package com.twitter.product\_mixer.core.functional\_component.common.access\_policy

import com.fasterxml.jackson.annotation.JsonSubTypes

import com.fasterxml.jackson.annotation.JsonTypeInfo

/\*\*

\* The Access Policy to set for gating querying in the turntable tool.

\*

\* @note implementations must be simple case classes with unique structures for serialization

\*/

@JsonTypeInfo(use = JsonTypeInfo.Id.NAME, include = JsonTypeInfo.As.PROPERTY)

@JsonSubTypes(

Array(

new JsonSubTypes.Type(value = classOf[AllowedLdapGroups], name = "AllowedLdapGroups"),

new JsonSubTypes.Type(value = classOf[BlockEverything], name = "BlockEverything")

)

)

sealed trait AccessPolicy

/\*\*

\* Users must be in \*at least\* one of the provided LDAP groups in order to make a query.

\*

\* @param groups LDAP groups allowed to access this product

\*/

case class AllowedLdapGroups(groups: Set[String]) extends AccessPolicy

object AllowedLdapGroups {

def apply(group: String): AllowedLdapGroups = AllowedLdapGroups(Set(group))

}

/\*\*

\* Block all requests to a product.

\*

\* @note this needs to be a case class rather than an object because classOf doesn't work on objects

\* and JsonSubTypes requires the annotation argument to be a constant (ruling out .getClass).

\* This issue may be resolved in Scala 2.13: https://github.com/scala/scala/pull/9279

\*/

case class BlockEverything() extends AccessPolicy