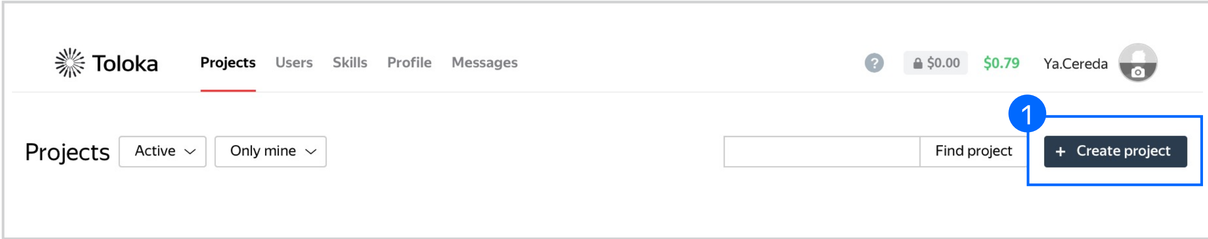
Toloka

Text classification manual

Create a project

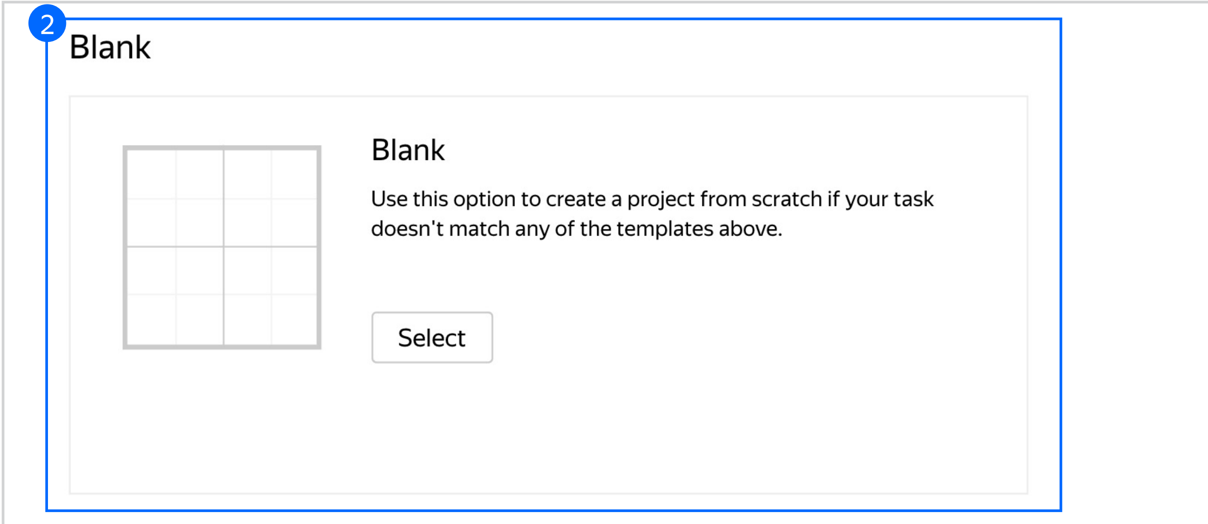
1. Click **Create project.**



2. Choose the **Blank** template.

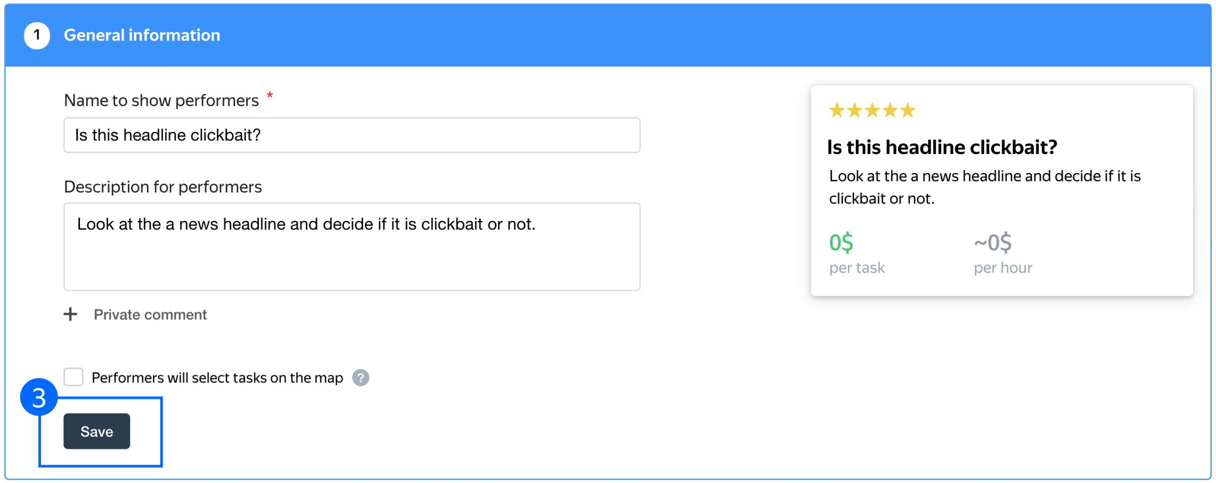
   You can also use the **Text Classification** preset and update it according to your task specifics.

