package com.twitter.frigate.pushservice.take

import com.twitter.frigate.pushservice.model.PushTypes.PushCandidate

import com.twitter.frigate.thriftscala.ChannelName

import com.twitter.util.Future

import java.util.concurrent.ConcurrentHashMap

import scala.collection.concurrent

import scala.collection.convert.decorateAsScala.\_

/\*\*

\* A class to save all the channel related information

\*/

trait ChannelForCandidate {

self: PushCandidate =>

// Cache of channel selection result

private[this] val selectedChannels: concurrent.Map[String, Future[Seq[ChannelName]]] =

new ConcurrentHashMap[String, Future[Seq[ChannelName]]]().asScala

// Returns the channel information from all ChannelSelectors.

def getChannels(): Future[Seq[ChannelName]] = {

Future.collect(selectedChannels.values.toSeq).map { c => c.flatten }

}

}