namespace java com.twitter.follow\_recommendations.logging.thriftjava

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

#@namespace strato com.twitter.follow\_recommendations.logging

include "client\_context.thrift"

include "debug.thrift"

include "display\_context.thrift"

include "display\_location.thrift"

include "recommendations.thrift"

struct OfflineRecommendationRequest {

1: required client\_context.OfflineClientContext clientContext

2: required display\_location.OfflineDisplayLocation displayLocation

3: optional display\_context.OfflineDisplayContext displayContext

4: optional i32 maxResults

5: optional string cursor

6: optional list<i64> excludedIds(personalDataType='UserId')

7: optional bool fetchPromotedContent

8: optional debug.OfflineDebugParams debugParams

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

struct OfflineRecommendationResponse {

1: required list<recommendations.OfflineRecommendation> recommendations

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

struct RecommendationLog {

1: required OfflineRecommendationRequest request

2: required OfflineRecommendationResponse response

3: required i64 timestampMs

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

struct OfflineScoringUserRequest {

1: required client\_context.OfflineClientContext clientContext

2: required display\_location.OfflineDisplayLocation displayLocation

3: required list<recommendations.OfflineUserRecommendation> candidates

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

struct OfflineScoringUserResponse {

1: required list<recommendations.OfflineUserRecommendation> candidates

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

struct ScoredUsersLog {

1: required OfflineScoringUserRequest request

2: required OfflineScoringUserResponse response

3: required i64 timestampMs

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

struct OfflineRecommendationFlowUserMetadata {

1: optional i32 userSignupAge(personalDataType = 'AgeOfAccount')

2: optional string userState(personalDataType = 'UserState')

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

struct OfflineRecommendationFlowSignals {

1: optional string countryCode(personalDataType='InferredCountry')

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

struct OfflineRecommendationFlowCandidateSourceCandidates {

1: required string candidateSourceName

2: required list<i64> candidateUserIds(personalDataType='UserId')

3: optional list<double> candidateUserScores

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

struct RecommendationFlowLog {

1: required client\_context.OfflineClientContext clientContext

2: optional OfflineRecommendationFlowUserMetadata userMetadata

3: optional OfflineRecommendationFlowSignals signals

4: required i64 timestampMs

5: required string recommendationFlowIdentifier

6: optional list<OfflineRecommendationFlowCandidateSourceCandidates> filteredCandidates

7: optional list<OfflineRecommendationFlowCandidateSourceCandidates> rankedCandidates

8: optional list<OfflineRecommendationFlowCandidateSourceCandidates> truncatedCandidates

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