   Read more about configuring the task interface in the Requester's Guide.



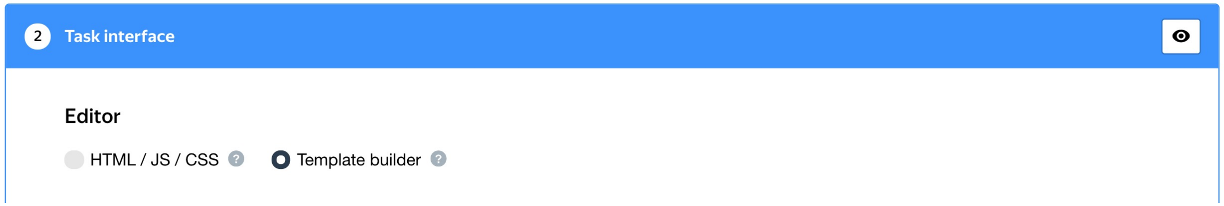
3. Enter a clear project name and description. Click **Save.**

   Note: The project name and description will be visible to the performers.



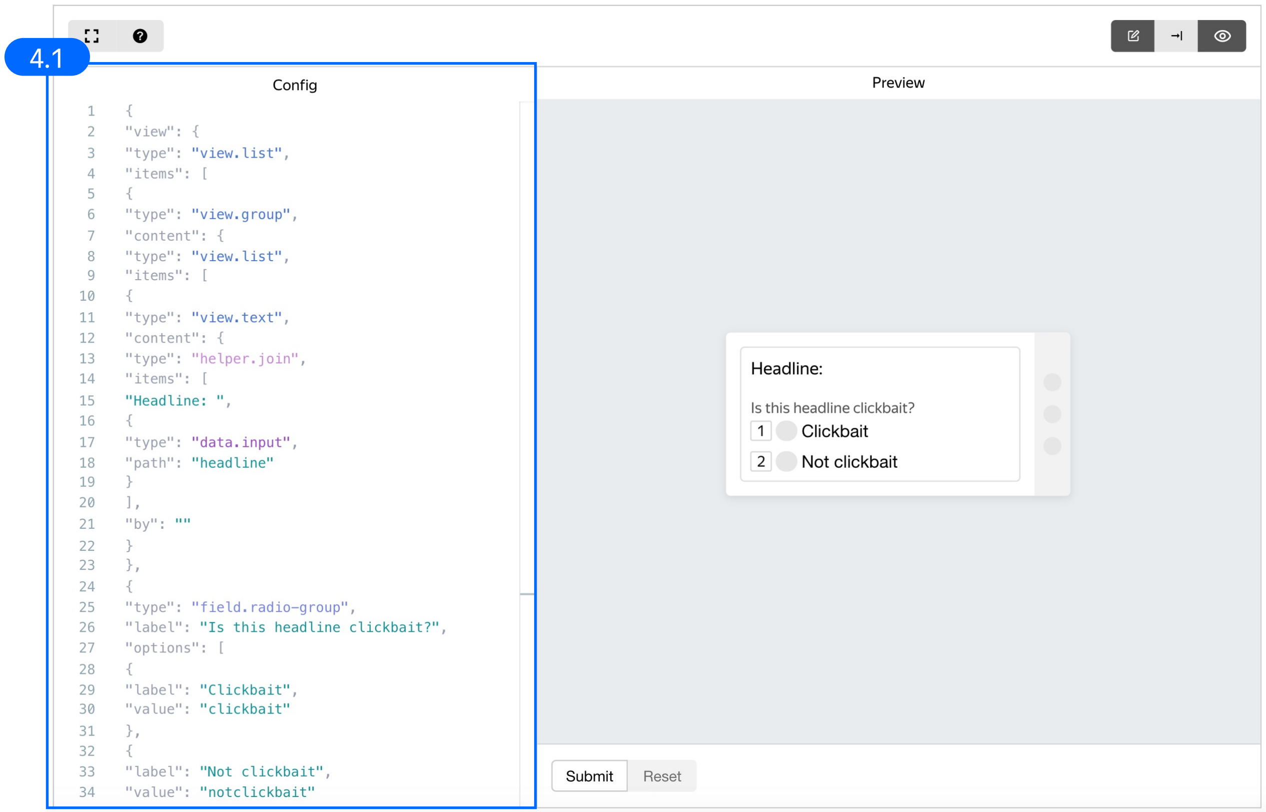
4. Update the task interface in the **Template Builder** block.

