namespace java com.twitter.simclusters\_v2.thriftjava

namespace py gen.twitter.simclusters\_v2.offline\_job\_internal

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

#@namespace strato com.twitter.simclusters\_v2

include "com/twitter/algebird\_internal/algebird.thrift"

// For internal usage only. Mainly for offline\_evaluation.

// Deprecated. Please use 'online\_store/ModelVersion'

enum PersistedModelVersion {

MODEL\_20M\_145K\_dec11 = 1,

MODEL\_20M\_145K\_updated = 2,

MODEL\_20M\_145K\_2020 = 3,

RESERVED\_4 = 4,

RESERVED\_5 = 5

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

enum PersistedScoreType {

NORMALIZED\_FAV\_8\_HR\_HALF\_LIFE = 1,

NORMALIZED\_FOLLOW\_8\_HR\_HALF\_LIFE = 2,

NORMALIZED\_LOG\_FAV\_8\_HR\_HALF\_LIFE = 3,

RESERVED\_4 = 4,

RESERVED\_5 = 5

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

struct PersistedScores {

1: optional algebird.DecayedValue score

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

struct TweetAndClusterScores {

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

2: required i32 clusterId(personalDataType = 'InferredInterests')

3: required PersistedModelVersion modelVersion

4: required PersistedScores scores(personalDataType = 'EngagementScore')

5: optional PersistedScoreType scoreType

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

struct TweetTopKClustersWithScores {

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

2: required PersistedModelVersion modelVersion

3: required map<i32, PersistedScores> topKClusters(personalDataTypeKey = 'InferredInterests')

4: optional PersistedScoreType scoreType

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

struct ClusterTopKTweetsWithScores {

1: required i32 clusterId(personalDataType = 'InferredInterests')

2: required PersistedModelVersion modelVersion

3: required map<i64, PersistedScores> topKTweets(personalDataTypeKey = 'TweetId')

4: optional PersistedScoreType scoreType

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

struct QueryAndClusterScores {

1: required string query(personalDataType = 'SearchQuery')

2: required i32 clusterId

3: required PersistedModelVersion modelVersion

4: required PersistedScores scores

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

struct QueryTopKClustersWithScores {

1: required string query(personalDataType = 'SearchQuery')

2: required PersistedModelVersion modelVersion

3: required map<i32, PersistedScores> topKClusters

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