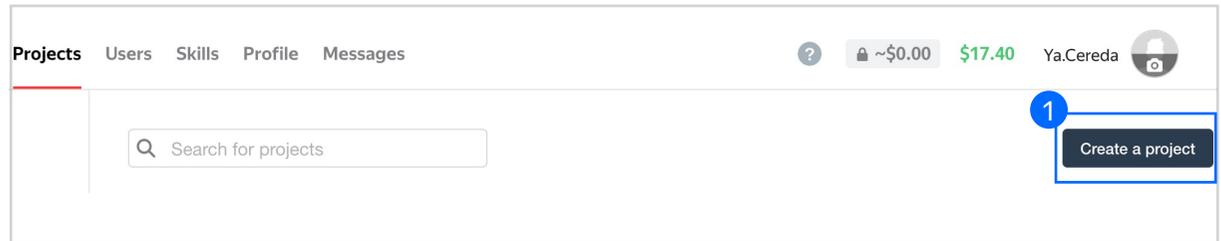




Audio transcription manual

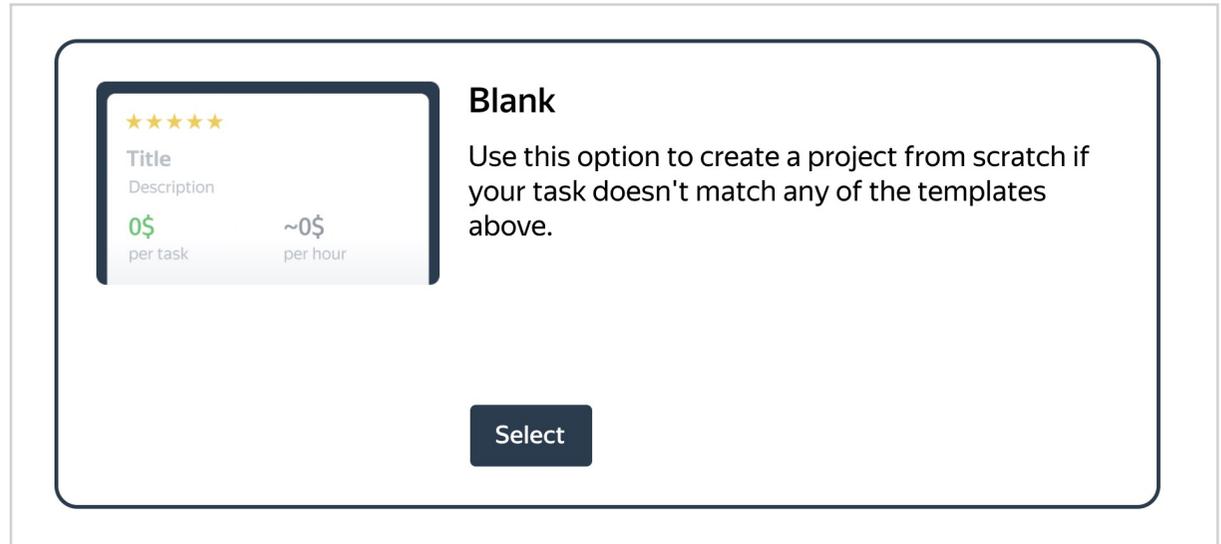
Create a project

1. Click **Create a project**.



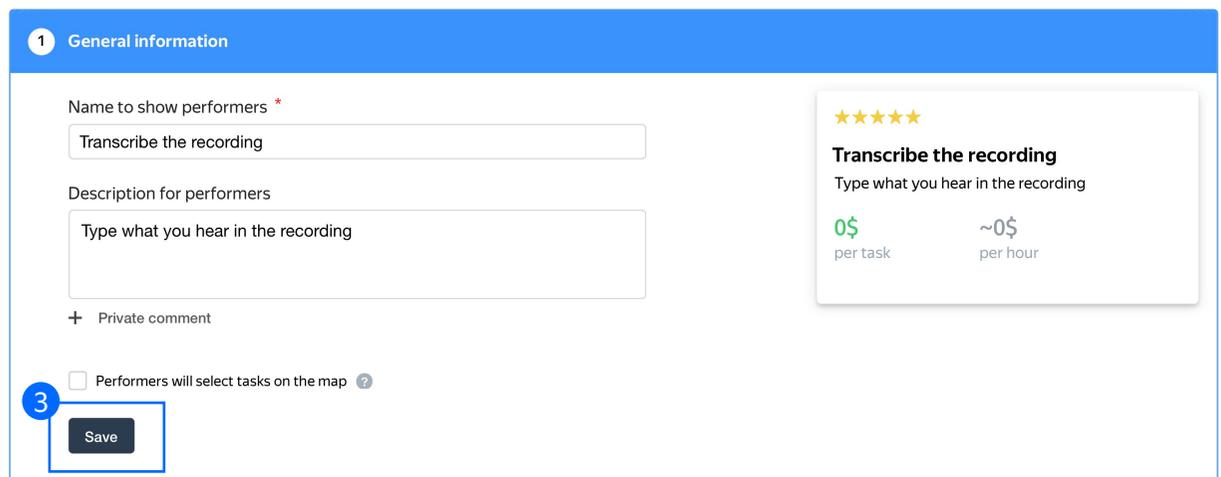
2. Choose the **Blank** template.

