package com.twitter.follow\_recommendations.configapi.candidates

import com.google.common.annotations.VisibleForTesting

import com.google.inject.Inject

import com.twitter.decider.Decider

import com.twitter.featureswitches.v2.FeatureSwitches

import com.twitter.featureswitches.{Recipient => FeatureSwitchRecipient}

import com.twitter.follow\_recommendations.common.constants.GuiceNamedConstants.PRODUCER\_SIDE\_FEATURE\_SWITCHES

import com.twitter.follow\_recommendations.common.models.CandidateUser

import com.twitter.follow\_recommendations.common.models.DisplayLocation

import com.twitter.timelines.configapi.FeatureContext

import com.twitter.timelines.configapi.featureswitches.v2.FeatureSwitchResultsFeatureContext

import javax.inject.Named

import javax.inject.Singleton

@Singleton

class CandidateUserContextFactory @Inject() (

@Named(PRODUCER\_SIDE\_FEATURE\_SWITCHES) featureSwitches: FeatureSwitches,

decider: Decider) {

def apply(

candidateUser: CandidateUser,

displayLocation: DisplayLocation

): CandidateUserContext = {

val featureContext = getFeatureContext(candidateUser, displayLocation)

CandidateUserContext(Some(candidateUser.id), featureContext)

}

private[configapi] def getFeatureContext(

candidateUser: CandidateUser,

displayLocation: DisplayLocation

): FeatureContext = {

val recipient = getFeatureSwitchRecipient(candidateUser).withCustomFields(

"display\_location" -> displayLocation.toFsName)

new FeatureSwitchResultsFeatureContext(featureSwitches.matchRecipient(recipient))

}

@VisibleForTesting

private[configapi] def getFeatureSwitchRecipient(

candidateUser: CandidateUser

): FeatureSwitchRecipient = {

FeatureSwitchRecipient(

userId = Some(candidateUser.id),

userRoles = None,

deviceId = None,

guestId = None,

languageCode = None,

countryCode = None,

isVerified = None,

clientApplicationId = None,

isTwoffice = None

)

}

}