namespace java com.twitter.tweetypie.thriftjava

#@namespace scala com.twitter.tweetypie.thriftscala

#@namespace strato com.twitter.tweetypie

namespace py gen.twitter.tweetypie.deprecated

namespace rb TweetyPie

include "com/twitter/expandodo/cards.thrift"

include "com/twitter/gizmoduck/user.thrift"

include "com/twitter/tweetypie/media\_entity.thrift"

include "com/twitter/tweetypie/tweet.thrift"

include "com/twitter/tweetypie/tweet\_service.thrift"

/\*\*

\* @deprecated Use Place

\*/

struct Geo {

/\*\*

\* @deprecated Use coordinates.latitude

\*/

1: double latitude = 0.0 (personalDataType = 'GpsCoordinates')

/\*\*

\* @deprecated Use coordinates.longitude

\*/

2: double longitude = 0.0 (personalDataType = 'GpsCoordinates')

/\*\*

\* @deprecated Use coordinates.geo\_precision

\*/

3: i32 geo\_precision = 0

/\*\*

\* 0: don't show lat/long

\* 2: show

\*

\* @deprecated

\*/

4: i64 entity\_id = 0

/\*\*

\* @deprecated Use place\_id

\*/

5: optional string name (personalDataType = 'PublishedCoarseLocationTweet')

6: optional tweet.Place place // provided if StatusRequestOptions.load\_places is set

7: optional string place\_id // ex: ad2f50942562790b

8: optional tweet.GeoCoordinates coordinates

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

/\*\*

\* @deprecated Use Tweet and APIs that accept or return Tweet.

\*/

struct Status {

1: i64 id (personalDataType = 'TweetId')

2: i64 user\_id (personalDataType = 'UserId')

3: string text (personalDataType = 'PrivateTweets, PublicTweets')

4: string created\_via (personalDataType = 'ClientType')

5: i64 created\_at // in seconds

6: list<tweet.UrlEntity> urls = []

7: list<tweet.MentionEntity> mentions = []

8: list<tweet.HashtagEntity> hashtags = []

29: list<tweet.CashtagEntity> cashtags = []

9: list<media\_entity.MediaEntity> media = []

10: optional tweet.Reply reply

31: optional tweet.DirectedAtUser directed\_at\_user

11: optional tweet.Share share

32: optional tweet.QuotedTweet quoted\_tweet

12: optional tweet.Contributor contributor

13: optional Geo geo

// has\_takedown indicates if there is a takedown specifically on this tweet.

// takedown\_country\_codes contains takedown countries for both the tweet and the user,

// so has\_takedown might be false while takedown\_country\_codes is non-empty.

14: bool has\_takedown = 0

15: bool nsfw\_user = 0

16: bool nsfw\_admin = 0

17: optional tweet.StatusCounts counts

// 18: obsoleted

19: optional tweet.DeviceSource device\_source // not set on DB failure

20: optional tweet.Narrowcast narrowcast

21: optional list<string> takedown\_country\_codes (personalDataType = 'ContentRestrictionStatus')

22: optional tweet.StatusPerspective perspective // not set if no user ID or on TLS failure

23: optional list<cards.Card> cards // only included if StatusRequestOptions.include\_cards == true

// only included when StatusRequestOptions.include\_cards == true

// and StatusRequestOptions.cards\_platform\_key is set to valid value

30: optional cards.Card2 card2

24: bool nullcast = 0

25: optional i64 conversation\_id (personalDataType = 'TweetId')

26: optional tweet.Language language

27: optional i64 tracking\_id (personalDataType = 'ImpressionId')

28: optional map<tweet.SpamSignalType, tweet.SpamLabel> spam\_labels

33: optional bool has\_media

// obsolete 34: optional list<tweet.TopicLabel> topic\_labels

// Additional fields for flexible schema

101: optional tweet.TweetMediaTags media\_tags

103: optional tweet.CardBindingValues binding\_values

104: optional tweet.ReplyAddresses reply\_addresses

105: optional tweet.TwitterSuggestInfo twitter\_suggest\_info

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