namespace java com.twitter.follow\_recommendations.thriftjava

#@namespace scala com.twitter.follow\_recommendations.thriftscala

#@namespace strato com.twitter.follow\_recommendations

include "com/twitter/ml/api/data.thrift"

struct CandidateSourceDetails {

1: optional map<string, double> candidateSourceScores

2: optional i32 primarySource

3: optional map<string, i32> candidateSourceRanks

}(hasPersonalData='false')

struct Score {

1: required double value

2: optional string rankerId

3: optional string scoreType

}(hasPersonalData='false')

// Contains (1) the ML-based heavy ranker and score (2) scores and rankers in producer experiment framework

struct Scores {

1: required list<Score> scores

2: optional string selectedRankerId

3: required bool isInProducerScoringExperiment

}(hasPersonalData='false')

struct RankingInfo {

1: optional Scores scores

2: optional i32 rank

}(hasPersonalData='false')

// this encapsulates all information related to the ranking process from generation to scoring

struct ScoringDetails {

1: optional CandidateSourceDetails candidateSourceDetails

2: optional double score

3: optional data.DataRecord dataRecord

4: optional list<string> rankerIds

5: optional DebugDataRecord debugDataRecord // this field is not logged as it's only used for debugging

6: optional map<string, RankingInfo> infoPerRankingStage // scoring and ranking info per ranking stage

}(hasPersonalData='true')

// exactly the same as a data record, except that we store the feature name instead of the id

struct DebugDataRecord {

1: optional set<string> binaryFeatures; // stores BINARY features

2: optional map<string, double> continuousFeatures; // stores CONTINUOUS features

3: optional map<string, i64> discreteFeatures; // stores DISCRETE features

4: optional map<string, string> stringFeatures; // stores STRING features

5: optional map<string, set<string>> sparseBinaryFeatures; // stores sparse BINARY features

6: optional map<string, map<string, double>> sparseContinuousFeatures; // sparse CONTINUOUS features

}(hasPersonalData='true')