;

@HCascade **private** List<SubItem2Bean> subItems2;

@HCascade **private** OtherBean otherBean;

1. @HParent，高级标注。在类属性上定义。表关联时子表对应的JAVA BEAN需要反向引用主表对应的JAVA BEAN时标注。

eg:

@HParent **private** MainBean mainBean;

1. 增删改查HDao操作
2. put/insert/update
3. get
4. delete
5. query
6. 对象与byte[]之间的互转
7. 基本类型与byte[]之间的互转
8. 实现BytesConvertable的JAVA BEAN与byte[]之间的互转
9. 其它JAVA BEAN与byte[]之间的互转