

DEPRECATED - visit vd-mit wiki

<https://userwiki.lst.uni-saarland.de/exc/doku.php?id=wiki:start>

README

This file contains tips and tricks for LingoTurk. Please use a similar vignette as the ones below (Header 1 for the topic and include information on the author, date and file that the code is for). To see the code snippets you might have to install the code-block add-on (see here: <https://theproductiveengineer.net/how-to-add-code-blocks-to-google-docs/>)

LingoTurk protocol

By: Marian
Created: 13/12/2023
Last updated:
File: –

When you are new to LingoTurk (or when you have a student), please read the following:

<https://git.sfb1102.uni-saarland.de/aanwari/lingoturk-multitude/-/blob/master/PROTOCOLS.md>

General structure of LingoTurk

By: Marian
Created: 03/10/2022
Last updated:
File: all

LingoTurk runs on two different servers:

- masses used for designing experiments
- multitude used for making final changes and running experiments

You can access the servers in two ways:

- frontend user interface via SERVERNAME.coli.uni-saarland.de:8080
- backend connecting with the servers using the terminal or an FTP like Filezilla
 - masses: /shared/newLT/lingoturk-masses

- multitude: /shared/LTCopy/lingoturk

Personally, I use the frontend to create and check the experiments and the backend to make small changes. For the latter, you need some basic programming skills and access to the servers (ask Mansoor). If you lack either of those, you can ask Mansoor to help you.

The file structure of an experiment looks like this:

- js contains procedure of the experiment
 - public/javascripts/ExperimentRendering/EXPNAMEExperiment.js
- html what the experiment looks like to the user
 - app/views/ExperimentRendering/EXPNAMEExperiment.js
- css not sure, maybe custom functions?
 - public/stylesheets/ExperimentRendering/EXPNAMEExperiment.css
- properties/json/sql general structure of the experiment, field names for csv files and code to retrieve data
 - app/models/Questions/EXPNAMEExperiment

Best practice for saving experiment

By: Marian
Created: 03/10/2022
Last updated:
File: all

Ever since I once lost one of my experiments (it was accidentally deleted), I now always download my experiment (the zip file) as soon as I publish my experiment. I do this to make sure that I can always go back to see what my experiment looks like and run it again in case anything gets deleted. Note that this will only contain the experiment script and not the results.

Activate the statistics slide

By: Iza
Created: 14/11/2022
Last updated:
File: js

In the js file, set `self.useStatistics` to true, so

```
self.useStatistics = true;
```

Question types statistics slide

By: Marian
Created: 30/09/2022
Last updated:
File: js

Statistics questions are created in a json like format:

```
self.statistics= [  
  {name: "What's your age?", // question text  
    type: "number" // question type, see below  
    answer: "" // set to empty or undefined  
  },  
  {name: ...}]
```

Questions can be of various types:

- Number

```
{name: "Age", type: "number",  
answer: undefined}
```

- Text

```
{name: "Gender",  
type: "text",  
answer: ""}
```

- Boolean

```
{name: "Are you bilingual?",  
type: "boolean", answer : undefined}
```

- Single choice

```
{name: "I liked this task (1=not at all, 5=very much)",  
type: "choice",  
choices: ["1", "2", "3", "4", "5"],  
answer: undefined}
```

- Multiple choice

```
{name: "Please select all that apply.",  
type: "multichoice",  
choices: ["A", "B", "C"],  
answer: undefined}
```

You can also set conditionals

```
{name: "Are you bilingual?",
```

```
type: "choice",
choices: ["Yes", "No"], answer: undefined,
      conditional: [{on: "Yes", statistics: [
        {name: "Please indicate which languages:"},
type: "text",
answer : ""}}]]}]}
```

Add a question counter in the header

By: Marian
Created: 30/09/2022
Last updated:
File: html

To add a counter in the header of the experimental pages, use `{{$index+1}}` (for the question number) and `{{RC.questions.length}}` for the total number of questions

It will look something like this:

```
<div id="experiment_heading_1_1" class="col-md-8 panel-heading"
style="font-weight: bolder;">
    Question {{$index+1}}/{{RC.questions.length}}: Provide a
logical continuation to the prompt below
</div>
```

Block copying and pasting

By: Iza
Created: 30/09/2022
Last updated:
File: html

To prevent participants from cheating (copying text from experimental items or pasting solutions) add this piece of code in the very beginning of the HTML file.

```
<script>
$(document).ready(function() {
    $('body').bind('cut copy', function(e) {
        alert("Kopieren und Einfuegen ist deaktiviert");
    });
});
```

```
</script>
```

Experiment keeps looping

By: Marian
Created: 03/10/2022
Last updated:
File: js

Comment out this.load() as shown below:

```
// Function to switch slides. Adding it into default file.  
this.switchSlide = function(){  
self.subslide = self.subslide + 1;  
//this.load();  
};
```

Next button not enabled

By: Marian
Created: 03/10/2022
Last updated:
File: html

If 'Next' button is not enabled after requirement is satisfied, replace
`question.answer.answer` by `question.answer`

```
<div id="experiment_body_1_1" class="panel-body">  
  <div style="display: flex; align-items: center;">  
    <div style="width: 300px; margin-left: 10px;">  
      <input type="text"  
ng-model="question.answer.answer.continuation" class="form-control">  
    </div>  
  </div>  
  <button ng-click="RC.nextQuestion()" type="button" class="btn  
btn-default" style="float:center ; margin-top:20px"  
ng-disabled="question.answer.continuation === undefined ||  
question.answer.continuation === '' || question.answer.continuation ==  
{}">Next</button>  
</div>
```

Accidentally multiple lists

By: Marian
Created: 03/10/2022
Last updated:
File: `properties`

An experiment type can either have a single list or multiple lists. This will show up as different links to the same experiment when publishing. You can still change this later by replacing `listType = MULTIPLE LISTS` with `listType = DISJOINT LISTS`

Html formatting in input files

By: Marian
Created: 03/10/2022
Last updated:
File: `js & html`

You can add html formatting inside the csv file (materials) that you upload. For example, I would upload:

Ian went home **because** he felt sick.

and wanted this to look like

Ian went home **because** he felt sick.

