LANG-TRACK-APP HANDBOOK FOR RESEARCHERS

Version 2022-12-20

Dr Henriette Arndt
Lund University Humanities Lab

This document contains instructions for how researchers can set up and run studies with the LANG-TRACK-APP once all system components (the mobile app, admin website, and backend server) have been set up by your system developers. The necessary code and other information for developers is available on our GitHub page: https://github.com/HumlabLu/

To learn more about the LANG-TRACK-APP and the Experience Sampling Method, please refer to the following publications:

Arndt, HL, Granfeldt, J & Gullberg, M (2022). The Lang-Track-App: Open-source tools for implementing the Experience Sampling Method in Second Language Acquisition Research. *Language Learning*, https://onlinelibrary.wiley.com/doi/10.1111/lang.12555

Arndt, HL, Granfeldt, J & Gullberg, M (2021). Reviewing the potential of the Experience Sampling Method (ESM) for capturing language exposure and use. *Second Language Research*, https://journals.sagepub.com/doi/10.1177/02676583211020055

Table of Contents

1. End-to-end process for new studies

This is an overview of the steps for starting and ending a study with the LANG-TRACK-APP. Each step is elaborated in a later section, with links to those sections highlighted in blue.

1.1 Starting a study

Make sure any previous study has ended

Edit the app menus (Dynamic text) in Firebase.

Add new users in Firebase and send log-in information to study participants.

Code surveys in JSON format.

On the Admin Website:

Assign users to groups (e.g., by study or subsample).

Upload survey code.

Schedule surveys.

1.2 Ending a study

Download survey data from the Admin Website.

Clear the live database.

Delete users from Firebase.

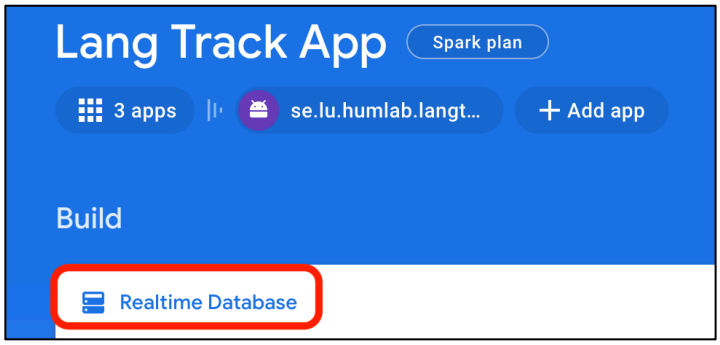
2. Firebase

To access firebase, go to https://console.firebase.google.com/ and log in with the username and password that your developers have shared with you.

Under *Your Firebase projects*, click on the project that has been set up for the LANG-TRACK-APP.

2.1 Dynamic text

To update the text that is displayed in the menu of the LANG-TRACK-APP, click on *Realtime Database* under the section *Build*.



Update the relevant text, for example under *Contact Info* and *Team*.

2.2 Add new users

Create a list of random usernames and passwords for participants, e.g., in Excel.

Usernames must be unique!

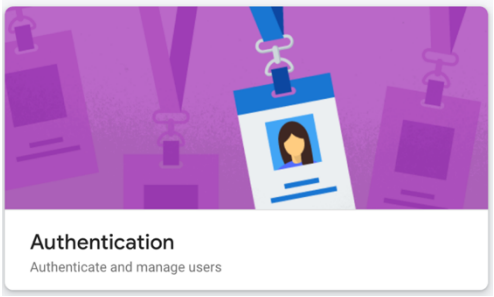
Usernames and passwords can use a-z and/or 0-9.

TIP: In Excel, the formula *=CHAR(RANDBETWEEN(97,122))* generates a random lowercase letter and *=RANDBETWEEN(100,999)* a three-digit number.

You can change the formula combination to your need. For example, for an alphanumeric password of the format ab123c use this formula:

=CHAR(RANDBETWEEN(97,122))&CHAR(RANDBETWEEN(97,122))&RANDBETWEEN(100,999)&CHAR(RANDBETWEEN(97,122))

In Firebase, scroll down to the section entitled *Store and sync app data in milliseconds* and click on *Authentication*.



Click on the *Add user* button and paste in your previously generated usernames and passwords, one-by-one.

