namespace java com.twitter.unified\_user\_actions.thriftjava

#@namespace scala com.twitter.unified\_user\_actions.thriftscala

#@namespace strato com.twitter.unified\_user\_actions

/\* Input source \*/

enum SourceLineage {

/\* Client-side. Also known as legacy client events or LCE. \*/

ClientEvents = 0

/\* Client-side. Also known as BCE. \*/

BehavioralClientEvents = 1

/\* Server-side Timelineservice favorites \*/

ServerTlsFavs = 2

/\* Server-side Tweetypie events \*/

ServerTweetypieEvents = 3

/\* Server-side SocialGraph events \*/

ServerSocialGraphEvents = 4

/\* Notification Actions responding to Your Highlights Emails \*/

EmailNotificationEvents = 5

/\*\*

\* Gizmoduck's User Modification events https://docbird.twitter.biz/gizmoduck/user\_modifications.html

\*\*/

ServerGizmoduckUserModificationEvents = 6

/\*\*

\* Server-side Ads callback engagements

\*\*/

ServerAdsCallbackEngagements = 7

/\*\*

\* Server-side favorite archival events

\*\*/

ServerFavoriteArchivalEvents = 8

/\*\*

\* Server-side retweet archival events

\*\*/

ServerRetweetArchivalEvents = 9

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

/\*

\* Only available in behavioral client events (BCE).

\*

\* A breadcrumb tweet is a tweet that was interacted with prior to the current action.

\*/

struct BreadcrumbTweet {

/\* Id for the tweet that was interacted with prior to the current action \*/

1: required i64 tweetId(personalDataType = 'TweetId')

/\*

\* The UI component that hosted the tweet and was interacted with preceeding to the current action.

\* - tweet: represents the parent tweet container that wraps the quoted tweet

\* - quote\_tweet: represents the nested or quoted tweet within the parent container

\*

\* See more details

\* https://docs.google.com/document/d/16CdSRpsmUUd17yoFH9min3nLBqDVawx4DaZoiqSfCHI/edit#heading=h.nb7tnjrhqxpm

\*/

2: required string sourceComponent(personalDataType = 'WebsitePage')

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

/\*

\* ClientEvent's namespaces. See https://docbird.twitter.biz/client\_events/client-event-namespaces.html

\*

\* - For Legacy Client Events (LCE), it excludes the client part of the

\* six part namespace (client:page:section:component:element:action)

\* since this part is better captured by clientAppid and clientVersion.

\*

\* - For Behavioral Client Events (BCE), use clientPlatform to identify the client.

\* Additionally, BCE contains an optional subsection to denote the UI component of

\* the current action. The ClientEventNamespace.component field will be always empty for

\* BCE namespace. There is no straightfoward 1-1 mapping between BCE and LCE namespace.

\*/

struct ClientEventNamespace {

1: optional string page(personalDataType = 'AppUsage')

2: optional string section(personalDataType = 'AppUsage')

3: optional string component(personalDataType = 'AppUsage')

4: optional string element(personalDataType = 'AppUsage')

5: optional string action(personalDataType = 'AppUsage')

6: optional string subsection(personalDataType = 'AppUsage')

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

/\*

\* Metadata that is independent of a particular (user, item, action type) tuple

\* and mostly shared across user action events.

\*/

struct EventMetadata {

/\* When the action happened according to whatever source we are reading from \*/

1: required i64 sourceTimestampMs(personalDataType = 'PrivateTimestamp, PublicTimestamp')

/\* When the action was received for processing internally

\* (compare with sourceTimestampMs for delay)

\*/

2: required i64 receivedTimestampMs

/\* Which source is this event derived, e.g. CE, BCE, TimelineFavs \*/

3: required SourceLineage sourceLineage

/\* To be deprecated and replaced by requestJoinId

\* Useful for joining with other datasets

\* \*/

4: optional i64 traceId(personalDataType = 'TfeTransactionId')

/\*

\* This is the language inferred from the request of the user action event (typically user's current client language)

\* NOT the language of any Tweet,

\* NOT the language that user sets in their profile!!!

\*

\* - ClientEvents && BehavioralClientEvents: Client UI language or from Gizmoduck which is what user set in Twitter App.

\* Please see more at https://sourcegraph.twitter.biz/git.twitter.biz/source/-/blob/finatra-internal/international/src/main/scala/com/twitter/finatra/international/LanguageIdentifier.scala

\* The format should be ISO 639-1.

