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

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

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.stringcenter.BaseModuleStringCenterPlaceholderBuilder

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

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

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

import com.twitter.stringcenter.client.StringCenter

import com.twitter.stringcenter.client.core.ExternalString

/\*\*

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

\* [[BaseModuleStringCenterPlaceholderBuilder]] when building the placeholders.

\*/

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

text: ExternalString,

stringCenter: StringCenter,

stringCenterPlaceholderBuilder: Option[

BaseModuleStringCenterPlaceholderBuilder[Query, Candidate]

] = None)

extends BaseModuleStr[Query, Candidate] {

def apply(query: Query, candidates: Seq[CandidateWithFeatures[Candidate]]): String = {

val placeholderMapOpt =

stringCenterPlaceholderBuilder.map(\_.apply(query, candidates))

stringCenter.prepare(

externalString = text,

placeholders = placeholderMapOpt.getOrElse(Map.empty[String, Any])

)

}

}