package com.twitter.product\_mixer.component\_library.decorator.urt.builder.timeline\_module

/\*\*

\* This trait is used for Module ID generation. Clients are safe to ignore this code unless they

\* have a specific use case that requires hard-coded, specific, module ids. In that scenario,

\* they can use the [[ManualModuleId]] case class.

\*/

sealed trait ModuleIdGeneration {

val moduleId: Long

}

object ModuleIdGeneration {

def apply(moduleId: Long): ModuleIdGeneration = moduleId match {

case moduleId if AutomaticUniqueModuleId.isAutomaticUniqueModuleId(moduleId) =>

AutomaticUniqueModuleId(moduleId)

case moduleId => ManualModuleId(moduleId)

}

}

/\*\*

\* Generate unique Ids for each module, which results in unique URT entryIds

\* for each module even if they share the same entryNamespace.

\* This is the default and recommended use case.

\* Note that the module Id value is just a placeholder

\*/

case class AutomaticUniqueModuleId private (moduleId: Long = 0L) extends ModuleIdGeneration {

def withOffset(offset: Long): AutomaticUniqueModuleId = copy(

AutomaticUniqueModuleId.idRange.min + offset)

}

object AutomaticUniqueModuleId {

// We use a specific numeric range to track whether IDs should be automatically generated.

val idRange: Range = Range(-10000, -1000)

def apply(): AutomaticUniqueModuleId = AutomaticUniqueModuleId(idRange.min)

def isAutomaticUniqueModuleId(moduleId: Long): Boolean = idRange.contains(moduleId)

}

/\*\*

\* ManualModuleId should normally not be required, but is helpful if the

\* entryId of the module must be controlled. A scenario where this may be

\* required is if a single candidate source returns multiple modules, and

\* each module has the same presentation (e.g. Header, Footer). By setting

\* different IDs, we signal to the platform that each module should be separate

\* by using a different manual Id.

\*/

case class ManualModuleId(override val moduleId: Long) extends ModuleIdGeneration {

// Negative module IDs are reserved for internal usage

if (moduleId < 0) throw new IllegalArgumentException("moduleId must be a positive number")

}