package com.twitter.product\_mixer.core.functional\_component.common.alert

import com.twitter.util.Try

import javax.mail.internet.InternetAddress

/\*\*

\* Destination represents the place to which alerts will be sent. Often you will only need one field

\* populated (either a Pager Duty key or a list of emails).

\*

\* See the Monitoring 2.0 docs for more information on [[https://docbird.twitter.biz/mon/how-to-guides.html?highlight=notificationgroup#set-up-email-pagerduty-and-slack-notifications setting up a Pager Duty rotation]]

\*/

case class Destination(

pagerDutyKey: Option[String] = None,

emails: Seq[String] = Seq.empty) {

require(

pagerDutyKey.forall(\_.length == 32),

s"Expected `pagerDutyKey` to be 32 characters long but got `$pagerDutyKey`")

emails.foreach { email =>

require(

Try(new InternetAddress(email).validate()).isReturn,

s"Expected only valid email addresses but got an invalid email address: `$email`")

}

require(

pagerDutyKey.nonEmpty || emails.nonEmpty,

s"Expected a `pagerDutyKey` or at least 1 email address but got neither")

}

/\*\*

\* NotificationGroup maps alert levels to destinations. Having different destinations based on the

\* urgency of the alert can sometimes be useful. For example, you could have a daytime on-call for

\* warn alerts and a 24 on-call for critical alerts.

\*/

case class NotificationGroup(

critical: Destination,

warn: Destination)