Read more about configuring the [task interface](#) in the Requester's Guide.



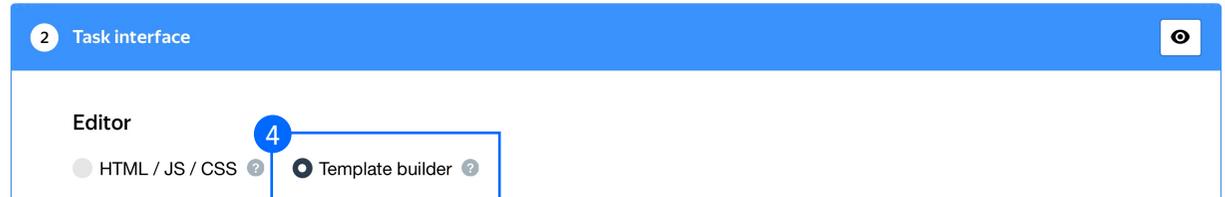
3. Enter a clear project name and description. Click **Save**.

Note: The project will be visible to others.



4. Update the task interface in the **Template Builder** block.

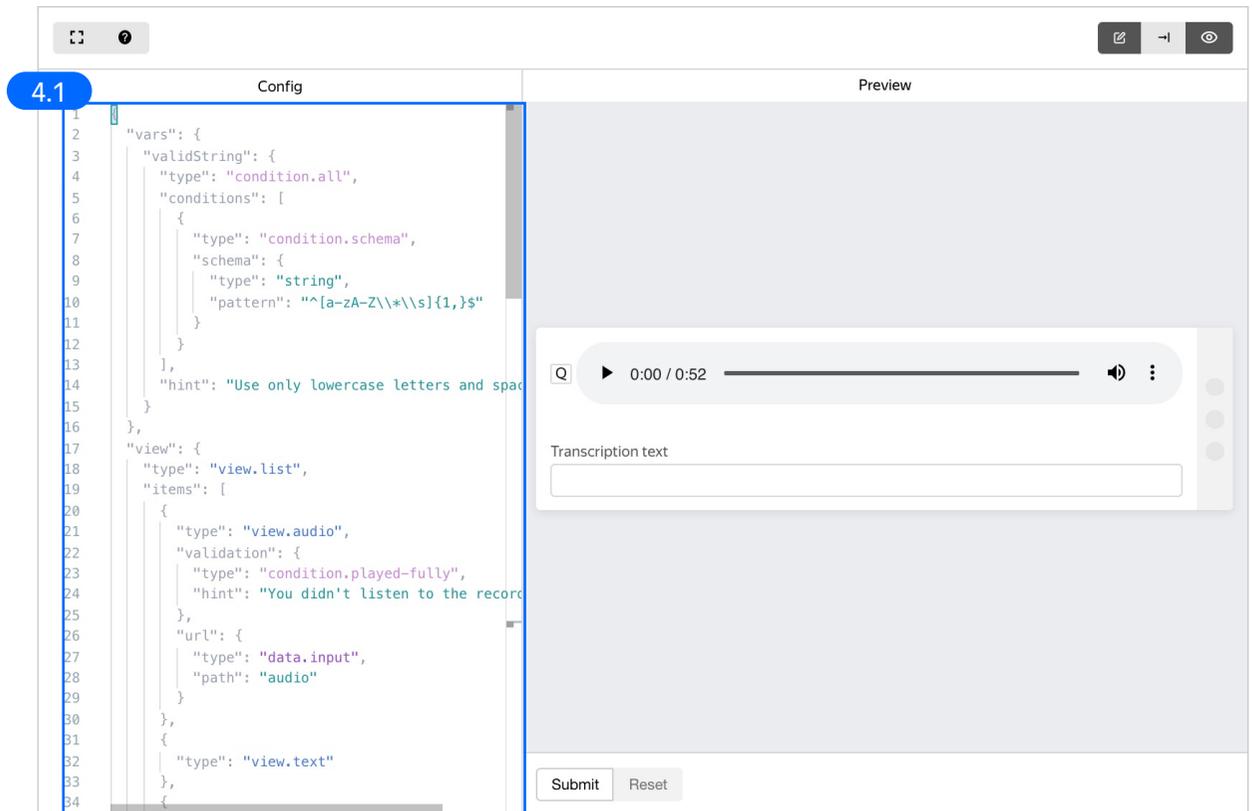
Read more about the [Template builder](#) in the Requester's Guide.



