package com.twitter.unified\_user\_actions.service

import com.twitter.decider.MockDecider

import com.twitter.inject.Test

import com.twitter.unified\_user\_actions.service.module.ClientEventDeciderUtils

import com.twitter.unified\_user\_actions.service.module.DefaultDeciderUtils

import com.twitter.unified\_user\_actions.thriftscala.\_

import com.twitter.util.Time

import com.twitter.util.mock.Mockito

import org.junit.runner.RunWith

import org.scalatestplus.junit.JUnitRunner

@RunWith(classOf[JUnitRunner])

class DeciderUtilsTest extends Test with Mockito {

trait Fixture {

val frozenTime = Time.fromMilliseconds(1658949273000L)

val publishActionTypes =

Set[ActionType](ActionType.ServerTweetFav, ActionType.ClientTweetRenderImpression)

def decider(

features: Set[String] = publishActionTypes.map { action =>

s"Publish${action.name}"

}

) = new MockDecider(features = features)

def mkUUA(actionType: ActionType) = UnifiedUserAction(

userIdentifier = UserIdentifier(userId = Some(91L)),

item = Item.TweetInfo(

TweetInfo(

actionTweetId = 1L,

actionTweetAuthorInfo = Some(AuthorInfo(authorId = Some(101L))),

)

),

actionType = actionType,

eventMetadata = EventMetadata(

sourceTimestampMs = 1001L,

receivedTimestampMs = frozenTime.inMilliseconds,

sourceLineage = SourceLineage.ServerTlsFavs,

traceId = Some(31L)

)

)

val uuaServerTweetFav = mkUUA(ActionType.ServerTweetFav)

val uuaClientTweetFav = mkUUA(ActionType.ClientTweetFav)

val uuaClientTweetRenderImpression = mkUUA(ActionType.ClientTweetRenderImpression)

}

test("Decider Utils") {

new Fixture {

Time.withTimeAt(frozenTime) { \_ =>

DefaultDeciderUtils.shouldPublish(

decider = decider(),

uua = uuaServerTweetFav,

sinkTopic = "") shouldBe true

DefaultDeciderUtils.shouldPublish(

decider = decider(),

uua = uuaClientTweetFav,

sinkTopic = "") shouldBe false

ClientEventDeciderUtils.shouldPublish(

decider = decider(),

uua = uuaClientTweetRenderImpression,

sinkTopic = "unified\_user\_actions\_engagements") shouldBe false

ClientEventDeciderUtils.shouldPublish(

decider = decider(),

uua = uuaClientTweetFav,

sinkTopic = "unified\_user\_actions\_engagements") shouldBe false

ClientEventDeciderUtils.shouldPublish(

decider = decider(features = Set[String](s"Publish${ActionType.ClientTweetFav.name}")),

uua = uuaClientTweetFav,

sinkTopic = "unified\_user\_actions\_engagements") shouldBe true

}

}

}

}