package com.twitter.follow\_recommendations.products

import com.twitter.follow\_recommendations.common.models.DisplayLocation

import com.twitter.follow\_recommendations.products.common.ProductRegistry

import com.twitter.follow\_recommendations.products.explore\_tab.ExploreTabProduct

import com.twitter.follow\_recommendations.products.home\_timeline.HomeTimelineProduct

import com.twitter.follow\_recommendations.products.home\_timeline\_tweet\_recs.HomeTimelineTweetRecsProduct

import com.twitter.follow\_recommendations.products.sidebar.SidebarProduct

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class ProdProductRegistry @Inject() (

exploreTabProduct: ExploreTabProduct,

homeTimelineProduct: HomeTimelineProduct,

homeTimelineTweetRecsProduct: HomeTimelineTweetRecsProduct,

sidebarProduct: SidebarProduct,

) extends ProductRegistry {

override val products: Seq[common.Product] =

Seq(

exploreTabProduct,

homeTimelineProduct,

homeTimelineTweetRecsProduct,

sidebarProduct

)

override val displayLocationProductMap: Map[DisplayLocation, common.Product] =

products.groupBy(\_.displayLocation).flatMap {

case (loc, products) =>

assert(products.size == 1, s"Found more than 1 Product for ${loc}")

products.headOption.map { product => loc -> product }

}

override def getProductByDisplayLocation(displayLocation: DisplayLocation): common.Product = {

displayLocationProductMap.getOrElse(

displayLocation,

throw new MissingProductException(displayLocation))

}

}

class MissingProductException(displayLocation: DisplayLocation)

extends Exception(s"No Product found for ${displayLocation}")