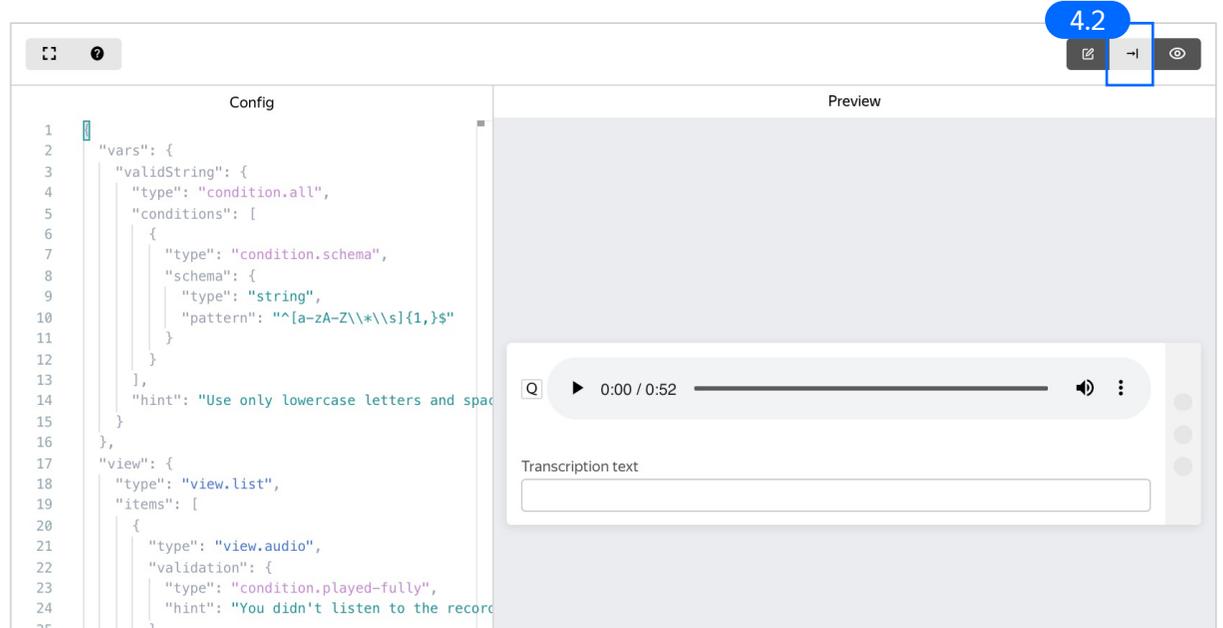
4.1. Delete the existing config and paste the code provided at the end of this manual (in the appendix).

Note: If you play around with the interface in preview mode, you will see that it gives a warning message when a user types any characters that aren't letters. This is an additional way to ensure quality. There's also a plugin that makes all texts lowercase.

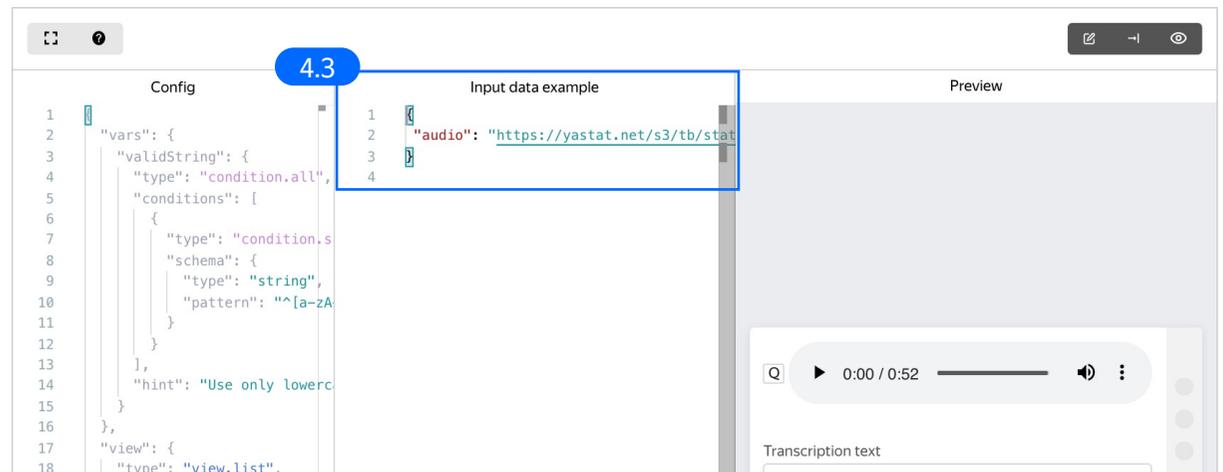
Read more tips on designing [interfaces](#) in our Knowledge Base.



