package com.twitter.home\_mixer.module

import com.twitter.finatra.thrift.exceptions.ExceptionMapper

import com.twitter.home\_mixer.{thriftscala => t}

import com.twitter.util.logging.Logging

import com.twitter.product\_mixer.core.pipeline.pipeline\_failure.PipelineFailure

import com.twitter.product\_mixer.core.pipeline.pipeline\_failure.ProductDisabled

import com.twitter.scrooge.ThriftException

import com.twitter.util.Future

import javax.inject.Singleton

@Singleton

class PipelineFailureExceptionMapper

extends ExceptionMapper[PipelineFailure, ThriftException]

with Logging {

def handleException(throwable: PipelineFailure): Future[ThriftException] = {

throwable match {

// SliceService (unlike UrtService) throws an exception when the requested product is disabled

case PipelineFailure(ProductDisabled, reason, \_, \_) =>

Future.exception(

t.ValidationExceptionList(errors =

Seq(t.ValidationException(t.ValidationErrorCode.ProductDisabled, reason))))

case \_ =>

error("Unhandled PipelineFailure", throwable)

Future.exception(throwable)

}

}

}