namespace java com.twitter.simclusters\_v2.thriftjava

namespace py gen.twitter.simclusters\_v2.evaluation

#@namespace scala com.twitter.simclusters\_v2.thriftscala

#@namespace strato com.twitter.simclusters\_v2

/\*\*

\* Surface area at which the reference tweet was displayed to the user

\*\*/

enum DisplayLocation {

TimelinesRecap = 1,

TimelinesRectweet = 2

}(hasPersonalData = 'false')

struct TweetLabels {

1: required bool isClicked = false(personalDataType = 'EngagementsPrivate')

2: required bool isLiked = false(personalDataType = 'EngagementsPublic')

3: required bool isRetweeted = false(personalDataType = 'EngagementsPublic')

4: required bool isQuoted = false(personalDataType = 'EngagementsPublic')

5: required bool isReplied = false(personalDataType = 'EngagementsPublic')

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

/\*\*

\* Data container of a reference tweet with scribed user engagement labels

\*/

struct ReferenceTweet {

1: required i64 tweetId(personalDataType = 'TweetId')

2: required i64 authorId(personalDataType = 'UserId')

3: required i64 timestamp(personalDataType = 'PublicTimestamp')

4: required DisplayLocation displayLocation

5: required TweetLabels labels

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

/\*\*

\* Data container of a candidate tweet generated by the candidate algorithm

\*/

struct CandidateTweet {

1: required i64 tweetId(personalDataType = 'TweetId')

2: optional double score(personalDataType = 'EngagementScore')

// The timestamp here is a synthetically generated timestamp.

// for evaluation purpose. Hence left unannotated

3: optional i64 timestamp

}(hasPersonalData = 'true')

/\*\*

\* An encapsulated collection of candidate tweets

\*\*/

struct CandidateTweets {

1: required i64 targetUserId(personalDataType = 'UserId')

2: required list<CandidateTweet> recommendedTweets

}(hasPersonalData = 'true')

/\*\*

\* An encapsulated collection of reference tweets

\*\*/

struct ReferenceTweets {

1: required i64 targetUserId(personalDataType = 'UserId')

2: required list<ReferenceTweet> impressedTweets

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

/\*\*

\* A list of candidate tweets

\*\*/

struct CandidateTweetsList {

1: required list<CandidateTweet> recommendedTweets

}(hasPersonalData = 'true')