package com.twitter.product\_mixer.core.functional\_component.feature\_hydrator

import com.twitter.product\_mixer.core.feature.Feature

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

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

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

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

import com.twitter.product\_mixer.core.model.common.identifier.FeatureHydratorIdentifier

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

import com.twitter.stitch.Stitch

/\*\*

\* Hydrate features for a specific candidate

\* e.g. if the candidate is a Tweet then a feature could be whether it's is marked as sensitive

\*

\* @note if you want to conditionally run a [[BaseCandidateFeatureHydrator]] you can use the mixin [[com.twitter.product\_mixer.core.model.common.Conditionally]]

\* or to gate on a [[com.twitter.timelines.configapi.Param]] you can use

\* [[com.twitter.product\_mixer.component\_library.feature\_hydrator.candidate.param\_gated.ParamGatedCandidateFeatureHydrator]] or

\* [[com.twitter.product\_mixer.component\_library.feature\_hydrator.candidate.param\_gated.ParamGatedBulkCandidateFeatureHydrator]]

\*/

sealed trait BaseCandidateFeatureHydrator[

-Query <: PipelineQuery,

-Result <: UniversalNoun[Any],

FeatureType <: Feature[\_, \_]]

extends FeatureHydrator[FeatureType]

with SupportsConditionally[Query]

/\*\*

\* A candidate feature hydrator that provides an implementation for hydrating a single candidate

\* at the time. Product Mixer core takes care of hydrating all your candidates for you by

\* calling this for each candidate. This is useful for Stitch-powered downstream APIs (such

\* as Strato, Gizmoduck, etc) where the API takes a single candidate/key and Stitch handles

\* batching for you.

\*

\* @note Any exceptions that are thrown or returned as [[Stitch.exception]] will be added to the

\* [[FeatureMap]] for \*all\* [[Feature]]s intended to be hydrated.

\* Accessing a failed Feature will throw if using [[FeatureMap.get]] for Features that aren't

\* [[com.twitter.product\_mixer.core.feature.FeatureWithDefaultOnFailure]]

\*

\* @tparam Query The query type

\* @tparam Result The Candidate type

\*/

trait CandidateFeatureHydrator[-Query <: PipelineQuery, -Result <: UniversalNoun[Any]]

extends BaseCandidateFeatureHydrator[Query, Result, Feature[\_, \_]] {

override val identifier: FeatureHydratorIdentifier

/\*\* Hydrates a [[FeatureMap]] for a single candidate \*/

def apply(query: Query, candidate: Result, existingFeatures: FeatureMap): Stitch[FeatureMap]

}

/\*\*

\* Hydrate features for a list of candidates

\* e.g. for a list of Tweet candidates, a feature could be the visibility reason whether to show or not show each Tweet

\*/

trait BaseBulkCandidateFeatureHydrator[

-Query <: PipelineQuery,

-Result <: UniversalNoun[Any],

FeatureType <: Feature[\_, \_]]

extends BaseCandidateFeatureHydrator[Query, Result, FeatureType] {

/\*\*

\* Hydrates a set of [[FeatureMap]]s for the bulk list of candidates. Every input candidate must

\* have corresponding entry in the returned seq with a feature map.

\*/

def apply(

query: Query,

candidates: Seq[CandidateWithFeatures[Result]]

): Stitch[Seq[FeatureMap]]

}

/\*\*

\* A candidate feature hydrator that allows a user to bulk hydrate features for all candidates

\* at once. This is useful for downstream APIs that take a list of candidates in one go such

\* as feature store or scorers.

\*

\* @note Any exceptions that are thrown or returned as [[Stitch.exception]] will be added to the

\* [[FeatureMap]] for \*all\* [[Feature]]s of \*all\* candidates intended to be hydrated.

\* An alternative to throwing an exception is per-candidate failure handling (e.g. adding

\* a failed [[Feature]] with `addFailure`, a Try with `add`, or an optional value with `add`

\* using [[FeatureMapBuilder]]).

\* Accessing a failed Feature will throw if using [[FeatureMap.get]] for Features that aren't

\* [[com.twitter.product\_mixer.core.feature.FeatureWithDefaultOnFailure]].

\*

\* @tparam Query The query type

\* @tparam Result The Candidate type

\*/

trait BulkCandidateFeatureHydrator[-Query <: PipelineQuery, Candidate <: UniversalNoun[Any]]

extends BaseBulkCandidateFeatureHydrator[Query, Candidate, Feature[\_, \_]] {

override val identifier: FeatureHydratorIdentifier

override def apply(

query: Query,

candidates: Seq[CandidateWithFeatures[Candidate]]

): Stitch[Seq[FeatureMap]]

}