package com.twitter.product\_mixer.component\_library.feature.featurestorev1

import com.twitter.ml.featurestore.catalog.entities

import com.twitter.ml.featurestore.lib.EntityId

import com.twitter.ml.featurestore.lib.UserId

import com.twitter.ml.featurestore.lib.entity.Entity

import com.twitter.ml.featurestore.lib.entity.EntityWithId

import com.twitter.ml.featurestore.lib.feature.{Feature => FSv1Feature}

import com.twitter.product\_mixer.component\_library.model.candidate.BaseUserCandidate

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMap

import com.twitter.product\_mixer.core.feature.featurestorev1.\_

import com.twitter.product\_mixer.core.pipeline.PipelineQuery

import com.twitter.timelines.configapi.FSParam

object FeatureStoreV1UserCandidateUserIdFeature {

def apply[Query <: PipelineQuery, Candidate <: BaseUserCandidate, Value](

feature: FSv1Feature[UserId, Value],

legacyName: Option[String] = None,

defaultValue: Option[Value] = None,

enabledParam: Option[FSParam[Boolean]] = None

): FeatureStoreV1CandidateFeature[Query, Candidate, \_ <: EntityId, Value] =

FeatureStoreV1CandidateFeature(

feature,

UserCandidateUserIdEntity,

legacyName,

defaultValue,

enabledParam)

}

object UserCandidateUserIdEntity

extends FeatureStoreV1CandidateEntity[PipelineQuery, BaseUserCandidate, UserId] {

override val entity: Entity[UserId] = entities.core.User

override def entityWithId(

query: PipelineQuery,

user: BaseUserCandidate,

existingFeatures: FeatureMap

): EntityWithId[UserId] =

entity.withId(UserId(user.id))

}