package com.twitter.frigate.pushservice.module

import com.google.inject.Provides

import com.google.inject.Singleton

import com.twitter.decider.Decider

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.pushservice.config.DeployConfig

import com.twitter.frigate.pushservice.target.PushTargetUserBuilder

import com.twitter.inject.TwitterModule

object PushTargetUserBuilderModule extends TwitterModule {

@Provides

@Singleton

def providesPushTargetUserBuilder(

decider: Decider,

config: DeployConfig,

statsReceiver: StatsReceiver

): PushTargetUserBuilder = {

PushTargetUserBuilder(

historyStore = config.historyStore,

emailHistoryStore = config.emailHistoryStore,

labeledPushRecsStore = config.labeledPushRecsDecideredStore,

onlineUserHistoryStore = config.onlineUserHistoryStore,

pushRecItemsStore = config.pushRecItemStore,

userStore = config.safeUserStore,

pushInfoStore = config.pushInfoStore,

userCountryStore = config.userCountryStore,

userUtcOffsetStore = config.userUtcOffsetStore,

dauProbabilityStore = config.dauProbabilityStore,

nsfwConsumerStore = config.nsfwConsumerStore,

genericNotificationFeedbackStore = config.genericNotificationFeedbackStore,

userFeatureStore = config.userFeaturesStore,

mrUserStateStore = config.mrUserStatePredictionStore,

tweetImpressionStore = config.tweetImpressionStore,

timelinesUserSessionStore = config.timelinesUserSessionStore,

cachedTweetyPieStore = config.cachedTweetyPieStoreV2,

strongTiesStore = config.strongTiesStore,

userHTLLastVisitStore = config.userHTLLastVisitStore,

userLanguagesStore = config.userLanguagesStore,

inputDecider = decider,

inputAbDecider = config.abDecider,

realGraphScoresTop500InStore = config.realGraphScoresTop500InStore,

recentFollowsStore = config.recentFollowsStore,

resurrectedUserStore = config.reactivatedUserInfoStore,

configParamsBuilder = config.configParamsBuilder,

optOutUserInterestsStore = config.optOutUserInterestsStore,

deviceInfoStore = config.deviceInfoStore,

pushcapDynamicPredictionStore = config.pushcapDynamicPredictionStore,

appPermissionStore = config.appPermissionStore,

optoutModelScorer = config.optoutModelScorer,

userTargetingPropertyStore = config.userTargetingPropertyStore,

ntabCaretFeedbackStore = config.ntabCaretFeedbackStore,

genericFeedbackStore = config.genericFeedbackStore,

inlineActionHistoryStore = config.inlineActionHistoryStore,

featureHydrator = config.featureHydrator,

openAppUserStore = config.openAppUserStore,

openedPushByHourAggregatedStore = config.openedPushByHourAggregatedStore,

geoduckStoreV2 = config.geoDuckV2Store,

superFollowEligibilityUserStore = config.superFollowEligibilityUserStore,

superFollowApplicationStatusStore = config.superFollowApplicationStatusStore

)(statsReceiver)

}

}