package com.twitter.home\_mixer.product.for\_you.param

import com.twitter.home\_mixer.param.decider.DeciderKey

import com.twitter.home\_mixer.product.for\_you.param.ForYouParam.\_

import com.twitter.product\_mixer.core.product.ProductParamConfig

import com.twitter.servo.decider.DeciderKeyName

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class ForYouParamConfig @Inject() () extends ProductParamConfig {

override val enabledDeciderKey: DeciderKeyName = DeciderKey.EnableForYouProduct

override val supportedClientFSName: String = SupportedClientFSName

override val booleanDeciderOverrides = Seq(

EnableScoredTweetsCandidatePipelineParam

)

override val booleanFSOverrides = Seq(

ClearCacheOnPtr.EnableParam,

EnableFlipInjectionModuleCandidatePipelineParam,

EnablePushToHomeMixerPipelineParam,

EnableScoredTweetsMixerPipelineParam,

EnableServedCandidateKafkaPublishingParam,

EnableTimelineScorerCandidatePipelineParam,

EnableTopicSocialContextFilterParam,

EnableVerifiedAuthorSocialContextBypassParam,

EnableWhoToFollowCandidatePipelineParam,

EnableWhoToSubscribeCandidatePipelineParam,

EnableTweetPreviewsCandidatePipelineParam,

EnableClearCacheOnPushToHome

)

override val boundedIntFSOverrides = Seq(

AdsNumOrganicItemsParam,

ClearCacheOnPtr.MinEntriesParam,

FlipInlineInjectionModulePosition,

ServerMaxResultsParam,

WhoToFollowPositionParam,

WhoToSubscribePositionParam,

TweetPreviewsPositionParam,

TweetPreviewsMaxCandidatesParam

)

override val stringFSOverrides = Seq(

WhoToFollowDisplayLocationParam,

ExperimentStatsParam

)

override val boundedDurationFSOverrides = Seq(

WhoToFollowMinInjectionIntervalParam,

WhoToSubscribeMinInjectionIntervalParam,

TweetPreviewsMinInjectionIntervalParam

)

override val enumFSOverrides = Seq(

WhoToFollowDisplayTypeIdParam,

WhoToSubscribeDisplayTypeIdParam

)

}