package com.twitter.product\_mixer.core.controllers

import com.fasterxml.jackson.annotation.JsonIgnoreProperties

import com.twitter.product\_mixer.core.functional\_component.common.alert.Alert

import com.twitter.product\_mixer.core.functional\_component.common.alert.NotificationGroup

import com.twitter.product\_mixer.core.functional\_component.common.alert.Source

/\*\*

\* Simple representation for an [[Alert]] used for Product Mixer's JSON API, which in turn is

\* consumed by our monitoring script generation job and Turntable.

\*

\* @note not all mixers will upgrade at the same time so new fields should be added with backwards

\* compatibility in mind.

\*/

@JsonIgnoreProperties(ignoreUnknown = true)

private[core] case class AlertConfig(

source: Source,

metricType: String,

notificationGroup: NotificationGroup,

warnPredicate: PredicateConfig,

criticalPredicate: PredicateConfig,

runbookLink: Option[String],

metricSuffix: Option[String])

private[core] object AlertConfig {

/\*\* Represent this [[Alert]] as an [[AlertConfig]] case class \*/

private[core] def apply(alert: Alert): AlertConfig =

AlertConfig(

alert.source,

alert.alertType.metricType,

alert.notificationGroup,

PredicateConfig(alert.warnPredicate),

PredicateConfig(alert.criticalPredicate),

alert.runbookLink,

alert.metricSuffix

)

}