Codebook for MultiLA web API raw tracking data

The data export archive that can be downloaded from the MultiLA administration interface in the *Data manager > Export* view contains three files in CSV format, all of which can be joined via common identifiers that are documented below and highlighted in **bold.**

File `app_sessions.csv`

Contains data on application sessions, i.e. information on applications and the configured sessions that can be visited by users.

- `app_id`: ID of the application – integer
- `app_name`: name of the application – character string
- `app_url`: URL where the application was served – character string
- `app_config_id`: ID of the application configuration – integer
- `app_config_label`: name of the application configuration – character string
- **"app_sess_code": session code of the application session (session code for a configured application that was shared to the users) – character string**
- `app_sess_auth_mode`: authentication mode of the application session – categorical; `"none"` or `"login"`

File `tracking_sessions.csv`

Contains data on user and tracking sessions, i.e. information on users and their interaction sessions with the applications starting with the first visit of an application session and ending with closing the browser window.

- **"app_sess_code": session code of the application session (session code for a configured application that was shared to the users) – character string**
- `user_app_sess_code`: user application session code (session code for an individual anonymous or registered user interacting with a specific application session) – character string

- `user_app_sess_user_id`: user ID for registered users; no further data on individual users is provided in this dataset – integer for registered users or NA for anonymous users
- **"track_sess_id": tracking session ID (ID indicating a continuous interaction of a user with the application session on a single device) – integer**
- `track_sess_start`: start of the tracking session (first visit of a user on this device for this application session) – UTC date and time in format `Y-M-D H:M:S.ms`
- `track_sess_end`: end of the tracking session (user closes the browser window of logs out) – UTC date and time in format `Y-M-D H:M:S.ms`
- `track_sess_device_info`: information on the device used by the user in this tracking session – JSON with the following information:
  - `user_agent`: "user agent" string from the browser – character string
  - `client_ip`: client IP address (if it could be determined)
  - `form_factor`: categorical; `"desktop"`, `"tablet"` or `"phone"`
  - `window_size`: browser window size; array with two elements as integers: `[window width, window height]`
  - `main_content_viewsize`: main content view size; array with two elements as integers: `[window width, window height]`
  - `main_content_scrollsize`: main content scroll area size; array with two elements as integers: `[window width, window height]`

File `user_feedback.csv`

Contains data on user feedback provided by users during a user application session.

- **"app_sess_code": session code of the application session (session code for a configured application that was shared to the users) – character string**
- **"user_app_sess_code": user application session code (session code for an individual anonymous or registered user interacting with a specific application session) – character string**
- `user_app_sess_user_id`: user ID for registered users; no further data on individual users is provided in this dataset – integer for registered users or NA for anonymous users
- **"track_sess_id": optional tracking session ID (ID indicating a continuous interaction of a user with the application session on a single device) – integer**
- `feedback_created`: time when the event took place – UTC date and time in format `Y-M-D H:M:S.ms`
- `feedback_content_section`: content section of the application for which the feedback was recorded
- `feedback_score`: score given by the user in range [0, 1]; can be NA when

no score was given or giving scores was disabled
- **feedback_text**: comment given by the user; can be NA when giving comments was disabled

File `tracking_events.csv`

Contains data on events produced by users within a tracking session.

- **"track_sess_id": tracking session ID (ID indicating a continuous interaction of a user with the application session on a single device) – integer**
- **event_time**: time when the event took place – UTC date and time in format `Y-M-D H:M:S.ms`
- **event_type**: type of the event – categorical; `"device_info_update"`, `"visibility_change"`, `"summary_shown"`, `"summary_topic_added"`, `"input_change"`, `"learnr_event_*"`, `chatbot_communication` (see below for possible *learnr* events in `*` placeholder) or `"mouse"`
- **event_value**: event data – JSON; depends on `event_type`:
  - for `"device_info_update"`: changed window and/or main content sizes as as in `track_sess_device_info` column in `tracking_sessions.csv`
  - for `"visibility_change"`: document visibility state change; object with key state that can be either of value "visible" or "hidden"'
  - for `"summary_shown"`: always `null`; simply indicates that the summary panel was first shown at that time
  - for `"summary_topic_added"`: an object with `id` (summary content HTML ID) and `key` (summary content index as `<chapter index>.<summary content index>`)
  - for `"learnr_event_*"`: see learnr documentation and "learnr events" section below
  - for `chatbot_communication` an object with the following keys and values:
    * `user`: the user prompt
    * `assistant`: the bot response
    * `assistant_content_section_ref`: the linked content section if provided by bot response
    * `model`: the used chat API provider and model
  - for `"input_change"`: an object with the following keys and values:
    * `id`: HTML ID of the tracked input element – may be empty
    * `xpath`: XPath to the tracked input element
    * `value`: new value of the input element
  - for `"mouse"`: raw mouse tracking data as collected with an adapted version of mus.js; data is collected in chunks and must be concatenated to form the trace for the whole tracking session
    * `frames`: array with mouse interactions; each item is an array

[type, ..., timestamp] which indicates that the item type always comes first and the timestamp always comes last; a variable number of elements which depends on the item type comes between the type and the timestamp

- · `type` can be: `"m"` – move; `"c"` – click; `"s"` – scroll; `"i"` – key input; `"o"` – input value change (sliders, checkboxes, etc.)
- · for types `"m"`, `"c"`, `"s"` follow two coordinates as cursor x and y positions within the window
- · for types `"m"`, `"c"`, `"i"`, `"o"` follow the XPath and the CSS selector for the current element or `null` if the element is the same as in the previous record
- · for types `"i"`, `"o"` follows the the entered/changed element value
- · `timestamp` is the time passed since `startedAtISODate` (see below) in ms
- ∗ `window`: window size
- ∗ `timeElapsed`: time in ms since mouse tracking started
- ∗ `startedAtISODate`: ISO-8601 date string that indicates the start of the mouse data recording

Learnr events

- `exercise_hint`: User requested a hint or solution for an exercise.
- `exercise_submitted`: User submitted an answer for an exercise.
- `exercise_result`: The evaluation of an exercise has completed.
- `question_submission`: User submitted an answer for a multiple-choice question.
- `video_progress`: User watched a segment of a video.
- `section_skipped`: A section of the tutorial was skipped.
- `section_viewed`: A section of the tutorial became visible.