package com.twitter.product\_mixer.component\_library.experiments.metrics

import scala.collection.immutable.ListSet

/\*\*

\*

\* @param id optional metric group id. If id is None, this means the group

\* is being newly created and the id is not provisioned by go/ddg. Otherwise, the metric

\* group is present in DDG and has a corresponding id.

\* @param name metric group name

\* @param description metric group description

\* @param metrics set of metrics that belong to this metric group

\*/

case class MetricGroup(

id: Option[Long],

name: String,

description: String,

metrics: ListSet[Metric]) {

/\*

\* Returns a CSV representation of this metric group that can be imported via DDG's bulk import tool

\* The bulk import tool consumes CSV data with the following columns:

\* 1. group name

\* 2. group description

\* 3. metric name

\* 4. metric description

\* 5. metric pattern

\* 6. group id -- numeric id

\* 7. (optional) metric type -- `NAMED\_PATTERN`, `STRAINER`, or `LAMBDA`.

\*/

def toCsv: String = {

val metricCsvLines: ListSet[String] = for {

metric <- metrics

definition <- metric.definition.toCsvField

} yield {

Seq(

name,

description,

metric.name,

metric.name,

// wrap in single quotes so that DDG bulk import tool correctly parses

s""""$definition"""",

id.map(\_.toString).getOrElse(""),

metric.definition.metricDefinitionType

).mkString(",")

}

println(s"Generated metrics in CSV count: ${metricCsvLines.size}")

metricCsvLines.mkString("\n")

}

// Unique metric names based on globally unique metric name

def uniqueMetricNames: Set[String] =

metrics.groupBy(\_.name).keys.toSet

}