package com.twitter.product\_mixer.core.model.marshalling.response.urt

import scala.util.matching.Regex

/\*\*

\* Entry Identifiers (commonly entry ids) are a type of identifier used in URT to identify

\* unique timeline entries - tweets, users, modules, etc.

\*

\* Entry Identifiers are formed from two parts - a namespace (EntryNamespace) and an underlying

\* id.

\*

\* A Entry Namespace is restricted to:

\* - 3 to 60 characters to ensure reasonable length

\* - a-z and dashes (kebab-case)

\* - Examples include "user" and "tweet"

\*

\* When specific entries identifiers are created, they will be appended with a dash and their

\* own id, like user-12 or tweet-20

\*/

trait HasEntryNamespace {

val entryNamespace: EntryNamespace

}

// sealed abstract case class is basically a scala 2.12 opaque type -

// you can only create them via the factory method on the companion

// allowing us to enforce validation

sealed abstract case class EntryNamespace(override val toString: String)

object EntryNamespace {

val AllowedCharacters: Regex = "[a-z-]+".r // Allows for kebab-case

def apply(str: String): EntryNamespace = {

val isValid = str match {

case n if n.length < 3 =>

false

case n if n.length > 60 =>

false

case AllowedCharacters() =>

true

case \_ =>

false

}

if (isValid)

new EntryNamespace(str) {}

else

throw new IllegalArgumentException(s"Illegal EntryNamespace: $str")

}

}