4.2. Click **Input data example** in the right corner.



4.3. Paste the code provided at the end of this manual (in the appendix) into **Input data example**.



4.4. Make sure the specifications look like this.

Note: You can see that there are two output data fields.

result_unformat will contain the actual text typed by the performer.
result will contain the lowercase version.

This happens because of a plugin mentioned in **step 4.1**.

Then click **Save**.

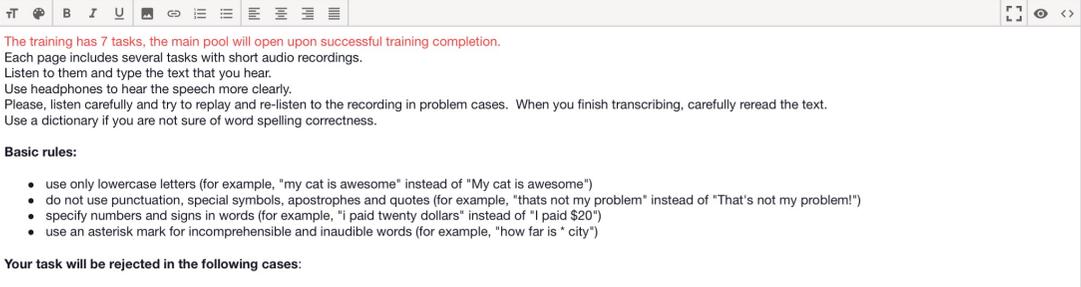
The screenshot displays the 'Data specification' interface, which is divided into two main sections: 'Input data' and 'Output data'. The 'Input data' section contains one field: 'audio (URL)' with a radio button selected. The 'Output data' section contains two fields: 'result_unformat (string)' and 'result (json)', both with radio buttons. Below each section is an 'Add field' button. At the bottom of the interface, there is a 'Show common interface elements' button and a 'Save' button. A blue callout box with the number '4.4' is positioned over the 'Save' button, indicating the current step in the process.

5. Write short and simple instructions. Click **Save**.

Get more tips on designing [instructions](#) in our Knowledge Base.

3 Instructions for performers

When a performer selects a task, they first see the instructions that you wrote. Describe what needs to be done and give examples. You can prepare your instructions in HTML format, then copy and paste them into the editor. Press < > to switch to HTML mode. To learn more, see the [documentation](#).



The training has 7 tasks, the main pool will open upon successful training completion.
Each page includes several tasks with short audio recordings.
Listen to them and type the text that you hear.
Use headphones to hear the speech more clearly.
Please, listen carefully and try to replay and re-listen to the recording in problem cases. When you finish transcribing, carefully reread the text.
Use a dictionary if you are not sure of word spelling correctness.

Basic rules:

- use only lowercase letters (for example, "my cat is awesome" instead of "My cat is awesome")
- do not use punctuation, special symbols, apostrophes and quotes (for example, "that's not my problem!" instead of "That's not my problem!")
- specify numbers and signs in words (for example, "I paid twenty dollars" instead of "I paid \$20")
- use an asterisk mark for incomprehensible and inaudible words (for example, "how far is * city")

Your task will be rejected in the following cases:

5 Save

6. Leave the **Translations** block as default and click **Save**.

4 Translations

i Performers from different countries will understand the task better if the instructions and all descriptions are in their native language. Fill in "Name and description" and "Instructions" for each language that you want performers to see. If you leave these empty, the language is inactive.

Source language
—

Translations

| Language | Name and description for performers | Instructions for performers |
|----------|-------------------------------------|-----------------------------|
| ✓ Source | ✓ | ✓ |

Add translation

6 Save

7. Click **Finish** to save the project.

Edit project

Back to the old interface Cancel **Finish**

- ✓ General information
- ✓ Task interface
- ✓ Instructions for performers
- ✓ Translations

Note. To edit project parameters, click the button in the list of projects or go to **Project actions** → **Edit** on the project page.

Transcribe the recording — active

Project actions ^

- Edit
- Clone
- Archive
- Preview

Statistics for 7 days

| Submitted tasks | Spent | Quality: control tasks | Quality: training tasks | Average submit time | Users | Banned users |
|-----------------|-------|------------------------|-------------------------|---------------------|-------|--------------|
| 0 | 0 \$ | - | - | - | 0 | 0 |

Create an exam pool

1. Click **Add a pool**.

Note: We are adding an exam pool without a training pool because our project does not involve complicated guidelines that need to be learned. Instead, we need to check that a person is able to recognize the speech in a specific language recorded under specific conditions.

This is dataset-specific and can't be learned quickly, so we will select performers who are already able to do it instead of training them.

