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

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

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

/\*

\* Internal key used for controling UUA enrichment & caching process. It contains very minimal

\* information to allow for efficient serde, fast data look-up and to drive the partioning logics.

\*

\* NOTE: Don't depend on it in your application.

\* NOTE: This is used internally by UUA and may change at anytime. There's no guarantee for

\* backward / forward-compatibility.

\* NOTE: Don't add any other metadata unless it is needed for partitioning logic. Extra enrichment

\* metdata can go into the envelop.

\*/

struct EnrichmentKey {

/\*

\* The internal type of the primary ID used for partitioning UUA data.

\*

\* Each type should directly correspond to an entity-level ID in UUA.

\* For example, TweetInfo.actionTweetId & TweetNotification.tweetId are all tweet-entity level

\* and should correspond to the same primary ID type.

\*\*/

1: required EnrichmentIdType keyType

/\*\*

\* The primary ID. This is usually a long, for other incompatible data type such as string or

\* a bytes array, they can be converted into a long using their native hashCode() function.

\*\*/

2: required i64 id

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

/\*\*

\* The type of the primary ID. For example, tweetId on a tweet & tweetId on a notification are

\* all TweetId type. Similarly, UserID of a viewer and AuthorID of a tweet are all UserID type.

\*

\* The type here ensures that we will partition UUA data correctly across different entity-type

\* (user, tweets, notification, etc.)

\*\*/

enum EnrichmentIdType {

TweetId = 0

}