package com.twitter.graph\_feature\_service.scalding

import com.twitter.scalding.DateRange

import com.twitter.scalding.Execution

import com.twitter.scalding.RichDate

import com.twitter.scalding.UniqueID

import java.util.Calendar

import java.util.TimeZone

import sun.util.calendar.BaseCalendar

/\*\*

\* To launch an adhoc run:

\*

scalding remote run --target graph-feature-service/src/main/scalding/com/twitter/graph\_feature\_service/scalding:graph\_feature\_service\_adhoc\_job

\*/

object GraphFeatureServiceAdhocApp

extends GraphFeatureServiceMainJob

with GraphFeatureServiceAdhocBaseApp {}

/\*\*

\* To schedule the job, upload the workflows config (only required for the first time and subsequent config changes):

\* scalding workflow upload --jobs graph-feature-service/src/main/scalding/com/twitter/graph\_feature\_service/scalding:graph\_feature\_service\_daily\_job --autoplay --build-cron-schedule "20 23 1 \* \*"

\* You can then build from the UI by clicking "Build" and pasting in your remote branch, or leave it empty if you're redeploying from master.

\* The workflows config above should automatically trigger once each month.

\*/

object GraphFeatureServiceScheduledApp

extends GraphFeatureServiceMainJob

with GraphFeatureServiceScheduledBaseApp {

override def firstTime: RichDate = RichDate("2018-05-18")

override def runOnDateRange(

enableValueGraphs: Option[Boolean],

enableKeyGraphs: Option[Boolean]

)(

implicit dateRange: DateRange,

timeZone: TimeZone,

uniqueID: UniqueID

): Execution[Unit] = {

// Only run the value Graphs on Tuesday, Thursday, Saturday

val overrideEnableValueGraphs = {

val dayOfWeek = dateRange.start.toCalendar.get(Calendar.DAY\_OF\_WEEK)

dayOfWeek == BaseCalendar.TUESDAY |

dayOfWeek == BaseCalendar.THURSDAY |

dayOfWeek == BaseCalendar.SATURDAY

}

super.runOnDateRange(

Some(true),

Some(false) // disable key Graphs since we are not using them in production

)

}

}