   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).

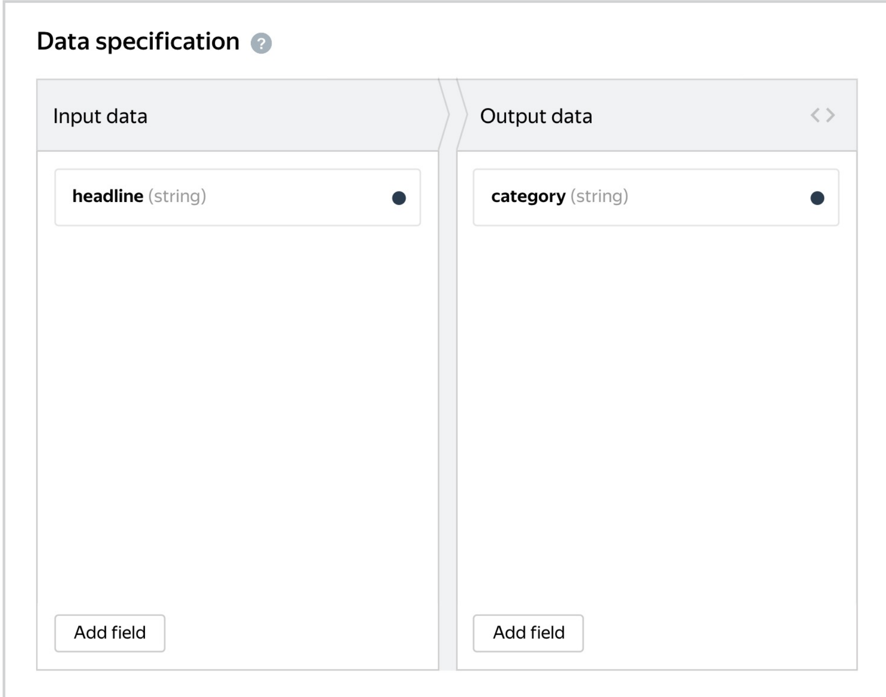
   Check the [Interfaces section](#) of our Knowledge Base for more tips on interface design.



4.2. Make sure the specifications look like this:

Note: Specifications are a description of input data that will be used in a project and the output data that will be collected from the performers.

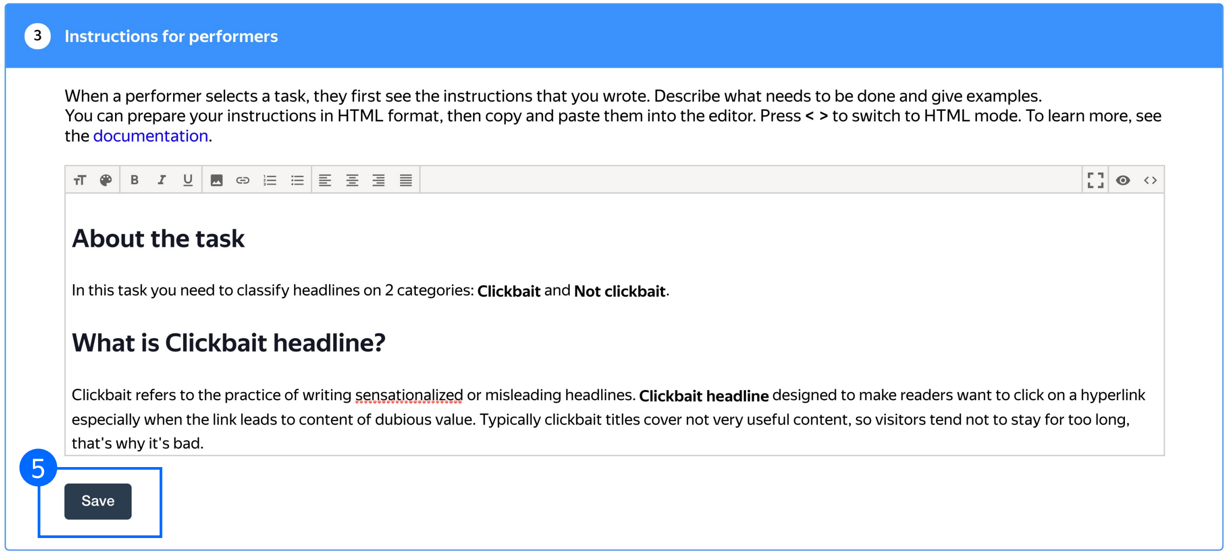
Read more about [input and output data specifications](#) in the Requester's Guide.



