namespace java com.twitter.recos.thriftjava

#@namespace scala com.twitter.recos.thriftscala

namespace rb Recos

include "com/twitter/recos/features/tweet.thrift"

enum RecommendTweetDisplayLocation {

HomeTimeline = 0

Peek = 1

WelcomeFlow = 2

NetworkDigest = 3

BackfillDigest = 4

NetworkDigestExp1 = 5

NetworkDigestExp2 = 6 // deprecated

NetworkDigestExp3 = 7 // deprecated

HttpEndpoint = 8

HomeTimeline1 = 9

HomeTimeline2 = 10

HomeTimeline3 = 11

HomeTimeline4 = 12

Poptart = 13

NetworkDigestExp4 = 14

NetworkDigestExp5 = 15

NetworkDigestExp6 = 16

NetworkDigestExp7 = 17

NetworkDigestExp8 = 18

NetworkDigestExp9 = 19

InstantTimeline1 = 20 // AB1 + whitelist

InstantTimeline2 = 21 // AB1 + !whitelist

InstantTimeline3 = 22 // AB2 + whitelist

InstantTimeline4 = 23 // AB2 + !whitelist

BackfillDigestActive = 24 // deprecated

BackfillDigestDormant = 25 // deprecated

ExploreUS = 26 // deprecated

ExploreBR = 27 // deprecated

ExploreIN = 28 // deprecated

ExploreES = 29 // deprecated

ExploreJP = 30 // deprecated

MagicRecs = 31

MagicRecs1 = 32

MagicRecs2 = 33

MagicRecs3 = 34

SMSDiscover = 35

FastFollower = 36

InstantTimeline5 = 37 // for instant timeline experiment

InstantTimeline6 = 38 // for instant timeline experiment

InstantTimeline7 = 39 // for instant timeline experiment

InstantTimeline8 = 40 // for instant timeline experiment

LoggedOutProfile = 41

LoggedOutPermalink = 42

Poptart2 = 43

}

enum RelatedTweetDisplayLocation {

Permalink = 0

Permalink1 = 1

MobilePermalink = 2

Permalink3 = 3

Permalink4 = 4

RelatedTweets = 5

RelatedTweets1 = 6

RelatedTweets2 = 7

RelatedTweets3 = 8

RelatedTweets4 = 9

LoggedOutProfile = 10

LoggedOutPermalink = 11

}

enum DDGBucket {

Control = 0

Treatment = 1

None = 2

}

struct RecommendTweetRequest {

1: required i64 requesterId // user id of the requesting user

2: required RecommendTweetDisplayLocation displayLocation // display location from the client

3: optional i64 clientId // twitter api client id

4: optional i32 maxResults // number of suggested results to return

5: optional list<i64> excludedTweetIds // list of tweet ids to exclude from response

6: optional list<i64> excludedAuthorIds // list of author ids to exclude from response

7: optional i64 guestId // guestId

8: optional string languageCode // Language code

9: optional string countryCode // Country code

10: optional string ipAddress // ip address of the user

11: optional string deviceId // udid/uuid of device

12: optional bool populateTweetFeatures // whether to populate tweet features. RecommendedTweet.tweetFeatures in the response will only be populated if this is set.

}

struct Bucket {

1: required string experimentName // name of experiment (or not). experiment could be production or whatever fits

2: required string bucket // name of bucket (may or may not be a DDG bucket, e.g., production)

}

struct RelatedTweetRequest {

1: required i64 tweetId // original tweet id

2: required RelatedTweetDisplayLocation displayLocation // display location from the client

3: optional i64 clientId // twitter api client id

4: optional i64 requesterId // user id of the requesting user

5: optional i32 maxResults // number of suggested results to return

6: optional list<i64> excludeTweetIds // list of tweet ids to exclude from response

7: optional list<i64> excludedAuthorIds // list of author ids to exclude from response

8: optional i64 guestId // guestId

9: optional string languageCode // Language code

10: optional string countryCode // Country code

11: optional string ipAddress // ip address of the user

12: optional string deviceId // udid/uuid of device

13: optional string userAgent // userAgent of the requesting user

}

enum SocialProofType {

FollowedBy = 1,

FavoritedBy = 2,

RetweetedBy = 3,

SimilarTo = 4,

RESERVED\_2 = 5,

RESERVED\_3 = 6,

RESERVED\_4 = 7,

RESERVED\_5 = 8,

RESERVED\_6 = 9,

RESERVED\_7 = 10

}

enum Algorithm {

Salsa = 1,

PastEmailClicks = 2,

SimilarToEmailClicks = 3,

PastClientEventClicks = 4,

VitNews = 5,

StrongTieScoring = 6,

PollsFromGraph = 7,

PollsBasedOnGeo = 8,

RESERVED\_9 = 9,

RESERVED\_10 = 10,

RESERVED\_11 = 11,

}

struct RecommendedTweet {

1: required i64 tweetId

2: required i64 authorId

3: required list<i64> socialProof

4: required string feedbackToken

5: optional list<i64> favBy // optionally provide a list of users who fav'ed the tweet if exist

6: optional tweet.RecommendedTweetFeatures tweetFeatures // the features of a recommended tweet

7: optional SocialProofType socialProofType // type of social proof. favBy should be deprecated soon

8: optional string socialProofOverride // should be set only for DDGs, for en-only experiments. SocialProofType is ignored when this field is set

9: optional Algorithm algorithm // algorithm used

10: optional double score // score

11: optional bool isFollowingAuthor // true if the target user follows the author of the tweet

}

struct RelatedTweet {

1: required i64 tweetId

2: required i64 authorId

3: required double score

4: required string feedbackToken

}

struct RecommendTweetResponse {

1: required list<RecommendedTweet> tweets

2: optional DDGBucket bucket // deprecated

3: optional Bucket assignedBucket // for client-side experimentation

}

struct RelatedTweetResponse {

1: required list<RelatedTweet> tweets // a list of related tweets

2: optional Bucket assignedBucket // the bucket used for treatment

}

/\*\*

\* The main interface-definition for Recos.

\*/

service Recos {

RecommendTweetResponse recommendTweets (RecommendTweetRequest request)

RelatedTweetResponse relatedTweets (RelatedTweetRequest request)

}