namespace java com.twitter.unified\_user\_actions.enricher.internal.thriftjava

#@namespace scala com.twitter.unified\_user\_actions.enricher.internal.thriftscala

#@namespace strato com.twitter.unified\_user\_actions.enricher.internal

/\*\*

\* An enrichment plan. It has multiple stages for different purposes during the enrichment process.

\*\*/

struct EnrichmentPlan {

1: required list<EnrichmentStage> stages

}(persisted='true', hasPersonalData='false')

/\*\*

\* A stage in the enrichment process with respect to the current key. Currently it can be of 2 options:

\* - re-partitioning on an id of type X

\* - hydrating metadata on an id of type X

\*

\* A stage also moves through different statues from initialized, processing until completion.

\* Each stage contains one or more instructions.

\*\*/

struct EnrichmentStage {

1: required EnrichmentStageStatus status

2: required EnrichmentStageType stageType

3: required list<EnrichmentInstruction> instructions

// The output topic for this stage. This information is not available when the stage was

// first setup, and it's only available after the driver has finished working on

// this stage.

4: optional string outputTopic

}(persisted='true', hasPersonalData='false')

/\*\*

\* The current processing status of a stage. It should either be done (completion) or not done (initialized).

\* Transient statuses such as "processing" is dangerous since we can't exactly be sure that has been done.

\*\*/

enum EnrichmentStageStatus {

Initialized = 0

Completion = 20

}

/\*\*

\* The type of processing in this stage. For example, repartioning the data or hydrating the data.

\*\*/

enum EnrichmentStageType {

Repartition = 0

Hydration = 10

}

enum EnrichmentInstruction {

// all enrichment based on a tweet id in UUA goes here

TweetEnrichment = 0

NotificationTweetEnrichment = 10

}