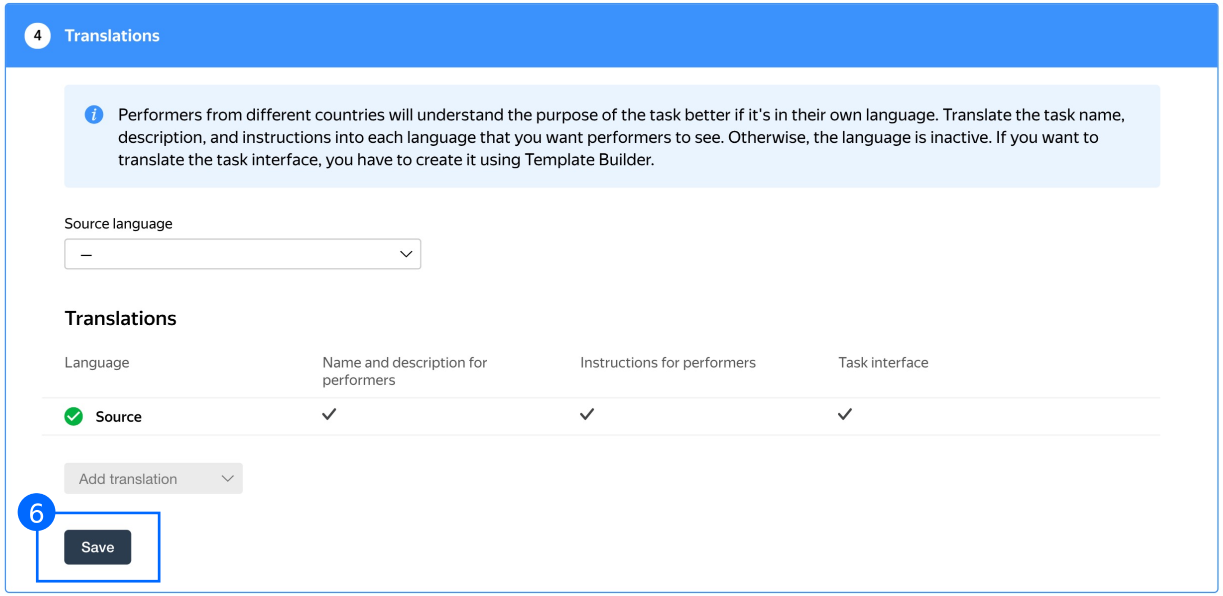
5. Write comprehensive instructions. Make sure to add examples of what should be regarded as clickbait and define the possible grey areas. There is a validation rule in the template. It checks whether an answer has been chosen.

   Click **Save.**

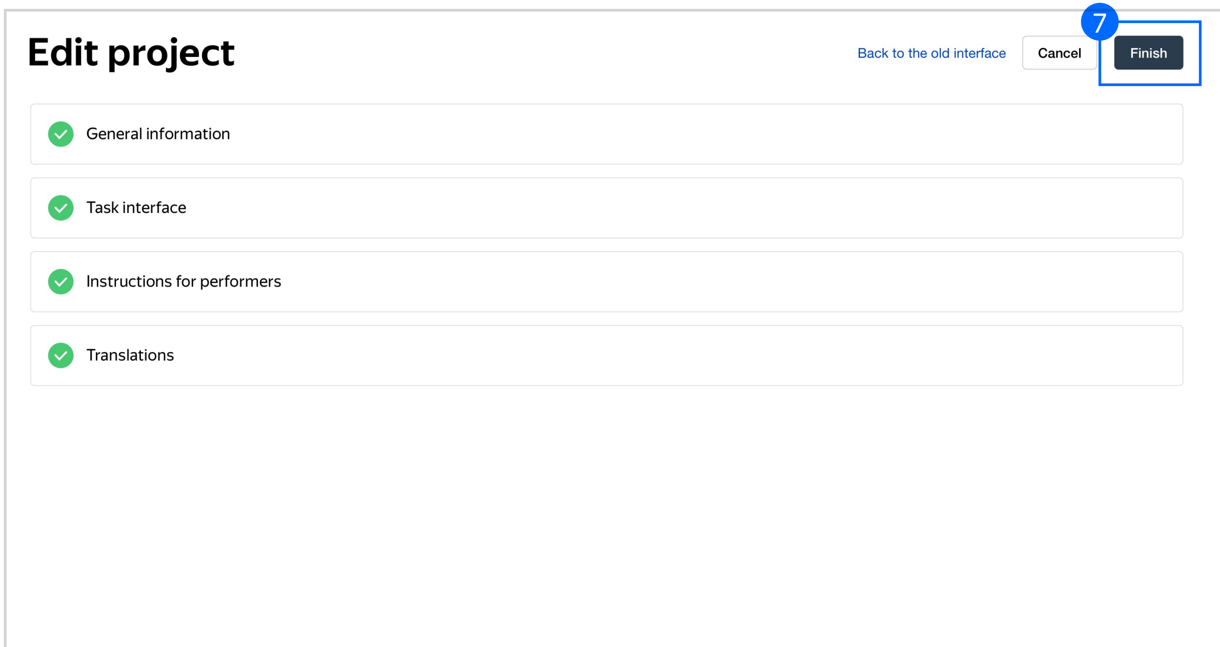
   Get more tips on designing [instructions](#) in our Knowledge Base.



