namespace java com.twitter.simclusters\_v2.thriftjava

namespace py gen.twitter.simclusters\_v2.identifier

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

#@namespace strato com.twitter.simclusters\_v2

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

/\*\*

\* The uniform type for a SimClusters Embeddings.

\* Each embeddings have the uniform underlying storage.

\* Warning: Every EmbeddingType should map to one and only one InternalId.

\*\*/

enum EmbeddingType {

// Reserve 001 - 99 for Tweet embeddings

FavBasedTweet = 1, // Deprecated

FollowBasedTweet = 2, // Deprecated

LogFavBasedTweet = 3, // Production Version

FavBasedTwistlyTweet = 10, // Deprecated

LogFavBasedTwistlyTweet = 11, // Deprecated

LogFavLongestL2EmbeddingTweet = 12, // Production Version

// Tweet embeddings generated from non-fav events

// Naming convention: {Event}{Score}BasedTweet

// {Event}: The interaction event we use to build the tweet embeddings

// {Score}: The score from user InterestedIn embeddings

VideoPlayBack50LogFavBasedTweet = 21,

RetweetLogFavBasedTweet = 22,

ReplyLogFavBasedTweet = 23,

PushOpenLogFavBasedTweet = 24,

// [Experimental] Offline generated FavThroughRate-based Tweet Embedding

Pop1000RankDecay11Tweet = 30,

Pop10000RankDecay11Tweet = 31,

OonPop1000RankDecayTweet = 32,

// [Experimental] Offline generated production-like LogFavScore-based Tweet Embedding

OfflineGeneratedLogFavBasedTweet = 40,

// Reserve 51-59 for Ads Embedding

LogFavBasedAdsTweet = 51, // Experimental embedding for ads tweet candidate

LogFavClickBasedAdsTweet = 52, // Experimental embedding for ads tweet candidate

// Reserve 60-69 for Evergreen content

LogFavBasedEvergreenTweet = 60,

LogFavBasedRealTimeTweet = 65,

// Reserve 101 to 149 for Semantic Core Entity embeddings

FavBasedSematicCoreEntity = 101, // Deprecated

FollowBasedSematicCoreEntity = 102, // Deprecated

FavBasedHashtagEntity = 103, // Deprecated

FollowBasedHashtagEntity = 104, // Deprecated

ProducerFavBasedSemanticCoreEntity = 105, // Deprecated

ProducerFollowBasedSemanticCoreEntity = 106,// Deprecated

FavBasedLocaleSemanticCoreEntity = 107, // Deprecated

FollowBasedLocaleSemanticCoreEntity = 108, // Deprecated

LogFavBasedLocaleSemanticCoreEntity = 109, // Deprecated

LanguageFilteredProducerFavBasedSemanticCoreEntity = 110, // Deprecated

LanguageFilteredFavBasedLocaleSemanticCoreEntity = 111, // Deprecated

FavTfgTopic = 112, // TFG topic embedding built from fav-based user interestedIn

LogFavTfgTopic = 113, // TFG topic embedding built from logfav-based user interestedIn

FavInferredLanguageTfgTopic = 114, // TFG topic embedding built using inferred consumed languages

FavBasedKgoApeTopic = 115, // topic embedding using fav-based aggregatable producer embedding of KGO seed accounts.

LogFavBasedKgoApeTopic = 116, // topic embedding using log fav-based aggregatable producer embedding of KGO seed accounts.

FavBasedOnboardingApeTopic = 117, // topic embedding using fav-based aggregatable producer embedding of onboarding seed accounts.

LogFavBasedOnboardingApeTopic = 118, // topic embedding using log fav-based aggregatable producer embedding of onboarding seed accounts.

LogFavApeBasedMuseTopic = 119, // Deprecated

LogFavApeBasedMuseTopicExperiment = 120 // Deprecated

// Reserved 201 - 299 for Producer embeddings (KnownFor)

FavBasedProducer = 201

FollowBasedProducer = 202

AggregatableFavBasedProducer = 203 // fav-based aggregatable producer embedding.

AggregatableLogFavBasedProducer = 204 // logfav-based aggregatable producer embedding.

RelaxedAggregatableLogFavBasedProducer = 205 // logfav-based aggregatable producer embedding.

AggregatableFollowBasedProducer = 206 // follow-based aggregatable producer embedding.

KnownFor = 300

// Reserved 301 - 399 for User InterestedIn embeddings

FavBasedUserInterestedIn = 301

FollowBasedUserInterestedIn = 302

LogFavBasedUserInterestedIn = 303

RecentFollowBasedUserInterestedIn = 304 // interested-in embedding based on aggregating producer embeddings of recent follows

FilteredUserInterestedIn = 305 // interested-in embedding used by twistly read path

LogFavBasedUserInterestedInFromAPE = 306

FollowBasedUserInterestedInFromAPE = 307

TwiceUserInterestedIn = 308 // interested-in multi-embedding based on clustering producer embeddings of neighbors

UnfilteredUserInterestedIn = 309

UserNextInterestedIn = 310 // next interested-in embedding generated from BeT

// Denser User InterestedIn, generated by Producer embeddings.

