## DataSet API

1. coGroup:

data1**.**coGroup**(**data2**)**

**.**where**(**1**)**

**.**equalTo**(**0**)**

**.**with**(new** CoGroupFunction**<**String**,** String**,** String**>()** **{**

**public** **void** **coGroup(**Iterable**<**String**>** in1**,** Iterable**<**String**>** in2**,** Collector**<**String**>** out**)** **{**

out**.**collect**(...);**

**}**

**});**

data1与data2做左外join操作，形象描述如下：

data1 left out join data2 on data1.value[1] = data2.value[0]

CoGroupFunction coGroup(in1，in2，out)

in1是一次left out join操作后data1集合,in2是一次left out join操作后data2集合，out是要输出的集合。

举例：

data1:

(

("user\_1", "song\_1", 100),

("user\_2", "song\_1", 10)

("user\_2", "song\_7", 10),

("user\_2", "song\_8", 3)

)

data2:

(

(song\_8)

)

coGroup(in1, in2, out)执行过程：

第一轮：

coGroup((("user\_2", "song\_7", 10),), (Empty, size=0), out)

第二轮：

coGroup((("user\_2", "song\_8", 3),), ((song\_8),), out)

第三轮：

coGroup((("user\_1", "song\_1", 100),("user\_2", "song\_1", 10),), (Empty, size=0), out) //因为data2不包含song\_1，所以in2为空的集合。而in1是song = song\_1的集合。

每一轮得到的out会合并在一起。

1. ReduceGroup

//对values执行某种操作，得到结果out，下面是将values的每个元素加1

data**.**reduceGroup**(new** GroupReduceFunction**<**Integer**,** Integer**>** **{**

**public** **void** **reduce(**Iterable**<**Integer**>** values**,** Collector**<**Integer**>** out**)** **{**

**int** prefixSum **=** 0**;**

**for** **(**Integer i **:** values**)** **{**

prefixSum **+=** i**;**

out**.**collect**(**prefixSum**);**

**}**

**}**

**});**