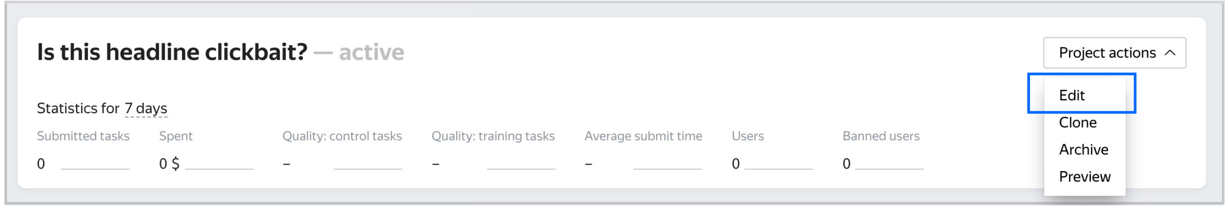
6. Leave the **Translations block** as default and click **Save.**



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



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

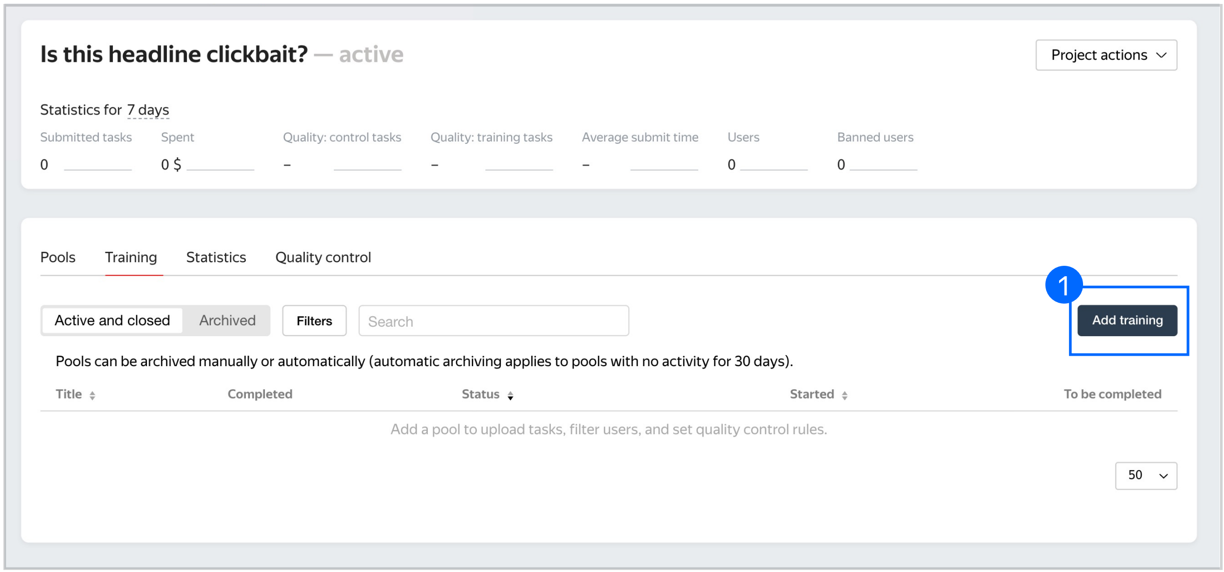


Create a training pool

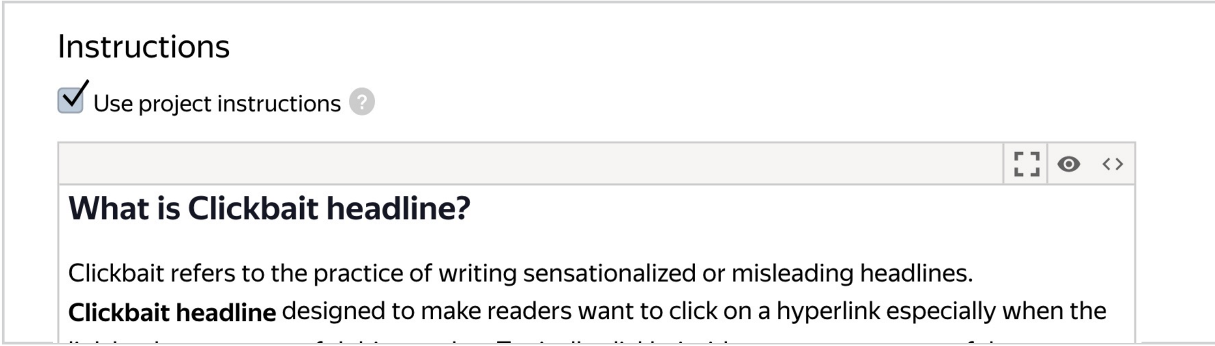
1. Click on the **Training** tab and then click **Add a pool.**

