package com.twitter.timelineranker.server

import com.twitter.thriftwebforms.MethodOptions

import com.twitter.thriftwebforms.view.ServiceResponseView

import com.twitter.timelineranker.{thriftscala => thrift}

import com.twitter.util.Future

object TimelineRankerThriftWebForms {

private def renderTweetIds(tweetIDs: Seq[Long]): Future[ServiceResponseView] = {

val html = tweetIDs.map { tweetID =>

s"""<blockquote class="twitter-tweet"><a href="https://twitter.com/tweet/statuses/$tweetID"></a></blockquote>"""

}.mkString

Future.value(

ServiceResponseView(

"Tweets",

html,

Seq("//platform.twitter.com/widgets.js")

)

)

}

private def renderGetCandidateTweetsResponse(r: AnyRef): Future[ServiceResponseView] = {

val responses = r.asInstanceOf[Seq[thrift.GetCandidateTweetsResponse]]

val tweetIds = responses.flatMap(

\_.candidates.map(\_.flatMap(\_.tweet.map(\_.id))).getOrElse(Nil)

)

renderTweetIds(tweetIds)

}

def methodOptions: Map[String, MethodOptions] =

Map(

thrift.TimelineRanker.GetRecycledTweetCandidates.name -> MethodOptions(

responseRenderers = Seq(renderGetCandidateTweetsResponse)

),

thrift.TimelineRanker.HydrateTweetCandidates.name -> MethodOptions(

responseRenderers = Seq(renderGetCandidateTweetsResponse)

),

thrift.TimelineRanker.GetRecapCandidatesFromAuthors.name -> MethodOptions(

responseRenderers = Seq(renderGetCandidateTweetsResponse)

),

thrift.TimelineRanker.GetEntityTweetCandidates.name -> MethodOptions(

responseRenderers = Seq(renderGetCandidateTweetsResponse)

)

)

}