package com.twitter.cr\_mixer.scribe

/\*\*

\* Categories define scribe categories used in cr-mixer service.

\*/

object ScribeCategories {

lazy val AllCategories =

List(AbDecider, TopLevelApiDdgMetrics, TweetsRecs)

/\*\*

\* AbDecider represents scribe logs for experiments

\*/

lazy val AbDecider: ScribeCategory = ScribeCategory(

"abdecider\_scribe",

"client\_event"

)

/\*\*

\* Top-Level Client event scribe logs, to record changes in system metrics (e.g. latency,

\* candidates returned, empty rate ) per experiment bucket, and store them in DDG metric group

\*/

lazy val TopLevelApiDdgMetrics: ScribeCategory = ScribeCategory(

"top\_level\_api\_ddg\_metrics\_scribe",

"client\_event"

)

lazy val TweetsRecs: ScribeCategory = ScribeCategory(

"get\_tweets\_recommendations\_scribe",

"cr\_mixer\_get\_tweets\_recommendations"

)

lazy val VITTweetsRecs: ScribeCategory = ScribeCategory(

"get\_vit\_tweets\_recommendations\_scribe",

"cr\_mixer\_get\_vit\_tweets\_recommendations"

)

lazy val RelatedTweets: ScribeCategory = ScribeCategory(

"get\_related\_tweets\_scribe",

"cr\_mixer\_get\_related\_tweets"

)

lazy val UtegTweets: ScribeCategory = ScribeCategory(

"get\_uteg\_tweets\_scribe",

"cr\_mixer\_get\_uteg\_tweets"

)

lazy val AdsRecommendations: ScribeCategory = ScribeCategory(

"get\_ads\_recommendations\_scribe",

"cr\_mixer\_get\_ads\_recommendations"

)

}

/\*\*

\* Category represents each scribe log data.

\*

\* @param loggerFactoryNode loggerFactory node name in cr-mixer associated with this scribe category

\* @param scribeCategory scribe category name (globally unique at Twitter)

\*/

case class ScribeCategory(

loggerFactoryNode: String,

scribeCategory: String) {

def getProdLoggerFactoryNode: String = loggerFactoryNode

def getStagingLoggerFactoryNode: String = "staging\_" + loggerFactoryNode

}