package com.twitter.follow\_recommendations.common.models

import com.twitter.util.Time

trait HasWtfImpressions {

def wtfImpressions: Option[Seq[WtfImpression]]

lazy val numWtfImpressions: Int = wtfImpressions.map(\_.size).getOrElse(0)

lazy val candidateImpressions: Map[Long, WtfImpression] = wtfImpressions

.map { imprMap =>

imprMap.map { i =>

i.candidateId -> i

}.toMap

}.getOrElse(Map.empty)

lazy val latestImpressionTime: Time = {

if (wtfImpressions.exists(\_.nonEmpty)) {

wtfImpressions.get.map(\_.latestTime).max

} else Time.Top

}

def getCandidateImpressionCounts(id: Long): Option[Int] =

candidateImpressions.get(id).map(\_.counts)

def getCandidateLatestTime(id: Long): Option[Time] = {

candidateImpressions.get(id).map(\_.latestTime)

}

}