   Training is an essential part of almost every crowdsourcing project. It allows you to select performers who have really mastered the task, and thus improve quality. Training is also a great tool for scaling your task because you can run it any time you need new performers.

   Read more about [selecting performers]() in our Knowledge Base.



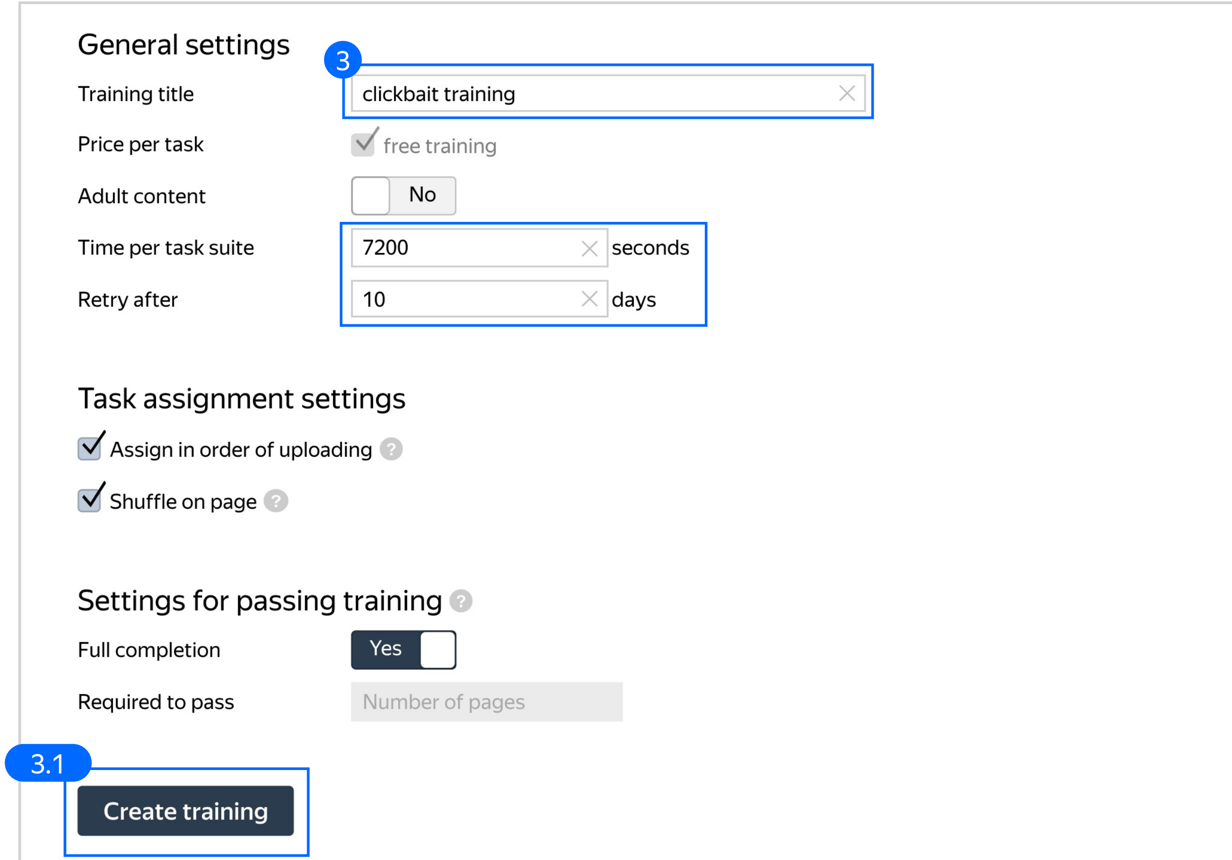
2. Use the existing project instructions.