The screenshot shows the 'Transcribe the recording' project page in an active state. At the top, there's a header with the project name and a 'Project actions' dropdown. Below this is a 'Statistics for 7 days' section with a table of metrics: Submitted tasks (0), Spent (0 \$), Quality: control tasks (-), Quality: training tasks (-), Average submit time (-), Users (0), and Banned users (0). The main content area has tabs for 'Pools', 'Training', 'Statistics', and 'Quality control', with 'Pools' selected. There are buttons for 'Active and closed', 'Archived', and 'Filters', along with a search bar. A blue circle with the number '1' highlights the 'Add a pool' button. Below the buttons, a note states: 'Pools can be archived manually or automatically (automatic archiving applies to pools with no activity for 30 days)'. A table header is visible with columns: Title, Priority, Progress, Status, Started, and To be completed. A message below the table reads: 'To launch a project, you first need to add a pool, set user filters and quality control rules, and upload tasks.' A dropdown menu at the bottom right shows the number '50'.

2. Give the pool any name you find suitable. You are the only one who will see it.

The description can be either public or private. Choose the option you prefer.

2

POOL NAME (VISIBLE ONLY TO YOU) ? Transcribe the recording - exam

Use project description

PUBLIC DESCRIPTION ? Type what you hear in the recording

Add a private description

- 2.2. Select **Exam** as the pool type. This will let you launch these tasks at a price of zero.

Note: You can also set a regular price.

POOL TYPE ? Exam

2.2

—

Exam

Training

Retry

Other

PRICE IN US DOLLARS ?

FEE ?

- 2.3. Set the price per task suite (for example, \$0.03).

2.3

Price per task suite

Each task suite can have one or multiple tasks on the same page. Enter the total price for all tasks in the suite.

PRICE IN US DOLLARS ? 0.03

FEE ? 0.006

2.4. Add filters.

Since the recordings are in English, we will only allow English-speaking performers to do the task.

Also, set up access for both desktop and mobile users.

Performers [Copy settings from...](#)

Filter performers who can access the task.
Toloka has users from different countries, so don't forget to filter by language and region. [Learn more](#)

ADULT CONTENT No

PERFORMER PROFILE

2.4 Languages = English

Performers who passed the language test

AND

CALCULATED DATA

Client = Toloka for mobile

OR = Toloka web version

3. Set up the exam quality calculation.

3.1. Click **Add skill**.

Performers [Copy settings from...](#)

Filter performers who can access the task.
Toloka has users from different countries, so don't forget to filter by language and region. [Learn more](#)

ADULT CONTENT No

Add filter Add skill

3.1

3.2. Fill in the skill title. Add a description if you want to.

Click **Add**.

Add skill

TITLE

audio-transcription_exam

DESCRIPTION

Public? No

Cancel **Add**

3.2

3.3. Scroll down to **Quality control**.

Quality control

Add rules to get more accurate responses.
All rules work independently.

NON-AUTOMATIC ACCEPTANCE ? No

REVIEW PERIOD IN DAYS

CAPTCHA FREQUENCY ?

+ Add Quality Control Rule

3.4. Add a **Control tasks** rule with the following settings. We will have 10 tasks in the exam pool, so the quality will be calculated when the whole exam has been passed.

Note: We will then use this parameter as an entry filter for the main pool.

CONTROL TASKS ?

3.4 Recent control task responses to use

If =

then

4. Set an overlap.

Note: The more complicated the dataset, the bigger the overlap should be, so that we can select a reasonable number of performers.

Overlap

Specify how many performers you want to complete each task in the pool.

4

OVERLAP ? 250

DYNAMIC OVERLAP ? Off

5. Select the speed/quality balance if you would like to, but since we are assessing the people in this pool, this is not required.

Top % Online Time

5

Specify the percentage of top-rated active users who can access tasks in the pool.

18700 Speed All 90% 80% 70% 60% 50% 40% 30% 20% 10% 18700 Quality

All users selected
The task is available to 18700 active users.

6. Leave the other parameters with their default settings, and click **Save**.

Parameters

TIME PER TASK SUITE IN SECONDS ? 600

POOL CLOSING DATE ? 2022-09-13

KEEP TASK ORDER ? No

WAITING TIME FOR THE POOL TO CLOSE IN SECONDS ? 0

POOL PRIORITY WITHIN THE PROJECT ? 0

Cancel **Save**

7. Upload exam tasks to the pool.
You can download the file with the tasks [here](#).

License: CC BY 4.0

[Origin](#)

[Citation](#)

Valentini-Botinhao, Cassia. (2017). Noisy speech database for training speech enhancement algorithms and TTS models, 2016 [sound]. University of Edinburgh. School of Informatics. Centre for Speech Technology Research (CSTR). <https://doi.org/10.7488/ds/2117>.

7.1

Transcribe the recording - exam — closed

Statistics Download results Edit

Download the sample file, add your task data, and upload the file to the pool. The sample file uses TSV format, which is the same as CSV but using tab as the separator. Make sure you choose UTF-8 encoding when saving the file. [Learn more in the guide](#).

[Template for general tasks.tsv](#)
[Template for control tasks.tsv](#)
[Template for training tasks.tsv](#)

Upload

0 task pages 0 training tasks
0 tasks 0 control tasks

0% Completed 0

Note: You can create the exam tasks locally and upload them in a file like this one, or upload regular tasks and create controls via the UI.

