package com.twitter.tweetypie

package core

import com.twitter.tweetypie.thriftscala.CardReference

import java.net.URI

sealed trait CardUri

object Tombstone extends CardUri

case class NonTombstone(uri: String) extends CardUri

object CardReferenceUriExtractor {

private def parseAsUri(cardRef: CardReference) = Try(new URI(cardRef.cardUri)).toOption

private def isTombstone(uri: URI) = uri.getScheme == "tombstone"

/\*\*

\* Parses a CardReference to return Option[CardUri] to differentiate among:

\* - Some(NonTombstone): hydrate card2 with provided uri

\* - Some(Tombstone): don't hydrate card2

\* - None: fallback and attempt to use url entities uris

\*/

def unapply(cardRef: CardReference): Option[CardUri] =

parseAsUri(cardRef) match {

case Some(uri) if !isTombstone(uri) => Some(NonTombstone(uri.toString))

case Some(uri) => Some(Tombstone)

// If a cardReference is set, but does not parse as a URI, it's likely a https? URL with

// incorrectly encoded query params. Since these occur frequently in the wild, we'll

// attempt a card2 hydration with it

case None => Some(NonTombstone(cardRef.cardUri))

}

}