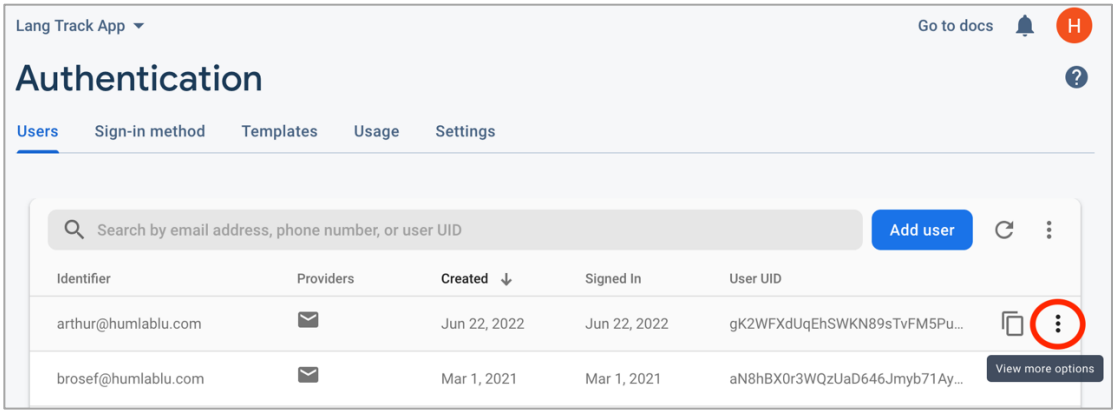
In Firebase (not in the LANG-TRACK-APP itself), usernames must be entered in the format of an email, for example: username@university.com (e.g., 123xyz@humlablu.com).

Send each participant one unique set of login details (username *without '@university.com'* and password).

2.3 Delete users

In Firebase, scroll down to the section titled *Store and sync app data in milliseconds* and click on *Authentication*.

Click on the three dots next to the relevant User ID and select *Delete account* from the menu.



3. Database

The database should be cleared between studies because

(a) the service gets slower as the database grows, and
(b) some of the stored data is considered sensitive.

The process for clearing the database will differ depending on how it was set up. Ask your systems developer to show you how you can access and clear the database.

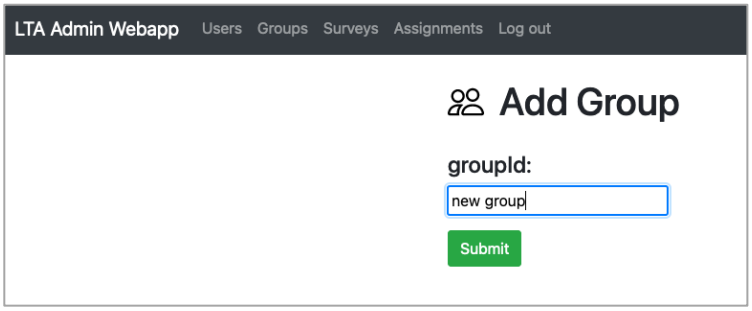
4. Admin website

Open a web browser (Chrome works best), navigate to the URL where you have set up your Admin site, and log in with your admin username and password.

You can add users and designate them as admins in Firebase.

4.1 Create/manage user groups

User groups make it easier to schedule surveys for multiple research participants at once. To create a new group, go to the *Groups* tab and click on *Add Group*.

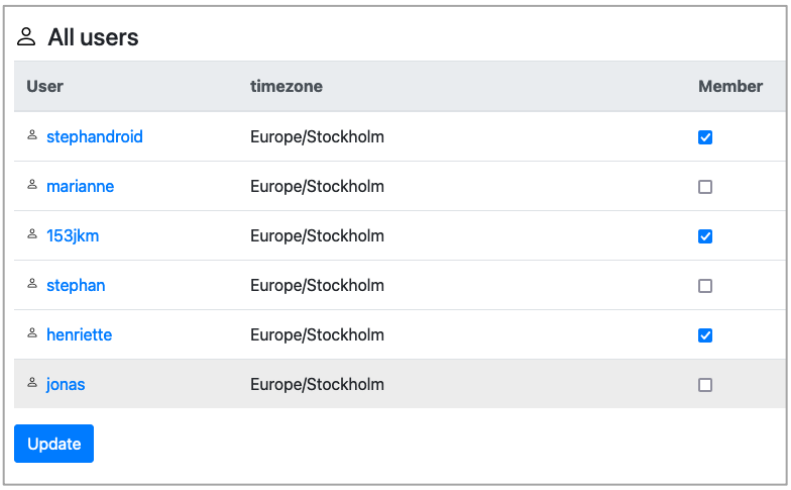


Enter a name for the new group into the *groupId* field and click *Submit*.