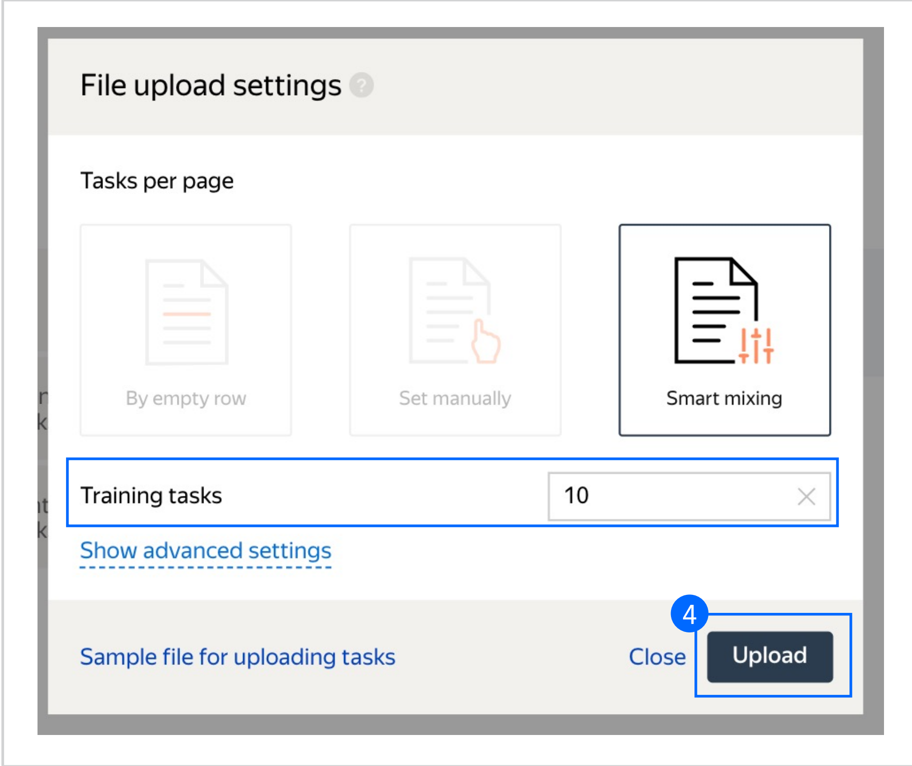
3. Specify the training pool settings:

3.1. Click **Save training.**

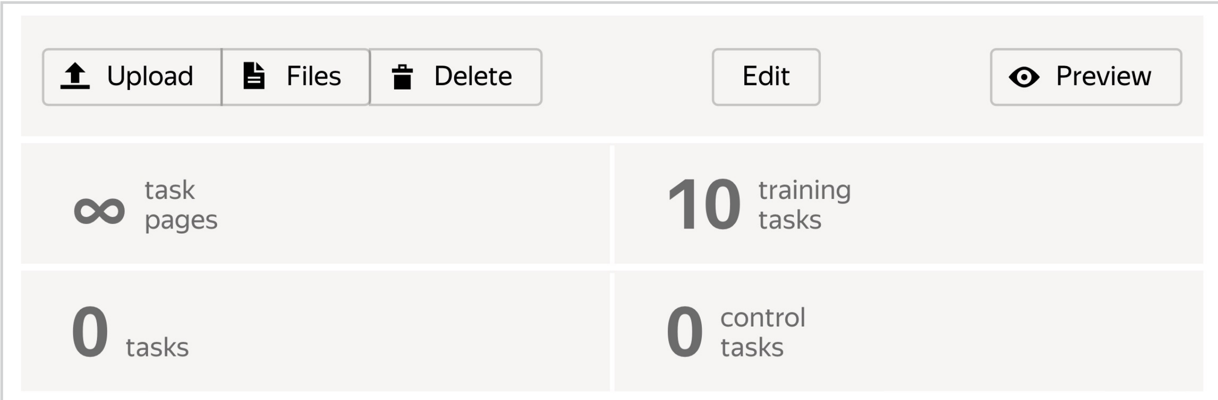
Read more about training pools in our Requester's Guide.