However, this would not always work. In these cases, replace in the `html` file, the following lines

```
<center>
<span><font size="+1">{{"question.targetitem"}}</span></font></span>
</center>
```

with

```
<center>
<span><font size="+1"><span ng-bind-html="question.targetitem |
unsafe"></span></font></span>
</center>
```

In addition, change in the `js` file:

```
Added to 3rd line after var app =
.....app.filter('unsafe', function($sce) { return $sce.trustAsHtml; });
```

Linking experiments

By: Marian
Created: 03/10/2022
Last updated:
File: js

For more info, see following issue on gitlab:

<https://git.sfb1102.uni-saarland.de/aanwari/lingoturk-masses/-/issues/97>

If you want to redirect your participants to a second experiment on LingoTurk, take the following steps:

1. Make a copy of the finalized version of the experiment type (this type is going to be for your use only and you will need a new copy of the experiment type for each new instance and Prolific completion code)
2. Obtain the `yourexperimenturl` of the next experiment, which is the first part of the URL that you would insert in Prolific up until `&redirectUrl=` (often ending in `partId=SOMENUMBER`).
3. Search for `self.redirectUrl` in the js file and replace the following line

```
self.redirectUrl = ($("#redirectUrl").length > 0) ?  
$("#redirectUrl").val() : null;
```

with

```
self.redirectUrl = "yourexperimenturl" // redirect to my next experiment
```

4. Publish your experiment on Prolific with the experiment URL being the `yourexperimenturl` of the first experiment (so again without `&redirectURL=` in your URL)
5. In the final experiment of your list of experiments you can
 - a. either link to Prolific or
 - b. have a `yourfinalexperimenturl&redirect=` where you include your Prolific URL in the pre-final experiment.

Copying experiment types (templates)

By: Iza
Created: 23/11/2022
Last updated: 17/5/2024
File: html, js and the web interface

DEPRECATED: DO NOT COPY EXPERIMENTS ON MULTITUDE; CONSULT MANSOOR

Sometimes you want to create a new experiment type that is based on an existing one and then make some changes to the hard-coded parts of the new experiment type (like order of slides, the question in the statistics slide, the language of the interface).

There are 2 ways of making a copy of an experiment type:

- a) the back-end one
- b) the web-interface one.

The web-interface method often resulted in crashing the LingoTurk server, so using the back-end method might be safer. Consult Mansoor in case you have doubts.

How to make a copy of the experiment type following the back-end method:

- 1) Let's assume we have an experiment type called "A" and we want to make a copy of it and this new experiment type "B".
- 2) Go to the LingoTurk web-interface on Multitude (if experiment A is on Multitude), go to Create experiment, find your experiment A and click on the export button (the arrow). This will download the experiment type A as a .zip folder.
- 3) On the web-interface, now create a new experiment type, choose the name (here B). Keep this window open and return to it in step 5).
- 4) Go to the downloaded folder .zip folder of experiment A und unzip it. We want to use the same field names and types as in experiment A. For that go to the unzipped folder of experiment A,
app__models__Questions/experimentAExperiment/fields.json. Open the file in a text editor to see the field names and types this experiment uses.
- 5) Go to the window of the web-interface. Enter the field names and types as you see in the fields.json. After that, push the button "Create experiment type". **Because the LingoTurk server is slow, it can happened, this will raise an error saying "Oops, an error occurred". Don't get discouraged: try again until it works. Sometimes it needs a minute or two after a user clicked the button for experiment creation.**
- 6) The next step is about the HTML file. Use Filezilla (or similar) to access the .html of experiment B, so
/shared/LTCopy/lingoturk/app/views/ExperimentRendering/experimentBExperiment/. The file, in our case, is called experimentBExperiment_rendering.html. Open it.
- 7) In the downloaded folder of experiment A, go to
app__views__ExperimentRendering/experimentAExperiment/ and copy the entire content of the html file into the file that you opened on Filezilla. Save your changes. Basically this means the new created experiment type should have the same contents in the html file as the original experiment type.
- 8) For the js file, some editing is needed aside from copy-pasting. Copy the contents of the js file from the downloaded folder in
public__javascripts__ExperimentRendering/experimentAExperiment/.
- 9) In Filezilla, open the js file of the new experiment at
/shared/LTCopy/lingoturk/public/javascripts/ExperimentRendering/experimentBExperiment/. Paste the copied content into this file. You have to substitute the experiment type name in two places in this file:
 - line 2, var app angular.module("experimentAExperimentApp", ["LingoTurk"]);
→ here change "experimentAExperimentApp" to
"experimentBExperimentApp".

- line 139, experimentType : "experiment**A**Experiment", → change to experiment**B**Experiment
Keep other content of the js file the same as in the original experiment type.
Save the changes.

10) That's it. :)