You are now sent to the newly created page for this group. You can access this page again later by going to the *Groups* tab and clicking on the group name. The page shows when the group was created and updated, a list of the group members, and a list of survey assignments that have been scheduled for the group.

To add/remove users, click on *Edit* in the *Members* section of the group page.

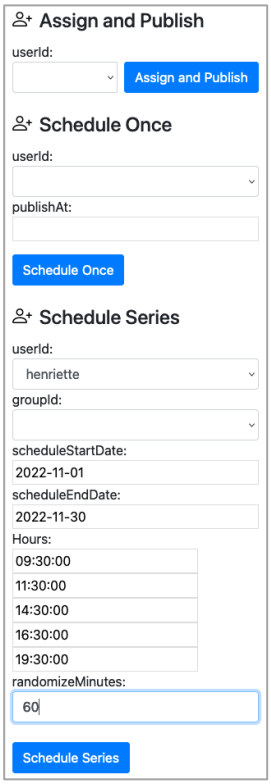
Tick the checkbox for each user that you want to add to the group or untick it for each user you want to remove. Click the *Update* button to save the new list of group members.



4.2 Upload surveys

To upload a new survey, go to the *Surveys* tab and click on *Add Survey*. Paste your JSON code into the box labelled *Config* and click *Submit*.

You are now sent to the newly created page for this survey. You can access this page again later by going to the *Surveys* tab and clicking on the survey name. The page shows when the survey was created and how many questions it contains. You can schedule survey assignments for individual users or user groups and download the data that has been collected with this survey. At the bottom of the page, you can also see a list of assignments that have been scheduled for the survey.



4.3 Schedule surveys

Click on the *Surveys* tab to see the list of uploaded surveys and click on the name of the survey you'd like to assign. At the top of the page, you can see basic information like when this survey was created, how many questions it contains, and the full JSON code.

Further down, there are three sections for scheduling surveys in different ways:

1. **Assign and publish**: Push a survey to one user immediately — useful for researchers to test out their surveys in the app. Select one user from the *userId* dropdown menu, then click the *Assign and Publish* button.

2. **Schedule once**: Schedule a survey for one user at one specific time point in the future. Select one user from the dropdown menu, then click the publishAt field and select a date and time. Confirm with the *Schedule Once* button.

3. **Schedule series:** Schedule surveys for one user (group) repeatedly over a period of time. Select one user *or* a user group from the dropdown menus. Specify when you want the user(s) to receive the survey by selecting the start and end date (inclusive) and specifying at least one time point (select hours and minutes in the pop-up menu). Confirm with the *Schedule Series* button.

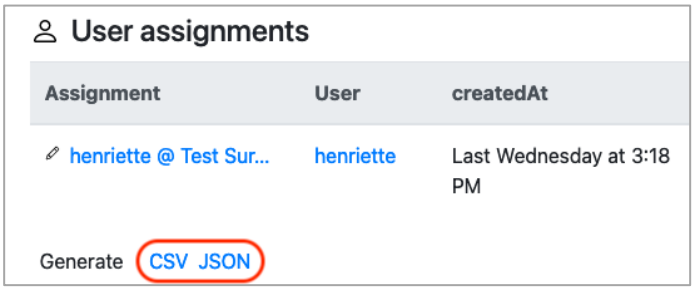
Optional: The exact time a survey is pushed to each user can be randomized around the previously selected hours by specifying a time frame in minutes. For example, specify "Hours: 09:30" and "randomizeMinutes: 60" to randomize a time between 08:30 and 10:30). To allow for double-checking after a survey has been scheduled, the assignment will show up in the *Assignments* section on the survey, user and/or group page, and in the *Assignments* tab. These lists also show whether push notifications have been sent, and whether a user has responded to a survey assignment.

IMPORTANT NOTE: New assignments may not appear in the list immediately after pushing a button to schedule a survey. Get the site to update the cache by clicking the *Refresh* button in your browser while holding down the *Tab key* on your keyboard.

Do not press the *Assign and Publish* / *Schedule Once* / *Schedule series* button more than once — this will lead to double scheduling! There is currently no way to delete accidental assignments except for clearing the entire database.

4.4 Download data

Click on the *Surveys* tab to see the list of uploaded surveys and click on the name of the survey for which you'd like to download the data.