Read [more about this](#) in the Requesters Guide.

We have 10 tasks, so we will make it two pages.

File upload settings ?

Tasks per page

By empty row Set manually Smart mixing

Tasks per page 5

Sample file for uploading tasks Close Upload

7.2

8. Run the pool.

Note: We recommend running the exam pool together with the main pool or just before. Otherwise the performers will pass the exam pool, but find no regular tasks available, and might not return to the project later.

8

Transcribe the recording - exam — closed

Statistics Download results Edit

Download the sample file, add your task data, and upload the file to the pool. The sample file uses TSV format, which is the same as CSV but using tab as the separator. Make sure you choose UTF-8 encoding when saving the file. [Learn more in the guide.](#)

- Template for general tasks.tsv
- Template for control tasks.tsv
- Template for training tasks.tsv

Upload Files Delete Preview

| | |
|--------------|------------------|
| 2 task pages | 0 training tasks |
| 0 tasks | 10 control tasks |

0% Completed 0

0 500

Create the main pool

1. Click Add a pool.

Transcribe the recording — active

Project actions

Statistics for 7 days

| | | | | | | |
|-----------------|-------|------------------------|-------------------------|---------------------|-------|--------------|
| Submitted tasks | Spent | Quality: control tasks | Quality: training tasks | Average submit time | Users | Banned users |
| 0 | 0 \$ | - | - | - | 0 | 0 |

Pools Training Statistics Quality control

Active and closed Archived Filters Search

1

Add a pool

Pools can be archived manually or automatically (automatic archiving applies to pools with no activity for 30 days).

| Title | Priority | Progress | Status | Started | To be completed |
|-------|----------|----------|--------|---------|-----------------|
|-------|----------|----------|--------|---------|-----------------|

To launch a project, you first need to add a pool, set user filters and quality control rules, and upload tasks.

50

2. Give the pool any name you find suitable. You are the only one who will see it.

The description can be either public or private. Choose the option you prefer.

POOL NAME (VISIBLE ONLY TO YOU) ? Transcribe the recording

Use project description

PUBLIC DESCRIPTION ? Type what you hear in the recording

Add a private description

3. Specify [pool parameters](#):

3.1. Select pool type.

POOL TYPE ? Other

—
Exam
Training
Retry
✓ Other

PRICE IN US DOLLARS ? FEE ?

3.2. Set the price per task suite (for example, \$0.03).

Price per task suite

Each task suite can have one or multiple tasks on the same page. Enter the total price for all tasks in the suite.

PRICE IN US DOLLARS ? 0.03 FEE ? 0.006

+ Dynamic pricing

3.3. [Filter](#) the performers who can access the task. Add a language filter and platform filters like you did for the exam.

3.3

PERFORMER PROFILE

Languages = English

Performers who passed the language test

AND

CALCULATED DATA

Client = Toloka for mobile

OR = Toloka web version

3.4. Add filtering by exam quality. Select **My skills** and find a skill where you recorded exam quality. Use it as a filter.

3.4

SKILLS

audio-transcription_exam ≥ 80

3.5. Create a new skill for recording quality in the main pool.

Click **Add skill**.

3.6. Fill in the skill title. Add a description if you want to.

Click **Add**.

3.7. Use this skill as a filter as well.

Note: The threshold is low. This is intentional, because standard control task rules only allow us to compare answers to control tasks by complete match. Audio transcription might be tricky and allow for insignificant mistakes (i.e. “we” instead of “he” etc.), which will be calculated as incorrect answers. So if we’re working through the UI, we can only use this filter to get rid of completely random answers.

However, there are ways to calculate quality in a fairer way via Toloka-Kit – see our solution for more details.

Performers Copy settings from...

Filter performers who can access the task. Toloka has users from different countries, so don't forget to filter by language and region. [Learn more](#)

ADULT CONTENT ? Yes

Add filter ▼ 3.5 **Add skill**

Add skill

TITLE

audio-transcription_main ×

DESCRIPTION

Public? No ?

3.6 **Cancel** **Add**

SKILLS

audio-transcription_main × \geq 30 🗑️ +

OR $=$ Missing 🗑️

3.7

3.8. Set up [Quality](#) control calculation:
Click + **Add Quality Control Rule**

Quality control
Add rules to get more accurate responses.
All rules work independently.

NON-AUTOMATIC ACCEPTANCE No

REVIEW PERIOD IN DAYS 14

CAPTCHA FREQUENCY

3.8

+ Add Quality Control Rule

3.9. Click **Control tasks** and fill in the options.

Note: The skill will start calculating after we get at least 3 responses, and will use a history of the last 10 responses. Based on this history, we will record a quality parameter and turn it into the skill created earlier.

