

詞条CRUD

20181227版

```
/**
詞条信息map (CRUD)
  /---- 操作   追加/检索
  /---- 目标   例、SDP
  /---- 条件   W詞条List
                /-----程序名   例, 累积成交量SDP
                /-----詞条A1   例, 股票代码
                /-----詞条A2   例, 时间
                /-----詞条B1   例, 股票累积成交量
                /-----说明     例, 这是一个通过股票代码和时间来计算累积成交量的SDP
                /-----SDP内容  例,      {a1,,,,},
                                   {a2,,,,},
                                   {a3,,,,},

*/

■追加詞条信息map (Map<String, Object> 詞条信息map){
```

注

(随便一个词条名)
(就是描述目标词条的各个词条, 类型是List<Map>)
(这个SDP的一个实际的名字)
(词条From1)
(词条From2)
(词条To), 暂时一个SDP一个目标, 且只有一个目标

I ■处理W词条List

※首先W词条是个List<Map>的类型

1、循环遍历每个Map。判断其中的类型

※如果是SDP的CRUD。只对W词条.get(0)中的Map进行处理。

※如果是普通词条信息的CRUD, 就再说

1-1 取得Map下的每个Entry

1-2 如果Entry的Value是个String

就走【正常流程】, 然后返回数据的采番ID

1-3 如果Entry的Value是个List

就走【文件流程】, 然后返回数据的采番ID

1-4 做成W词条的词条信息Map<词条ID, 数据ID>

2、最后返回 List<词条信息Map> 作为【W词条信息Map_List】

※, 根据1.4的结果累积而成。

■取得对象Entry的Value的类型。如果是String, 即String
如果是List, 即List

■取得Entry的Key的值

■取得Map中每个Entry的信息 () 返回Key的List, 和value类型的List。

■追加W词条的数据_正常流程 (需要把词条名, 具体数据一起给入)

■调用 THINGSdb的接口即可

最后返回采番ID。※, 最好返回 (词条ID, 数据ID) 的Map。这样便于以后处理。

※, 此时只是单纯的给W词条追加数据, 并没有建立主W间的关系信息。

■追加W词条的数据_文件流程 (需要把词条名, 具体数据一起给入)

■调用 追加文件

1 将具体的值List, 存入到文件, 返回文件地址

2 将地址存入目标词条的采番地址

3 针对目标词条 (W词条) 进行数据采番

4 返回采番ID

最后返回采番ID。※, 最好返回 (词条ID, 数据ID) 的Map。这样便于以后处理。

II ■处理主词条（目标）

1、为主词条采番

如果词条都不存在，就直接先为主词条采番。这些交给【THINGSdb】

※，这个主词条自己是不带数据的，例如，【社员】。

|---它的信息都在没有W词条中，例如，【社员】的【姓名】。只要建立起【社员】与【姓名】的关系即可。

2、建立主词条与W词条的关系

※ 这里叫【主词条】而不叫【G词条】是为了明确位置关系。

※ 这里有两个方向，即给主词条加W词条的信息

给W词条加组词条的信息。

2-1 为主词条追加W词条关系信息

2-2 为W词条追加主词条关系信息

■为主词条追加W词条关系信息（主词条信息Map, W词条信息MapList）

首先你要有主词条（主词条信息Map<key=词条ID, Value=数据ID>）

遍历每个W词条（W词条信息Map<key=词条ID, Value=数据ID>）

调用 ■追加Waiter关系（G词条信息, W词条信息）

■为W词条追加主词条关系信息（主词条信息Map, W词条信息MapList）

首先你要有主词条（主词条信息Map<key=词条ID, Value=数据ID>）

遍历每个W词条（W词条信息Map<key=词条ID, Value=数据ID>）

调用 ■追加Guest关系（G词条信息, W词条信息）