Scroll down to the *Assignments* section. Data from assignments of this survey to individual users can be downloaded by clicking on *CSV* or *JSON* at the bottom of the *User Assignments* list. Similarly, clicking on *CSV* or *JSON* at the bottom of the *Group Assignments* list exports the data from all assignments of this survey to groups of users. If the survey has been assigned both to individual users and groups separate CSV files should be merged before analysis (e.g., in Excel or R).

5. JSON code survey files

Surveys are coded in JavaScript Open Notation (JSON). Code can be written in dedicated software, or simple text editor (e.g., Notepad on Windows or TextEdit on Mac). As an example, you can take a look at two surveys about language use which we implemented in our first studies with the LANG-TRACK-APP: https://www.iris-database.org/iris/app/home/detail?id=york%3a940658

You can use an online service to check that there are no errors in your JSON file, such as: https://jsonlint.com/. This will help to make sure that there are no missing commas, brackets, etc.

5.1 Basic survey structure

All survey files need to have the following basic elements and structure:

```
{
      "title": "Survey title as shown to participants in the app",
      "name": "Survey name as shown to researchers on the admin site",
      "publishNotificationTitle": "Initial push notification title",
      "publishNotificationBody": "Initial push notification message",
      "expireNotificationTitle": "Reminder push notification title",
      "expireNotificationBody": "Reminder push notification message",
      "questions":
      [
            {Insert survey elements},
            {including questions, header, footer,
            and other text (see below)}
      ]
}
```

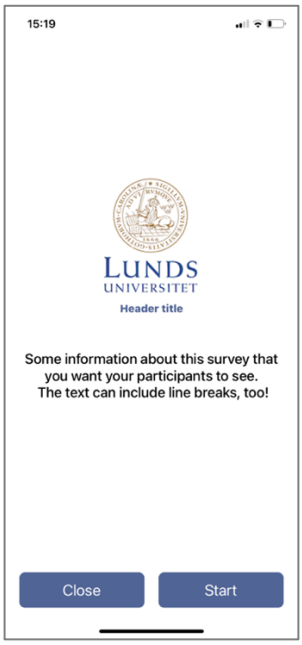
Highlighted text = to be replaced with your own contents.

5.2 Survey elements

The "questions" section of the JSON file can contain any number of the following elements (enclosed in curly brackets and comma-separated). They are displayed in the app one-by-one, following the numerical index order.

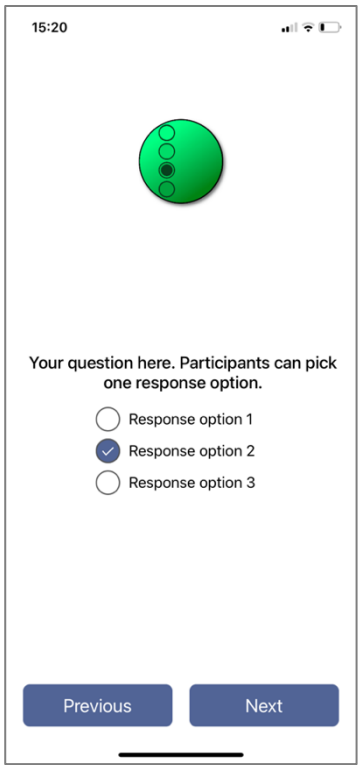
Header text:

```
{
   "index": 0,
   "type": "header",
   "title": "Header title",
   "text": "Some information about
this survey that you want your
participants to see.\n The text
can include line breaks, too!"
},
```



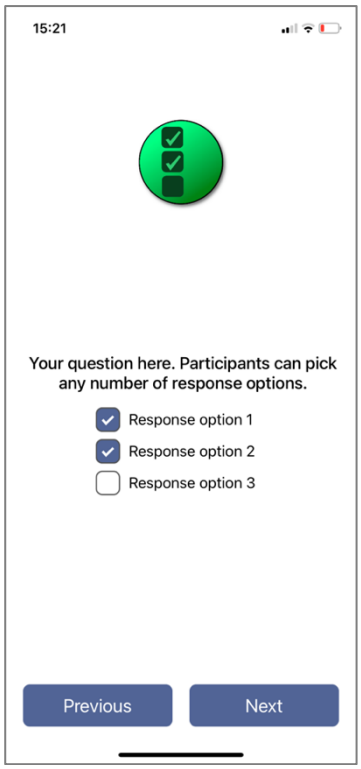
Single-response multiple choice:

```
{
   "index": 1,
   "type": "single",
   "text": "Your question here.
Participants can pick one response
option.",
   "values":
      [
         "Response option 1",
         "Response option 2",
         "Response option 3"
      ]
},
```



