package com.twitter.product\_mixer.component\_library.decorator.urt.builder.social\_context

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

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

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.GeneralContext

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

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

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

/\*\*

\* This class works the same as [[GeneralSocialContextBuilder]] but passes a list of candidates

\* into [[BaseModuleStr]] when rendering the string.

\*/

case class GeneralModuleSocialContextBuilder[

-Query <: PipelineQuery,

-Candidate <: UniversalNoun[Any]

](

textBuilder: BaseModuleStr[Query, Candidate],

contextType: GeneralContextType,

url: Option[String] = None,

contextImageUrls: Option[List[String]] = None,

landingUrl: Option[Url] = None)

extends BaseModuleSocialContextBuilder[Query, Candidate] {

def apply(

query: Query,

candidates: Seq[CandidateWithFeatures[Candidate]]

): Option[GeneralContext] =

Some(

GeneralContext(

text = textBuilder(query, candidates),

contextType = contextType,

url = url,

contextImageUrls = contextImageUrls,

landingUrl = landingUrl))

}