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

import com.twitter.ml.api.transform.FeatureRenameTransform

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

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

import com.twitter.ml.featurestore.lib.TweetId

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

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

import com.twitter.ml.featurestore.lib.feature.TimelinesAggregationFrameworkFeatureGroup

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

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

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 FeatureStoreV1TweetCandidateTweetIdFeature {

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

feature: FSv1Feature[TweetId, Value],

legacyName: Option[String] = None,

defaultValue: Option[Value] = None,

enabledParam: Option[FSParam[Boolean]] = None

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

FeatureStoreV1CandidateFeature(

feature,

TweetCandidateTweetIdEntity,

legacyName,

defaultValue,

enabledParam)

}

object FeatureStoreV1TweetCandidateTweetIdAggregateFeature {

def apply[Query <: PipelineQuery, Candidate <: BaseTweetCandidate](

featureGroup: TimelinesAggregationFrameworkFeatureGroup[TweetId],

enabledParam: Option[FSParam[Boolean]] = None,

keepLegacyNames: Boolean = false,

featureNameTransform: Option[FeatureRenameTransform] = None

): FeatureStoreV1CandidateFeatureGroup[Query, Candidate, \_ <: EntityId] =

FeatureStoreV1CandidateFeatureGroup(

featureGroup,

TweetCandidateTweetIdEntity,

enabledParam,

keepLegacyNames,

featureNameTransform

)

}

object TweetCandidateTweetIdEntity

extends FeatureStoreV1CandidateEntity[PipelineQuery, BaseTweetCandidate, TweetId] {

override val entity: Entity[TweetId] = entities.core.Tweet

override def entityWithId(

query: PipelineQuery,

tweet: BaseTweetCandidate,

existingFeatures: FeatureMap

): EntityWithId[TweetId] =

entity.withId(TweetId(tweet.id))

}