package com.twitter.recos.user\_tweet\_entity\_graph

import com.twitter.finagle.thrift.ClientId

import com.twitter.finagle.tracing.{Trace, TraceId}

import com.twitter.recos.user\_tweet\_entity\_graph.thriftscala.\_

import com.twitter.util.Future

object UserTweetEntityGraph {

def traceId: TraceId = Trace.id

def clientId: Option[ClientId] = ClientId.current

}

class UserTweetEntityGraph(

recommendationHandler: RecommendationHandler,

tweetSocialProofHandler: TweetSocialProofHandler,

socialProofHandler: SocialProofHandler)

extends thriftscala.UserTweetEntityGraph.MethodPerEndpoint {

override def recommendTweets(

request: RecommendTweetEntityRequest

): Future[RecommendTweetEntityResponse] = recommendationHandler(request)

/\*\*

\* Given a query user, its seed users, and a set of input tweets, return the social proofs of

\* input tweets if any.

\*

\* Currently this supports clients such as Email Recommendations, MagicRecs, and HomeTimeline.

\* In order to avoid heavy migration work, we are retaining this endpoint.

\*/

override def findTweetSocialProofs(

request: SocialProofRequest

): Future[SocialProofResponse] = tweetSocialProofHandler(request)

/\*\*

\* Find social proof for the specified RecommendationType given a set of input ids of that type.

\* Only find social proofs from the specified seed users with the specified social proof types.

\*

\* Currently this supports url social proof generation for Guide.

\*

\* This endpoint is flexible enough to support social proof generation for all recommendation

\* types, and should be used for all future clients of this service.

\*/

override def findRecommendationSocialProofs(

request: RecommendationSocialProofRequest

): Future[RecommendationSocialProofResponse] = socialProofHandler(request)

}