3.9

Recent control task responses to use 10

If number of responses \geq 3

then assign skill from the field audio_transcrip

correct responses (%)

3.10. Set up the overlap. This is the number of users who will complete the same task. For example, 3. We will aggregate the results after the pool is completed.

Overlap

Specify how many performers you want to complete each task in the pool.

3.10

OVERLAP ? 3

DYNAMIC OVERLAP ? Off

3.11. Optionally, specify the percentage of top-rated performers in the [Speed / quality ratio](#).

Note: This is not really necessary though, since we only grant access to performers who have already proved their competence in the exam.

Top% Online Time

Specify the percentage of top-rated active users who can access tasks in the pool.

3.11

Speed All 90% 80% 70% 60% 50% 40% 30% 20% 10% Quality

60% top-rated performers were selected.
Tasks are not available for active users.

3.12. Time given to complete a task suite (for example, 900 seconds)

Parameters

3.12

TIME PER TASK SUITE IN SECONDS ? 900

KEEP TASK ORDER ? No

POOL CLOSING DATE ? 2022-09-13

WAITING TIME FOR THE POOL TO CLOSE IN SECONDS ? 0

POOL PRIORITY WITHIN THE PROJECT ? 0

4. Click **Save** to save Pool parameters.

Parameters

TIME PER TASK SUITE IN SECONDS ? 900

POOL CLOSING DATE ? 2022-09-13

KEEP TASK ORDER ? No

WAITING TIME FOR THE POOL TO CLOSE IN SECONDS ? 0

POOL PRIORITY WITHIN THE PROJECT ? 0

Cancel Save

Prepare and upload a file with tasks

1. Prepare a TSV file with tasks as shown in our [example](#).
License: CC BY 4.0

[Origin](#)

[Citation](#)

Valentini-Botinhao, Cassia. (2017). Noisy speech database for training speech enhancement algorithms and TTS models, 2016 [sound]. University of Edinburgh. School of Informatics. Centre for Speech Technology Research (CSTR). <https://doi.org/10.7488/ds/2117>.

2. [Upload pool tasks](#) from this file.

Transcribe the recording — closed

Statistics Download results Edit

Download the sample file, add your task data, and upload the file to the pool.
The sample file uses TSV format, which is the same as CSV but using tab as the separator.
Make sure you choose UTF-8 encoding when saving the file. [Learn more in the guide.](#)

[Template for general tasks.tsv](#)
[Template for control tasks.tsv](#)
[Template for training tasks.tsv](#)

Upload

0 task pages 0 training tasks

0 tasks 0 control tasks

0% Completed 0

2.1. Select [Smart mixing](#) in **File upload settings** and specify the number of tasks per page. For example: 4 main tasks and 1 control task. Click **Upload** once again.

Note: If you changed the name of the input field, change it in the file as well.

File upload settings ?

Tasks per page

By empty row Set manually **Smart mixing**

Main tasks: 4

Training tasks: 0

Control tasks: 1

[Show advanced settings](#)

Sample file for uploading tasks **2.1** Close Upload

3. Create control tasks. Click **Edit**.

Note: Control tasks are tasks that already contain the correct response. They are used for checking the quality of responses from performers. The performer's response is compared to the response you provided. If they match, it means the performer answered correctly. By default Toloka measures full match of responses, but it is possible to calculate partial matches with external tools. See our Toloka-Kit solution for an example of such.

Transcribe the recording — closed

Statistics Download results Edit

Download the sample file, add your task data, and upload the file to the pool. The sample file uses TSV format, which is the same as CSV but using tab as the separator. Make sure you choose UTF-8 encoding when saving the file. [Learn more in the guide.](#)

- [Template for general tasks.tsv](#)
- [Template for control tasks.tsv](#)
- [Template for training tasks.tsv](#)

Upload Files Delete **3** Edit

| | |
|--------------|------------------|
| 0 task pages | 0 training tasks |
| 100 tasks | 0 control tasks |

0 % Completed 0

3.1. Click **Create control tasks**.

Edit tasks

Use main tasks as a starting point to create control tasks or training tasks. Control tasks are for checking the quality of responses from performers. They contain correct responses to compare with actual responses. Training tasks are for teaching performers how to complete tasks. They contain correct responses and hints. [Learn more](#)

Main 100 Control tasks 0 Training tasks 0

3.1 Create control tasks Create training tasks Download

| ID | Overlap | Responses from performers | Last updated |
|-------------|---------|---------------------------|-----------------------|
| ...3a2d0799 | 3 | 0 | 09/13/2021 2:32:43 PM |
| ...3a2d07a3 | 3 | 0 | 09/13/2021 2:32:43 PM |

3.2. Select **result** as the field to be used. This means that the performer's response will first be changed to lowercase and then compared to the true answer.

Enter correct responses for your control tasks. Listen to the audio, type in the text, click **Save**, and go to the next one. Repeat this until you have ~10 control tasks.

We recommend that you skip audio that is unclear even for you, to make this check fairer for good performers. In small pools, control tasks should account for 10–20% of all tasks.