\* - ServerTlsFavs: Client UI language, see more at http://go/languagepriority. The format should be ISO 639-1.

\* - ServerTweetypieEvents: UUA sets this to None since there is no request level language info.

\*/

5: optional string language(personalDataType = 'InferredLanguage')

/\*

\* This is the country inferred from the request of the user action event (typically user's current country code)

\* NOT the country of any Tweet (by geo-tagging),

\* NOT the country set by the user in their profile!!!

\*

\* - ClientEvents && BehavioralClientEvents: Country code could be IP address (geoduck) or

\* User registration country (gizmoduck) and the former takes precedence.

\* We don’t know exactly which one is applied, unfortunately,

\* see https://sourcegraph.twitter.biz/git.twitter.biz/source/-/blob/finatra-internal/international/src/main/scala/com/twitter/finatra/international/CountryIdentifier.scala

\* The format should be ISO\_3166-1\_alpha-2.

\* - ServerTlsFavs: From the request (user’s current location),

\* see https://sourcegraph.twitter.biz/git.twitter.biz/source/-/blob/src/thrift/com/twitter/context/viewer.thrift?L54

\* The format should be ISO\_3166-1\_alpha-2.

\* - ServerTweetypieEvents:

\* UUA sets this to be consistent with IESource to meet existing use requirement.

\* see https://sourcegraph.twitter.biz/git.twitter.biz/source/-/blob/src/thrift/com/twitter/tweetypie/tweet.thrift?L1001.

\* The definitions here conflicts with the intention of UUA to log the request country code

\* rather than the signup / geo-tagging country.

\*/

6: optional string countryCode(personalDataType = 'InferredCountry')

/\* Useful for debugging client application related issues \*/

7: optional i64 clientAppId(personalDataType = 'AppId')

/\* Useful for debugging client application related issues \*/

8: optional string clientVersion(personalDataType = 'ClientVersion')

/\* Useful for filtering \*/

9: optional ClientEventNamespace clientEventNamespace

/\*

\* This field is only populated in behavioral client events (BCE).

\*

\* The client platform such as one of ["iPhone", "iPad", "Mac", "Android", "Web"]

\* There can be multiple clientAppIds for the same platform.

\*/

10: optional string clientPlatform(personalDataType = 'ClientType')

/\*

\* This field is only populated in behavioral client events (BCE).

\*

\* The current UI hierarchy information with human readable labels.

\* For example, [home,timeline,tweet] or [tab\_bar,home,scrollable\_content,tweet]

\*

\* For more details see https://docs.google.com/document/d/16CdSRpsmUUd17yoFH9min3nLBqDVawx4DaZoiqSfCHI/edit#heading=h.uv3md49i0j4j

\*/

11: optional list<string> viewHierarchy(personalDataType = 'WebsitePage')

/\*

\* This field is only populated in behavioral client events (BCE).

\*

\* The sequence of views (breadcrumb) that was interacted with that caused the user to navigate to

\* the current product surface (e.g. profile page) where an action was taken.

\*

\* The breadcrumb information may only be present for certain preceding product surfaces (e.g. Home Timeline).

\* See more details in https://docs.google.com/document/d/16CdSRpsmUUd17yoFH9min3nLBqDVawx4DaZoiqSfCHI/edit#heading=h.nb7tnjrhqxpm

\*/

12: optional list<string> breadcrumbViews(personalDataType = 'WebsitePage')

/\*

\* This field is only populated in behavioral client events (BCE).

\*

\* The sequence of tweets (breadcrumb) that was interacted with that caused the user to navigate to

\* current product surface (e.g. profile page) where an action was taken.

\*

\* The breadcrumb information may only be present for certain preceding product surfaces (e.g. Home Timeline).

\* See more details in https://docs.google.com/document/d/16CdSRpsmUUd17yoFH9min3nLBqDVawx4DaZoiqSfCHI/edit#heading=h.nb7tnjrhqxpm

\*/

13: optional list<BreadcrumbTweet> breadcrumbTweets(personalDataType = 'TweetId')

/\*

\* A request join id is created by backend services and broadcasted in subsequent calls

\* to other downstream services as part of the request path. The requestJoinId is logged

\* in server logs and scribed in client events, enabling joins across client and server

\* as well as within a given request across backend servers. See go/joinkey-tdd for more

\* details.

\*/

14: optional i64 requestJoinId(personalDataType = 'TransactionId')

15: optional i64 clientEventTriggeredOn

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