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

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.richtext.RichTextReferenceObjectBuilder

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.richtext.twitter\_text.TwitterTextEntityProcessor.DefaultReferenceObjectBuilder

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

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

import com.twitter.product\_mixer.core.model.marshalling.response.urt.richtext.ReferenceObject

import com.twitter.product\_mixer.core.model.marshalling.response.urt.richtext.RichTextCashtag

import com.twitter.product\_mixer.core.model.marshalling.response.urt.richtext.RichTextHashtag

import com.twitter.twittertext.Extractor

import scala.collection.convert.ImplicitConversions.\_

object TwitterTextEntityProcessor {

object DefaultReferenceObjectBuilder extends RichTextReferenceObjectBuilder {

def apply(twitterEntity: Extractor.Entity): Option[ReferenceObject] = {

twitterEntity.getType match {

case Extractor.Entity.Type.URL =>

Some(Url(ExternalUrl, twitterEntity.getValue))

case Extractor.Entity.Type.HASHTAG =>

Some(RichTextHashtag(twitterEntity.getValue))

case Extractor.Entity.Type.CASHTAG =>

Some(RichTextCashtag(twitterEntity.getValue))

case \_ => None

}

}

}

}

/\*\*

\* Add the corresponding [[RichTextEntity]] extraction logic into [[TwitterTextRenderer]].

\* The [[TwitterTextRenderer]] after being processed will extract the defined entities.

\*/

case class TwitterTextEntityProcessor(

twitterTextReferenceObjectBuilder: RichTextReferenceObjectBuilder = DefaultReferenceObjectBuilder)

extends TwitterTextRendererProcessor {

private[this] val extractor = new Extractor()

def process(

twitterTextRenderer: TwitterTextRenderer

): TwitterTextRenderer = {

val twitterEntities = extractor.extractEntitiesWithIndices(twitterTextRenderer.text)

twitterEntities.foreach { twitterEntity =>

twitterTextReferenceObjectBuilder(twitterEntity).foreach { refObject =>

twitterTextRenderer.setRefObject(twitterEntity.getStart, twitterEntity.getEnd, refObject)

}

}

twitterTextRenderer

}

}