FavBasedUserInterestedInFromPE = 311

FollowBasedUserInterestedInFromPE = 312

LogFavBasedUserInterestedInFromPE = 313

FilteredUserInterestedInFromPE = 314 // interested-in embedding used by twistly read path

// [Experimental] Denser User InterestedIn, generated by aggregating IIAPE embedding from AddressBook

LogFavBasedUserInterestedMaxpoolingAddressBookFromIIAPE = 320

LogFavBasedUserInterestedAverageAddressBookFromIIAPE = 321

LogFavBasedUserInterestedBooktypeMaxpoolingAddressBookFromIIAPE = 322

LogFavBasedUserInterestedLargestDimMaxpoolingAddressBookFromIIAPE = 323

LogFavBasedUserInterestedLouvainMaxpoolingAddressBookFromIIAPE = 324

LogFavBasedUserInterestedConnectedMaxpoolingAddressBookFromIIAPE = 325

//Reserved 401 - 500 for Space embedding

FavBasedApeSpace = 401 // DEPRECATED

LogFavBasedListenerSpace = 402 // DEPRECATED

LogFavBasedAPESpeakerSpace = 403 // DEPRECATED

LogFavBasedUserInterestedInListenerSpace = 404 // DEPRECATED

// Experimental, internal-only IDs

ExperimentalThirtyDayRecentFollowBasedUserInterestedIn = 10000 // Like RecentFollowBasedUserInterestedIn, except limited to last 30 days

ExperimentalLogFavLongestL2EmbeddingTweet = 10001 // DEPRECATED

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

/\*\*

\* The uniform type for a SimClusters MultiEmbeddings.

\* Warning: Every MultiEmbeddingType should map to one and only one InternalId.

\*\*/

enum MultiEmbeddingType {

// Reserved 0-99 for Tweet based MultiEmbedding

// Reserved 100 - 199 for Topic based MultiEmbedding

LogFavApeBasedMuseTopic = 100 // Deprecated

LogFavApeBasedMuseTopicExperiment = 101 // Deprecated

// Reserved 301 - 399 for User InterestedIn embeddings

TwiceUserInterestedIn = 301 // interested-in multi-embedding based on clustering producer embeddings of neighbors

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

// Deprecated. Please use TopicId for future cases.

struct LocaleEntityId {

1: i64 entityId

2: string language

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

enum EngagementType {

Favorite = 1,

Retweet = 2,

}

struct UserEngagedTweetId {

1: i64 tweetId(personalDataType = 'TweetId')

2: i64 userId(personalDataType = 'UserId')

3: EngagementType engagementType(personalDataType = 'EventType')

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

struct TopicId {

1: i64 entityId (personalDataType = 'SemanticcoreClassification')

// 2-letter ISO 639-1 language code

2: optional string language

// 2-letter ISO 3166-1 alpha-2 country code

3: optional string country

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

struct TopicSubId {

1: i64 entityId (personalDataType = 'SemanticcoreClassification')

// 2-letter ISO 639-1 language code

2: optional string language

// 2-letter ISO 3166-1 alpha-2 country code

3: optional string country

4: i32 subId

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

// Will be used for testing purposes in DDG 15536, 15534

struct UserWithLanguageId {

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

2: optional string langCode(personalDataType = 'InferredLanguage')

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

/\*\*

\* The internal identifier type.

\* Need to add ordering in [[com.twitter.simclusters\_v2.common.SimClustersEmbeddingId]]

\* when adding a new type.

\*\*/

union InternalId {

1: i64 tweetId(personalDataType = 'TweetId')

2: i64 userId(personalDataType = 'UserId')

3: i64 entityId(personalDataType = 'SemanticcoreClassification')

4: string hashtag(personalDataType = 'PublicTweetEntitiesAndMetadata')

5: i32 clusterId

6: LocaleEntityId localeEntityId(personalDataType = 'SemanticcoreClassification')

7: UserEngagedTweetId userEngagedTweetId

8: TopicId topicId

9: TopicSubId topicSubId

10: string spaceId

11: UserWithLanguageId userWithLanguageId

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

/\*\*

\* A uniform identifier type for all kinds of SimClusters based embeddings.

\*\*/

struct SimClustersEmbeddingId {

1: required EmbeddingType embeddingType

2: required online\_store.ModelVersion modelVersion

3: required InternalId internalId

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

/\*\*

\* A uniform identifier type for multiple SimClusters embeddings

\*\*/

struct SimClustersMultiEmbeddingId {

1: required MultiEmbeddingType embeddingType

2: required online\_store.ModelVersion modelVersion

3: required InternalId internalId

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