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

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

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

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

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.timeline\_module.BaseModuleDisplayTypeBuilder

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.metadata.BaseFeedbackActionInfoBuilder

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.timeline\_module.BaseModuleFooterBuilder

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.timeline\_module.BaseModuleHeaderBuilder

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.timeline\_module.BaseModuleMetadataBuilder

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.timeline\_module.BaseModuleShowMoreBehaviorBuilder

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.timeline\_module.BaseTimelineModuleBuilder

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

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

entryNamespace: EntryNamespace,

displayTypeBuilder: BaseModuleDisplayTypeBuilder[Query, Candidate],

clientEventInfoBuilder: BaseClientEventInfoBuilder[Query, Candidate],

moduleIdGeneration: ModuleIdGeneration = AutomaticUniqueModuleId(),

feedbackActionInfoBuilder: Option[

BaseFeedbackActionInfoBuilder[Query, Candidate]

] = None,

headerBuilder: Option[BaseModuleHeaderBuilder[Query, Candidate]] = None,

footerBuilder: Option[BaseModuleFooterBuilder[Query, Candidate]] = None,

metadataBuilder: Option[BaseModuleMetadataBuilder[Query, Candidate]] = None,

showMoreBehaviorBuilder: Option[BaseModuleShowMoreBehaviorBuilder[Query, Candidate]] = None)

extends BaseTimelineModuleBuilder[Query, Candidate] {

override def apply(

query: Query,

candidates: Seq[CandidateWithFeatures[Candidate]]

): TimelineModule = {

val firstCandidate = candidates.head

TimelineModule(

id = moduleIdGeneration.moduleId,

// Sort indexes are automatically set in the domain marshaller phase

sortIndex = None,

entryNamespace = entryNamespace,

// Modules should not need an element by default; only items should

clientEventInfo =

clientEventInfoBuilder(query, firstCandidate.candidate, firstCandidate.features, None),

feedbackActionInfo = feedbackActionInfoBuilder.flatMap(

\_.apply(query, firstCandidate.candidate, firstCandidate.features)),

isPinned = None,

// Items are automatically set in the domain marshaller phase

items = Seq.empty,

displayType = displayTypeBuilder(query, candidates),

header = headerBuilder.flatMap(\_.apply(query, candidates)),

footer = footerBuilder.flatMap(\_.apply(query, candidates)),

metadata = metadataBuilder.map(\_.apply(query, candidates)),

showMoreBehavior = showMoreBehaviorBuilder.map(\_.apply(query, candidates))

)

}

}