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

import com.twitter.context.thriftscala.Viewer

import com.twitter.inject.Test

import com.twitter.timelineservice.thriftscala.\_

import com.twitter.unified\_user\_actions.adapter.tls\_favs\_event.TlsFavsAdapter

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

import com.twitter.util.Time

class TlsFavsAdapterSpec extends Test {

trait Fixture {

val frozenTime = Time.fromMilliseconds(1658949273000L)

val favEventNoRetweet = ContextualizedFavoriteEvent(

event = FavoriteEventUnion.Favorite(

FavoriteEvent(

userId = 91L,

tweetId = 1L,

tweetUserId = 101L,

eventTimeMs = 1001L

)

),

context = LogEventContext(hostname = "", traceId = 31L)

)

val favEventRetweet = ContextualizedFavoriteEvent(

event = FavoriteEventUnion.Favorite(

FavoriteEvent(

userId = 92L,

tweetId = 2L,

tweetUserId = 102L,

eventTimeMs = 1002L,

retweetId = Some(22L)

)

),

context = LogEventContext(hostname = "", traceId = 32L)

)

val unfavEventNoRetweet = ContextualizedFavoriteEvent(

event = FavoriteEventUnion.Unfavorite(

UnfavoriteEvent(

userId = 93L,

tweetId = 3L,

tweetUserId = 103L,

eventTimeMs = 1003L

)

),

context = LogEventContext(hostname = "", traceId = 33L)

)

val unfavEventRetweet = ContextualizedFavoriteEvent(

event = FavoriteEventUnion.Unfavorite(

UnfavoriteEvent(

userId = 94L,

tweetId = 4L,

tweetUserId = 104L,

eventTimeMs = 1004L,

retweetId = Some(44L)

)

),

context = LogEventContext(hostname = "", traceId = 34L)

)

val favEventWithLangAndCountry = ContextualizedFavoriteEvent(

event = FavoriteEventUnion.Favorite(

FavoriteEvent(

userId = 91L,

tweetId = 1L,

tweetUserId = 101L,

eventTimeMs = 1001L,

viewerContext =

Some(Viewer(requestCountryCode = Some("us"), requestLanguageCode = Some("en")))

)

),

context = LogEventContext(hostname = "", traceId = 31L)

)

val expectedUua1 = UnifiedUserAction(

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

item = Item.TweetInfo(

TweetInfo(

actionTweetId = 1L,

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

)

),

actionType = ActionType.ServerTweetFav,

eventMetadata = EventMetadata(

sourceTimestampMs = 1001L,

receivedTimestampMs = frozenTime.inMilliseconds,

sourceLineage = SourceLineage.ServerTlsFavs,

traceId = Some(31L)

)

)

val expectedUua2 = UnifiedUserAction(

userIdentifier = UserIdentifier(userId = Some(92L)),

item = Item.TweetInfo(

TweetInfo(

actionTweetId = 2L,

actionTweetAuthorInfo = Some(AuthorInfo(authorId = Some(102L))),

retweetingTweetId = Some(22L)

)

),

actionType = ActionType.ServerTweetFav,

eventMetadata = EventMetadata(

sourceTimestampMs = 1002L,

receivedTimestampMs = frozenTime.inMilliseconds,

sourceLineage = SourceLineage.ServerTlsFavs,

traceId = Some(32L)

)

)

val expectedUua3 = UnifiedUserAction(

userIdentifier = UserIdentifier(userId = Some(93L)),

item = Item.TweetInfo(

TweetInfo(

actionTweetId = 3L,

actionTweetAuthorInfo = Some(AuthorInfo(authorId = Some(103L))),

)

),

actionType = ActionType.ServerTweetUnfav,

eventMetadata = EventMetadata(

sourceTimestampMs = 1003L,

receivedTimestampMs = frozenTime.inMilliseconds,

sourceLineage = SourceLineage.ServerTlsFavs,

traceId = Some(33L)

)

)

val expectedUua4 = UnifiedUserAction(

userIdentifier = UserIdentifier(userId = Some(94L)),

item = Item.TweetInfo(

TweetInfo(

actionTweetId = 4L,

actionTweetAuthorInfo = Some(AuthorInfo(authorId = Some(104L))),

retweetingTweetId = Some(44L)

)

),

actionType = ActionType.ServerTweetUnfav,

eventMetadata = EventMetadata(

sourceTimestampMs = 1004L,

receivedTimestampMs = frozenTime.inMilliseconds,

sourceLineage = SourceLineage.ServerTlsFavs,

traceId = Some(34L)

)

)

val expectedUua5 = UnifiedUserAction(

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

item = Item.TweetInfo(

TweetInfo(

actionTweetId = 1L,

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

)

),

actionType = ActionType.ServerTweetFav,

eventMetadata = EventMetadata(

sourceTimestampMs = 1001L,

receivedTimestampMs = frozenTime.inMilliseconds,

sourceLineage = SourceLineage.ServerTlsFavs,

language = Some("EN"),

countryCode = Some("US"),

traceId = Some(31L)

)

)

}

test("fav event with no retweet") {

new Fixture {

Time.withTimeAt(frozenTime) { \_ =>

val actual = TlsFavsAdapter.adaptEvent(favEventNoRetweet)

assert(Seq(expectedUua1) === actual)

}

}

}

test("fav event with a retweet") {

new Fixture {

Time.withTimeAt(frozenTime) { \_ =>

val actual = TlsFavsAdapter.adaptEvent(favEventRetweet)

assert(Seq(expectedUua2) === actual)

}

}

}

test("unfav event with no retweet") {

new Fixture {

Time.withTimeAt(frozenTime) { \_ =>

val actual = TlsFavsAdapter.adaptEvent(unfavEventNoRetweet)

assert(Seq(expectedUua3) === actual)

}

}

}

test("unfav event with a retweet") {

new Fixture {

Time.withTimeAt(frozenTime) { \_ =>

val actual = TlsFavsAdapter.adaptEvent(unfavEventRetweet)

assert(Seq(expectedUua4) === actual)

}

}

}

test("fav event with language and country") {

new Fixture {

Time.withTimeAt(frozenTime) { \_ =>

val actual = TlsFavsAdapter.adaptEvent(favEventWithLangAndCountry)

assert(Seq(expectedUua5) === actual)

}

}

}

}