package com.twitter.product\_mixer.component\_library.pipeline.candidate.who\_to\_follow\_module

import com.twitter.product\_mixer.component\_library.candidate\_source.people\_discovery.PeopleDiscoveryCandidateSource

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

import com.twitter.product\_mixer.core.feature.Feature

import com.twitter.product\_mixer.core.functional\_component.common.alert.Alert

import com.twitter.product\_mixer.core.functional\_component.configapi.StaticParam

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.BaseModuleDisplayTypeBuilder

import com.twitter.product\_mixer.core.functional\_component.filter.Filter

import com.twitter.product\_mixer.core.functional\_component.gate.Gate

import com.twitter.product\_mixer.core.model.common.identifier.CandidatePipelineIdentifier

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

import com.twitter.timelines.configapi.FSParam

import com.twitter.timelines.configapi.Param

import com.twitter.timelines.configapi.decider.DeciderParam

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class WhoToFollowCandidatePipelineConfigBuilder @Inject() (

whoToFollowCandidateSource: PeopleDiscoveryCandidateSource) {

/\*\*

\* Build a WhoToFollowCandidatePipelineConfig

\*

\* To create a DependentCandidatePipelineConfig instead see [[WhoToFollowDependentCandidatePipelineConfigBuilder]].

\*

\* @note If injected classes are needed to populate parameters in this method, consider creating a

\* ProductWhoToFollowCandidatePipelineConfigBuilder with a single `def build()` method. That

\* product-specific builder class can then inject everything it needs (including this class),

\* and delegate to this class's build() method within its own build() method.

\*/

def build[Query <: PipelineQuery](

moduleDisplayTypeBuilder: BaseModuleDisplayTypeBuilder[Query, UserCandidate],

identifier: CandidatePipelineIdentifier = WhoToFollowCandidatePipelineConfig.identifier,

enabledDeciderParam: Option[DeciderParam[Boolean]] = None,

supportedClientParam: Option[FSParam[Boolean]] = None,

alerts: Seq[Alert] = Seq.empty,

gates: Seq[Gate[Query]] = Seq.empty,

filters: Seq[Filter[Query, UserCandidate]] = Seq.empty,

feedbackActionInfoBuilder: Option[BaseFeedbackActionInfoBuilder[

PipelineQuery,

UserCandidate

]] = None,

displayLocationParam: Param[String] =

StaticParam(WhoToFollowCandidatePipelineQueryTransformer.DisplayLocation),

supportedLayoutsParam: Param[Seq[String]] =

StaticParam(WhoToFollowCandidatePipelineQueryTransformer.SupportedLayouts),

layoutVersionParam: Param[Int] =

StaticParam(WhoToFollowCandidatePipelineQueryTransformer.LayoutVersion),

excludedUserIdsFeature: Option[Feature[PipelineQuery, Seq[Long]]] = None,

): WhoToFollowCandidatePipelineConfig[Query] =

new WhoToFollowCandidatePipelineConfig(

identifier = identifier,

enabledDeciderParam = enabledDeciderParam,

supportedClientParam = supportedClientParam,

alerts = alerts,

gates = gates,

moduleDisplayTypeBuilder = moduleDisplayTypeBuilder,

whoToFollowCandidateSource = whoToFollowCandidateSource,

filters = filters,

feedbackActionInfoBuilder = feedbackActionInfoBuilder,

displayLocationParam = displayLocationParam,

supportedLayoutsParam = supportedLayoutsParam,

layoutVersionParam = layoutVersionParam,

excludedUserIdsFeature = excludedUserIdsFeature

)

}