package com.twitter.tweetypie

package hydrator

import com.twitter.tweetypie.core.\_

import com.twitter.tweetypie.thriftscala.\_

/\*\*

\* Remove contributor data from tweet if it should not be available to the

\* caller. The contributor field is populated in the cached

\* [[ContributorHydrator]].

\*

\* Contributor data is always available on the write path. It is available on

\* the read path for the tweet author (or user authenticated as the tweet

\* author in the case of contributors/teams), or if the caller has disabled

\* visibility filtering.

\*

\* The condition for running this filtering hydrator (onlyIf) has been a

\* source of confusion. Keep in mind that the condition expresses when to

\* \*remove\* data, not when to return it.

\*

\* In short, keep data when:

\* !reading || requested by author || !(enforce visibility filtering)

\*

\* Remove data when none of these conditions apply:

\* reading && !(requested by author) && enforce visibility filtering

\*

\*/

object ContributorVisibilityFilter {

type Type = ValueHydrator[Option[Contributor], TweetCtx]

def apply(): Type =

ValueHydrator

.map[Option[Contributor], TweetCtx] {

case (Some(\_), \_) => ValueState.modified(None)

case (None, \_) => ValueState.unmodified(None)

}

.onlyIf { (\_, ctx) =>

ctx.opts.cause.reading(ctx.tweetId) &&

!ctx.opts.forUserId.contains(ctx.userId) &&

ctx.opts.enforceVisibilityFiltering

}

}