Multiple-response multiple choice:

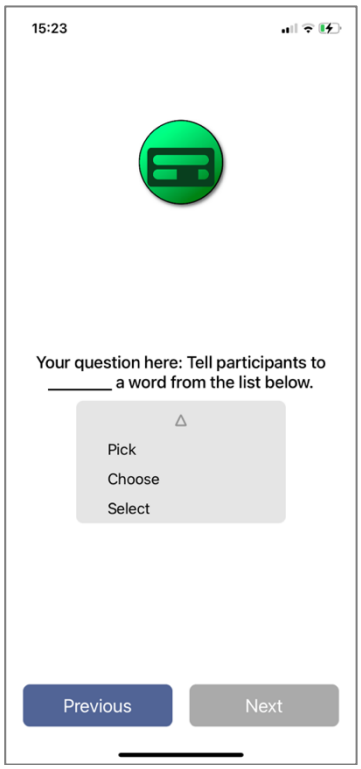
```
{
   "index": 2,
   "type": "multi",
   "text": "Your question here.
Participants can pick any number
of response options.",
   "values":
      [
         "Response option 1",
         "Response option 2",
         "Response option 3"
      ]
},
```



Fill-in-the-blank:

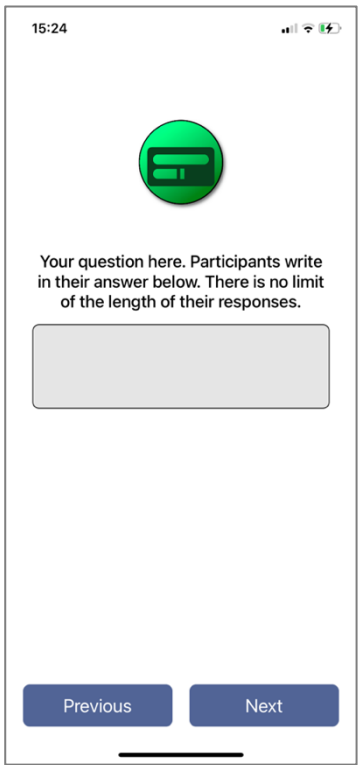
```
{
   "index": 3,
   "type": "blanks",
   "text": "Your question here:
Tell participants to _____ a word
from the list below.",
   "values":
      [
         "Pick",
         "Choose",
         "Select"
      ]
},
```

NOTE: The placeholder must consist of exactly 5 underscores and must have an empty space on each side (e.g., _____. will not work!).



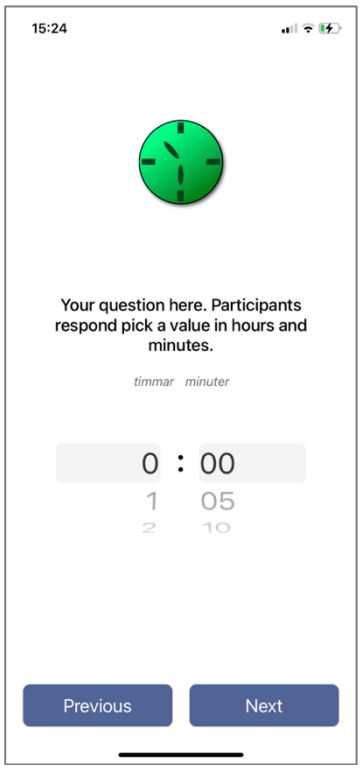
Open-ended text:

```
{
   "index": 4,
   "type": "open",
   "text": "Your question here. Participants write in their answer below. There is no limit of the length of their responses."
},
```



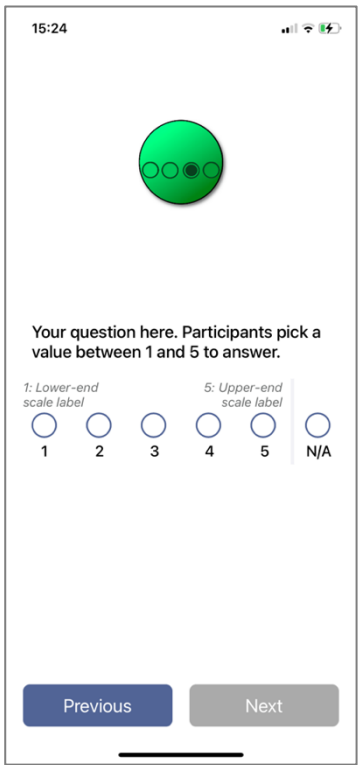
Questions about time:

```
{
   "index": 5,
   "type": "duration",
   "text": "Your question here. Participants respond pick a value in hours and minutes."
            },
```



