namespace java com.twitter.tweetypie.geo.thriftjava

#@namespace scala com.twitter.tweetypie.geo.thriftscala

#@namespace strato com.twitter.tweetypie.geo

namespace py gen.twitter.tweetypie.geo

namespace rb TweetyPie

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// //

// This file contains type definitions to support the Geo field added to Tweet flexible schema ONLY. //

// It is unlikely to be re-usable so treat it them as private outside the subpackage defined here. //

// //

// In respect to back storage, consider it has limited capacity, provisioned to address particular use cases. //

// There is no free resources outside its current usage plus a future projection (see Storage Capacity below). //

// For example: //

// 1- Adding extra fields to TweetLocationInfo will likely require extra storage. //

// 2- Increase on front-load QPS (read or write) may require extra sharding to not impact delay percentiles. //

// Failure to observe these may impact Tweetypie write-path and read-path. //

// //

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

/\*\*

\* Flags how a \_Place\_ is published into a tweet (a.k.a. geotagging).

\*/

enum GeoTagPlaceSource {

/\*\*

\* Tweet is tagged to a place but it is impossible to determine its source.

\* E.g.: created from non-TOO clients or legacy TOO clients

\*/

UNKNOWN = 0

/\*\*

\* Tweet is tagged to a Place by reverse geocoding its coordinates.

\*/

COORDINATES = 1

/\*\*

\* Tweet is tagged to a Place by the client application on user's behalf.

\* N.B.: COORDINATES is not AUTO because the API request doesn't publish a Place

\*/

AUTO = 2

EXPLICIT = 3

// free to use, added for backwards compatibility on client code.

RESERVED\_4 = 4

RESERVED\_5 = 5

RESERVED\_6 = 6

RESERVED\_7 = 7

}

/\*\*

\* Information about Tweet's Location(s).

\* Designed to enable custom consumption experiences of the Tweet's location(s).

\* E.g.: Tweet's perspectival view of a Location entity

\*

\* To guarantee user's rights of privacy:

\*

\* - Only include user's published location data or unpublished location data that

\* is EXPLICITLY set as publicly available by the user.

\*

\* - Never include user's unpublished (aka shared) location data that

\* is NOT EXPLICITLY set as publicly available by the user.

\*

\* E.g.: User is asked to share their GPS coordinates with Twitter from mobile client,

\* under the guarantee it won't be made publicly available.

\*

\* Design notes:

\* - Tweet's geotagged Place is represented by Tweet.place instead of being a field here.

\*/

struct TweetLocationInfo {

/\*\*

\* Represents how the Tweet author published the "from" location in a Tweet (a.k.a geo-tagged).

\*/

1: optional GeoTagPlaceSource geotag\_place\_source

}(persisted='true', hasPersonalData='false')