To learn more about [creating control tasks](#), go to the Requester's Guide.

Create control task

1. Enter correct responses

Select the fields to use

| Field | Value |
|--|-----------------------------|
| <input checked="" type="checkbox"/> result | It benefits the whole famil |
| <input type="checkbox"/> result_unformat | It benefits the whole famil |

3.2

0:01 / 0:01

Transcription text

It benefits the whole family

Save and go to next

3.3. To check the number of control tasks, go to the **Pool** page.

Note: You can prepare control tasks not only in the interface. You can collect them outside Toloka and upload a [task set with golden answers](#).

[Origin](#)

License: CC BY 4.0

[Citation](#)

Valentini-Botinhao, Cassia. (2017). Noisy speech database for training speech enhancement algorithms and TTS models, 2016 [sound]. University of Edinburgh. School of Informatics. Centre for Speech Technology Research (CSTR). <https://doi.org/10.7488/ds/2117>.

3.3

| | |
|----------------|------------------|
| ~69 task pages | 0 training tasks |
| 90 tasks | 10 control tasks |

4. Preview the pool.

Note: Remember that the tasks will be completed by actual Tolokers. Double check that everything is correct with your project configuration.

5. Start the pool.

5  Transcribe the recording — closed

Statistics Download results Edit

Download the sample file, add your task data, and upload the file to the pool.
The sample file uses TSV format, which is the same as CSV but using tab as the separator.
Make sure you choose UTF-8 encoding when saving the file. [Learn more in the guide.](#)

- Template for general tasks.tsv
- Template for control tasks.tsv
- Template for training tasks.tsv

Upload Files Delete Edit Preview

| | |
|----------------|------------------|
| ~69 task pages | 0 training tasks |
| 90 tasks | 10 control tasks |

0 % Completed 0

0 -69



Receiving responses

1. Wait until the pool is completed. Refresh the pool page to check progress.

2. Click **Download results**.

3. Make sure to uncheck **Separate assignments with empty row** and **Download the results**.

Aggregate the results by using **Text Aggregation**. You can find [the code](#) in the Toloka Crowd-Kit.

Transcribe the recording — closed

Download the sample file, add your task data, and upload the file to the pool. The sample file uses TSV format, which is the same as CSV but using tab as the separator. Make sure you choose UTF-8 encoding when saving the file. [Learn more in the guide.](#)

- Template for general tasks.tsv
- Template for control tasks.tsv
- Template for training tasks.tsv

64 task pages, 0 training tasks, 73 tasks, 27 control tasks

100 % Completed 64, accepted 64

View assignments

Download results

Status: Active, Submitted, Accepted, Rejected, Skipped, Expired

Columns: URL, Task suite ID, task ID, Performer ID, assignment ID, start time, submit time, status, reject time, skip time, accept time, expire time, price

Download data for the period

Separate assignments with empty row

Exclude assignments by banned users

Close Download results

Appendix

Interface code Step 4.1.

```
{
  "vars": {
    "validString": {
      "type": "condition.all",
      "conditions": [
        {
          "type": "condition.schema",
          "schema": {
            "type": "string",
            "pattern": "[a-zA-Z\\s]{1,5}"
          }
        }
      ],
      "hint": "Use only lowercase letters and spaces"
    }
  },
  "view": {
    "type": "view.list",
    "items": [
      {
        "type": "view.audio",
        "validation": {
          "type": "condition.played-fully",
          "hint": "You didn't listen to the recording"
        },
        "url": {
          "type": "data.input",
          "path": "audio"
        }
      },
      {
        "type": "view.text"
      },
      {
        "type": "field.text",
        "label": "Transcription text",
        "data": {
          "type": "data.output",
          "path": "result_unformat"
        },
        "validation": {
          "sref": "vars.validString"
        }
      }
    ],
    "plugins": [
      {
        "type": "plugin.trigger",
        "onChangeOf": {
          "type": "data.output",
          "path": "result_unformat"
        },
        "action": {
          "type": "action.set",
          "data": {
            "type": "data.output",
            "path": "result"
          },
          "payload": {
            "type": "helper.text-transform",
            "transformation": "lowercase",
            "data": {
              "type": "helper.replace",
              "data": {
                "type": "data.output",
                "path": "result_unformat"
              },
              "find": "/(\\s{2})/",
              "replace": ""
            }
          }
        }
      },
      {
        "type": "plugin.toloka",
        "layout": {
          "kind": "scroll",
          "taskWidth": 900
        }
      },
      {
        "type": "plugin.hotkeys",
        "q": {
          "type": "action.play-pause",
          "view": {
            "sref": "view.items.0"
          }
        }
      }
    ]
  }
}
```

Appendix

Input data example

Step 4.2.

```
{  
  "audio": "https://yastat.net/s3/tb/static/file-  
examples/audio/medium.mp3"  
}
```