package com.twitter.timelineranker.parameters.util

import com.twitter.servo.util.Gate

import com.twitter.timelines.configapi.BaseRequestContext

import com.twitter.timelines.configapi.WithExperimentContext

import com.twitter.timelines.configapi.WithFeatureContext

import com.twitter.timelines.configapi.WithUserId

import com.twitter.timelines.model.UserId

import com.twitter.timelineservice.DeviceContext

import com.twitter.timelineservice.model.RequestContextFactory

import com.twitter.util.Future

trait CommonRequestContext

extends BaseRequestContext

with WithExperimentContext

with WithUserId

with WithFeatureContext

trait RequestContextBuilder {

def apply(

recipientUserId: Option[UserId],

deviceContext: Option[DeviceContext]

): Future[CommonRequestContext]

}

class RequestContextBuilderImpl(requestContextFactory: RequestContextFactory)

extends RequestContextBuilder {

override def apply(

recipientUserId: Option[UserId],

deviceContextOpt: Option[DeviceContext]

): Future[CommonRequestContext] = {

val requestContextFut = requestContextFactory(

contextualUserIdOpt = recipientUserId,

deviceContext = deviceContextOpt.getOrElse(DeviceContext.empty),

experimentConfigurationOpt = None,

requestLogOpt = None,

contextualUserContext = None,

useRolesCache = Gate.True,

timelineId = None

)

requestContextFut.map { requestContext =>

new CommonRequestContext {

override val userId = recipientUserId

override val experimentContext = requestContext.experimentContext

override val featureContext = requestContext.featureContext

}

}

}

}