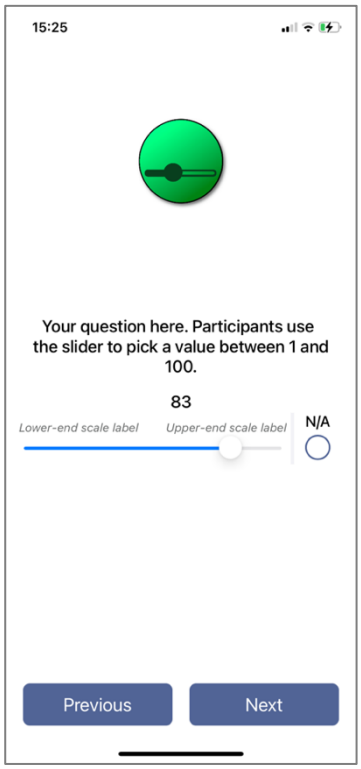
5-point Likert scale:

```
{
   "index": 6,
   "type": "likert",
   "text": "Your question here. Participants pick a value between 1 and 5 to answer.",
   "minAnnotation": "1: Lower-end scale label",
   "maxAnnotation": "5: Upper-end scale label"
},
```



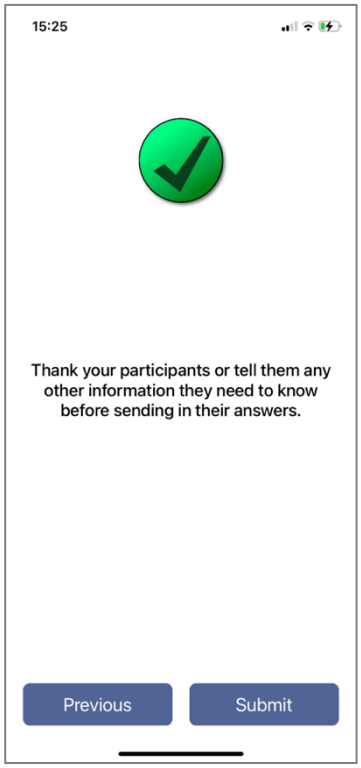
Slider scale:

```
{
   "index": 7,
   "type": "slider",
   "text": "Your question here. Participants use the slider to pick a value between 1 and 100.",
   "minAnnotation": "Lower-end scale label",
   "maxAnnotation": "Upper-end scale label"
},
```



Footer text:

```
{
   "index": 8,
   "type": "footer",
   "text": "Thank your
participants or tell them any
other information they need to
know before sending in their
answers."
            }
```



5.3 Survey logic

The **skip** function sends users to a later point in the survey (i.e., skip some elements) depending on the answer they gave to the current question.

For example:

```
{
   "index": 1,
   "type": "single",
   "text": "Do you want to answer
a lot of questions?",
   "values":
      [
         "Yes",
         "No"
      ],
   "skip":
      {
         "ifChosen": 1,
         "goto": 10
      }
},
```

In this case, if the participant chose “No”, then the survey skips to the question with index 10. Lists always start with index 0, so in this list of “values”, “Yes” has index 0 and “No” has index 1.

The **includeIf** function displays a question only to users who gave a certain answer to an earlier question.

For example:

```
{
   "index": 2,
   "type": "single",
   "text": "Why do you like
answering questions so much?",
   "values":
      [
         "I'm bored",
         "I'm crazy",
         "You're paying me",
         "I selected the wrong
      answer"
      ],
   "skip":
      {
         "ifIndex": 1,
         "ifValue": 0
      }
}
```

ifIndex refers to the index of the question to check, and ifValue is the value to check in that question.

The question on the left (index 2) will only be shown if the participant selected value 0 in the question with index 1 (i.e., they selected "Yes" in the question above).

If the condition is not fulfilled, the current question is hidden. The program then checks the next indices one-by-one until it finds an element without includeIf or where the includeIf conditions are fulfilled. That element will be shown next.