package com.twitter.product\_mixer.component\_library.side\_effect

import com.twitter.product\_mixer.core.functional\_component.side\_effect.PipelineResultSideEffect

import com.twitter.product\_mixer.core.model.common.presentation.CandidateWithDetails

import com.twitter.product\_mixer.core.model.marshalling.HasMarshalling

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

import com.twitter.stitch.Stitch

import com.twitter.strato.client.Inserter

/\*\*

\* Side effect that writes to Strato column's Insert Op. Create an implementation of this trait by

\* defining the `buildEvents` method and providing a Strato Column inserter of type

\* (StratoKeyarg, StratoValue) -> Any.

\* See https://docbird.twitter.biz/strato/ColumnCatalog.html#insert for information about

\* the Insert operation in Strato.

\*

\* @tparam StratoKeyarg Argument used as a key for Strato column. Could be Unit for common use-cases.

\* @tparam StratoValue Value that is inserted at the Strato column.

\* @tparam Query PipelineQuery

\* @tparam DomainResponseType Timeline response that is marshalled to domain model (e.g. URT, Slice etc).

\*/

trait StratoInsertSideEffect[

StratoKeyarg,

StratoValue,

Query <: PipelineQuery,

DomainResponseType <: HasMarshalling]

extends PipelineResultSideEffect[Query, DomainResponseType] {

/\*\*

\* Inserter for the InsertOp on a StratoColumn. In Strato, the InsertOp is represented as

\* (Keyarg, Value) => Key, where Key represents the result returned by the Insert operation.

\* For the side-effect behavior, we do not need the return value and use Any instead.

\*/

val stratoInserter: Inserter[StratoKeyarg, StratoValue, Any]

/\*\*

\* Builds the events that are inserted to the Strato column. This method supports generating

\* multiple events for a single side-effect invocation.

\*

\* @param query PipelineQuery

\* @param selectedCandidates Result after Selectors are executed

\* @param remainingCandidates Candidates which were not selected

\* @param droppedCandidates Candidates dropped during selection

\* @param response Timeline response that is marshalled to domain model (e.g. URT, Slice etc).

\* @return Tuples of (StratoKeyArg, StratoValue) that are used to call the stratoInserter.

\*/

def buildEvents(

query: Query,

selectedCandidates: Seq[CandidateWithDetails],

remainingCandidates: Seq[CandidateWithDetails],

droppedCandidates: Seq[CandidateWithDetails],

response: DomainResponseType

): Seq[(StratoKeyarg, StratoValue)]

final override def apply(

inputs: PipelineResultSideEffect.Inputs[Query, DomainResponseType]

): Stitch[Unit] = {

val events = buildEvents(

query = inputs.query,

selectedCandidates = inputs.selectedCandidates,

remainingCandidates = inputs.remainingCandidates,

droppedCandidates = inputs.droppedCandidates,

response = inputs.response

)

Stitch

.traverse(events) { case (keyarg, value) => stratoInserter.insert(keyarg, value) }

.unit

}

}