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

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.alert.ShowAlertCandidateUrtItemBuilder.ShowAlertClientEventInfoElement

import com.twitter.product\_mixer.component\_library.model.candidate.ShowAlertCandidate

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

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

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.item.alert.BaseDurationBuilder

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.item.alert.BaseShowAlertColorConfigurationBuilder

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.item.alert.BaseShowAlertDisplayLocationBuilder

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.item.alert.BaseShowAlertIconDisplayInfoBuilder

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.item.alert.BaseShowAlertNavigationMetadataBuilder

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.item.alert.BaseShowAlertUserIdsBuilder

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

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

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

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

import com.twitter.product\_mixer.core.model.marshalling.response.urt.alert.ShowAlertType

object ShowAlertCandidateUrtItemBuilder {

val ShowAlertClientEventInfoElement: String = "showAlert"

}

case class ShowAlertCandidateUrtItemBuilder[-Query <: PipelineQuery](

alertType: ShowAlertType,

displayLocationBuilder: BaseShowAlertDisplayLocationBuilder[Query],

colorConfigBuilder: BaseShowAlertColorConfigurationBuilder[Query],

triggerDelayBuilder: Option[BaseDurationBuilder[Query]] = None,

displayDurationBuilder: Option[BaseDurationBuilder[Query]] = None,

clientEventInfoBuilder: Option[BaseClientEventInfoBuilder[Query, ShowAlertCandidate]] = None,

collapseDelayBuilder: Option[BaseDurationBuilder[Query]] = None,

userIdsBuilder: Option[BaseShowAlertUserIdsBuilder[Query]] = None,

richTextBuilder: Option[BaseRichTextBuilder[Query, ShowAlertCandidate]] = None,

iconDisplayInfoBuilder: Option[BaseShowAlertIconDisplayInfoBuilder[Query]] = None,

navigationMetadataBuilder: Option[BaseShowAlertNavigationMetadataBuilder[Query]] = None)

extends CandidateUrtEntryBuilder[

Query,

ShowAlertCandidate,

ShowAlert

] {

override def apply(

query: Query,

candidate: ShowAlertCandidate,

features: FeatureMap,

): ShowAlert = ShowAlert(

id = candidate.id,

sortIndex = None,

alertType = alertType,

triggerDelay = triggerDelayBuilder.flatMap(\_.apply(query, candidate, features)),

displayDuration = displayDurationBuilder.flatMap(\_.apply(query, candidate, features)),

clientEventInfo = clientEventInfoBuilder.flatMap(

\_.apply(query, candidate, features, Some(ShowAlertClientEventInfoElement))),

collapseDelay = collapseDelayBuilder.flatMap(\_.apply(query, candidate, features)),

userIds = userIdsBuilder.flatMap(\_.apply(query, candidate, features)),

richText = richTextBuilder.map(\_.apply(query, candidate, features)),

iconDisplayInfo = iconDisplayInfoBuilder.flatMap(\_.apply(query, candidate, features)),

displayLocation = displayLocationBuilder(query, candidate, features),

colorConfig = colorConfigBuilder(query, candidate, features),

navigationMetadata = navigationMetadataBuilder.flatMap(\_.apply(query, candidate, features)),

)

}