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.RichTextRtlOptionBuilder

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

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

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMap

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

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.richtext.BaseRichTextBuilder

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

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

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

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

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

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

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

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

stringBuilder: BaseStr[Query, Candidate],

alignment: Option[RichTextAlignment] = None,

formats: Set[RichTextFormat] = Set(Plain, Strong),

twitterTextRtlOptionBuilder: RichTextRtlOptionBuilder[Query] =

StaticRichTextRtlOptionBuilder[Query](None),

twitterTextReferenceObjectBuilder: RichTextReferenceObjectBuilder = DefaultReferenceObjectBuilder)

extends BaseRichTextBuilder[Query, Candidate] {

def apply(query: Query, candidate: Candidate, candidateFeatures: FeatureMap): RichText = {

val twitterTextRenderer = TwitterTextRenderer(

text = stringBuilder(query, candidate, candidateFeatures),

rtl = twitterTextRtlOptionBuilder(query),

alignment = alignment)

twitterTextRenderer

.transform(TwitterTextFormatProcessor(formats))

.transform(TwitterTextEntityProcessor(twitterTextReferenceObjectBuilder))

.build

}

}