namespace java com.twitter.simclusters\_v2.thriftjava

namespace py gen.twitter.simclusters\_v2.embedding

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

#@namespace strato com.twitter.simclusters\_v2

include "com/twitter/simclusters\_v2/identifier.thrift"

include "com/twitter/simclusters\_v2/online\_store.thrift"

struct SimClusterWithScore {

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

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

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

struct TopSimClustersWithScore {

1: required list<SimClusterWithScore> topClusters

2: required online\_store.ModelVersion modelVersion

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

struct InternalIdWithScore {

1: required identifier.InternalId internalId

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

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

struct InternalIdEmbedding {

1: required list<InternalIdWithScore> embedding

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

struct SemanticCoreEntityWithScore {

1: required i64 entityId(personalDataType = 'SemanticcoreClassification')

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

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

struct TopSemanticCoreEntitiesWithScore {

1: required list<SemanticCoreEntityWithScore> topEntities

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

struct PersistedFullClusterId {

1: required online\_store.ModelVersion modelVersion

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

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

struct DayPartitionedClusterId {

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

2: required string dayPartition // format: yyyy-MM-dd

}

struct TopProducerWithScore {

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

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

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

struct TopProducersWithScore {

1: required list<TopProducerWithScore> topProducers

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

struct TweetWithScore {

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

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

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

struct TweetsWithScore {

1: required list<TweetWithScore> tweets

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

struct TweetTopKTweetsWithScore {

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

2: required TweetsWithScore topkTweetsWithScore

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

/\*\*

\* The generic SimClustersEmbedding for online long-term storage and real-time calculation.

\* Use SimClustersEmbeddingId as the only identifier.

\* Warning: Doesn't include model version and embedding type in the value struct.

\*\*/

struct SimClustersEmbedding {

1: required list<SimClusterWithScore> embedding

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

struct SimClustersEmbeddingWithScore {

1: required SimClustersEmbedding embedding

2: required double score

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

/\*\*

\* This is the recommended structure for aggregating embeddings with time decay - the metadata

\* stores the information needed for decayed aggregation.

\*\*/

struct SimClustersEmbeddingWithMetadata {

1: required SimClustersEmbedding embedding

2: required SimClustersEmbeddingMetadata metadata

}(hasPersonalData = 'true')

struct SimClustersEmbeddingIdWithScore {

1: required identifier.SimClustersEmbeddingId id

2: required double score

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

struct SimClustersMultiEmbeddingByValues {

1: required list<SimClustersEmbeddingWithScore> embeddings

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

struct SimClustersMultiEmbeddingByIds {

1: required list<SimClustersEmbeddingIdWithScore> ids

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

/\*\*

\* Generic SimClusters Multiple Embeddings. The identifier.SimClustersMultiEmbeddingId is the key of

\* the multiple embedding.

\*\*/

union SimClustersMultiEmbedding {

1: SimClustersMultiEmbeddingByValues values

2: SimClustersMultiEmbeddingByIds ids

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

/\*\*

\* The metadata of a SimClustersEmbedding. The updatedCount represent the version of the Embedding.

\* For tweet embedding, the updatedCount is same/close to the favorite count.

\*\*/

struct SimClustersEmbeddingMetadata {

1: optional i64 updatedAtMs

2: optional i64 updatedCount

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

/\*\*

\* The data structure for PersistentSimClustersEmbedding Store

\*\*/

struct PersistentSimClustersEmbedding {

1: required SimClustersEmbedding embedding

2: required SimClustersEmbeddingMetadata metadata

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

/\*\*

\* The data structure for the Multi Model PersistentSimClustersEmbedding Store

\*\*/

struct MultiModelPersistentSimClustersEmbedding {

1: required map<online\_store.ModelVersion, PersistentSimClustersEmbedding> multiModelPersistentSimClustersEmbedding

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