package com.twitter.product\_mixer.core.pipeline.product

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

import com.twitter.product\_mixer.core.model.common.identifier.ProductPipelineIdentifier

import com.twitter.product\_mixer.core.model.marshalling.request.Request

import com.twitter.product\_mixer.core.pipeline.Pipeline

import com.twitter.stitch.Arrow

/\*\*

\* A Product Pipeline

\*

\* This is an abstract class, as we only construct these via the [[ProductPipelineBuilder]].

\*

\* A [[ProductPipeline]] is capable of processing a [[Request]] and returning a response.

\*

\* @tparam RequestType the domain model for the query or request

\* @tparam ResponseType the final marshalled result type

\*/

abstract class ProductPipeline[RequestType <: Request, ResponseType] private[product]

extends Pipeline[ProductPipelineRequest[RequestType], ResponseType]

with WithDebugAccessPolicies {

override private[core] val config: ProductPipelineConfig[RequestType, \_, ResponseType]

override val arrow: Arrow[

ProductPipelineRequest[RequestType],

ProductPipelineResult[ResponseType]

]

override val identifier: ProductPipelineIdentifier

}