package com.twitter.product\_mixer.component\_library.decorator.urt.builder.timeline\_module

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.icon.BaseHorizonIconBuilder

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.metadata.BaseStr

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.social\_context.BaseModuleSocialContextBuilder

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.timeline\_module.BaseModuleHeaderBuilder

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.timeline\_module.BaseModuleHeaderDisplayTypeBuilder

import com.twitter.product\_mixer.core.model.marshalling.response.urt.timeline\_module.Classic

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

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

import com.twitter.product\_mixer.core.model.marshalling.response.urt.metadata.ImageVariant

import com.twitter.product\_mixer.core.model.marshalling.response.urt.timeline\_module.ModuleHeader

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

case class ModuleHeaderBuilder[-Query <: PipelineQuery, -Candidate <: UniversalNoun[Any]](

textBuilder: BaseStr[Query, Candidate],

isSticky: Option[Boolean] = None,

moduleHeaderIconBuilder: Option[BaseHorizonIconBuilder[Query, Candidate]] = None,

customIcon: Option[ImageVariant] = None,

moduleSocialContextBuilder: Option[BaseModuleSocialContextBuilder[Query, Candidate]] = None,

moduleHeaderDisplayTypeBuilder: BaseModuleHeaderDisplayTypeBuilder[Query, Candidate] =

ModuleHeaderDisplayTypeBuilder(Classic))

extends BaseModuleHeaderBuilder[Query, Candidate] {

override def apply(

query: Query,

candidates: Seq[CandidateWithFeatures[Candidate]]

): Option[ModuleHeader] = {

val firstCandidate = candidates.head

Some(

ModuleHeader(

text = textBuilder(query, firstCandidate.candidate, firstCandidate.features),

sticky = isSticky,

customIcon = customIcon,

socialContext = moduleSocialContextBuilder.flatMap(\_.apply(query, candidates)),

icon = moduleHeaderIconBuilder.flatMap(\_.apply(query, candidates)),

moduleHeaderDisplayType = moduleHeaderDisplayTypeBuilder(query, candidates),

)

)

}

}