package com.twitter.product\_mixer.core.feature.featurestorev1

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

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

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

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

import com.twitter.product\_mixer.core.model.common.UniversalNoun

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

sealed trait FeatureStoreV1Entity[

-Query <: PipelineQuery,

-Input,

FeatureStoreEntityId <: EntityId] {

val entity: Entity[FeatureStoreEntityId]

}

trait FeatureStoreV1QueryEntity[-Query <: PipelineQuery, FeatureStoreEntityId <: EntityId]

extends FeatureStoreV1Entity[Query, Query, FeatureStoreEntityId] {

def entityWithId(query: Query): EntityWithId[FeatureStoreEntityId]

}

trait FeatureStoreV1CandidateEntity[

-Query <: PipelineQuery,

-Input <: UniversalNoun[Any],

FeatureStoreEntityId <: EntityId]

extends FeatureStoreV1Entity[Query, Input, FeatureStoreEntityId] {

def entityWithId(

query: Query,

input: Input,

existingFeatures: FeatureMap

): EntityWithId[FeatureStoreEntityId]

}