4. Upload training tasks to the pool. You can download a sample file with tasks here:
   Origin
   License: MIT

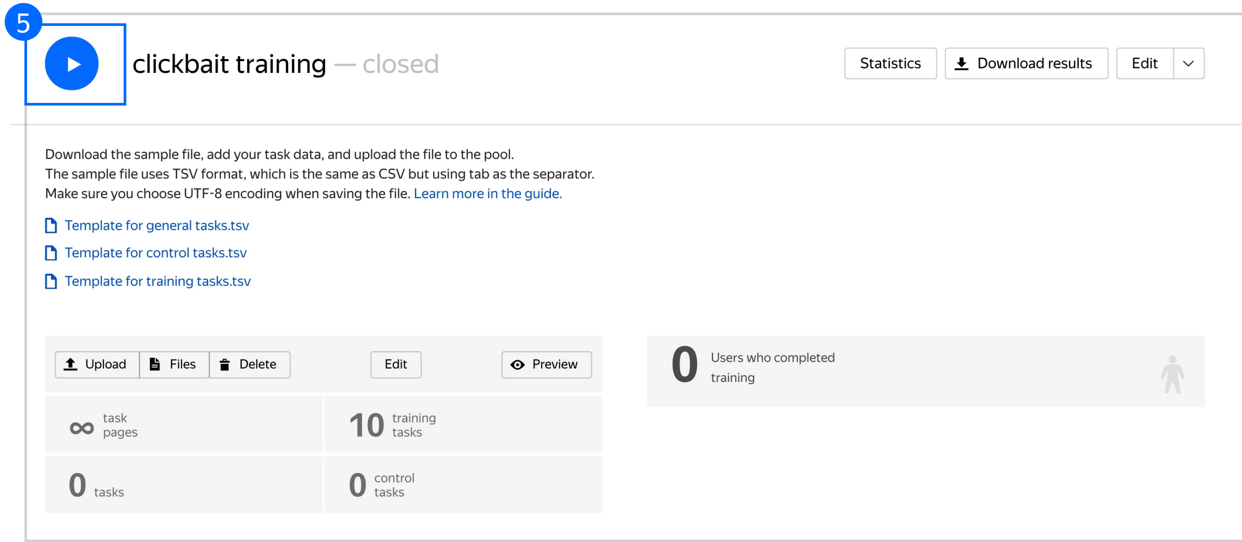


If the upload is successful, you will see the updated task counter.



5. Run the training pool

We recommend opening the training pool along with the main pool. Otherwise Tolokers will spend their time on training but get no access to real tasks, which is frustrating. Also, do not forget to close the training pools when there are no main tasks available anymore.

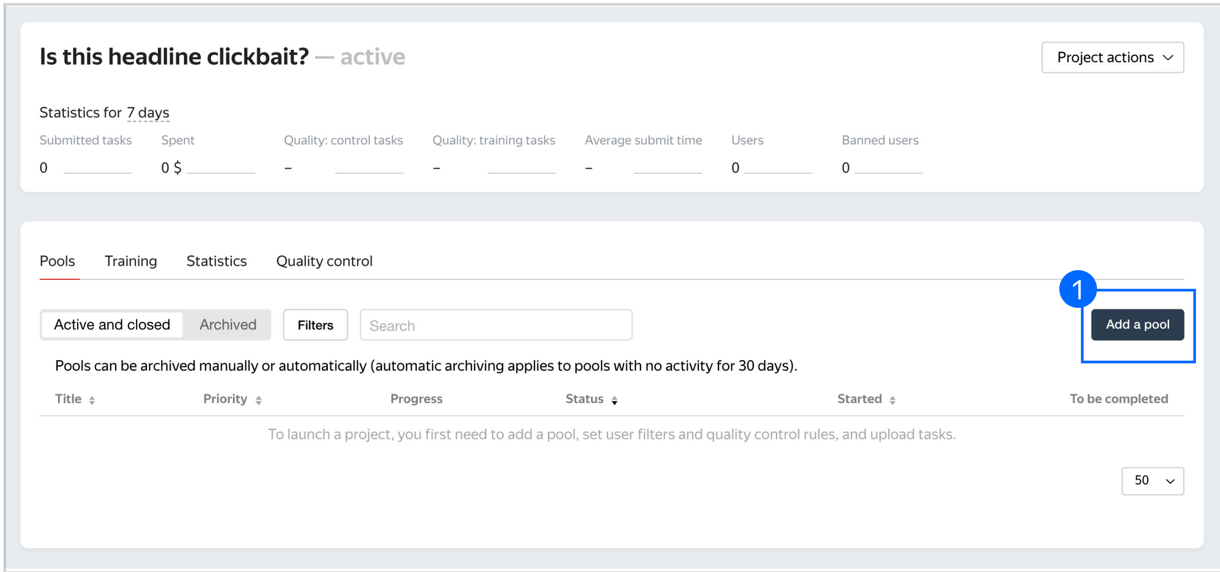


Create the main pool

1. Click **Add a pool.**

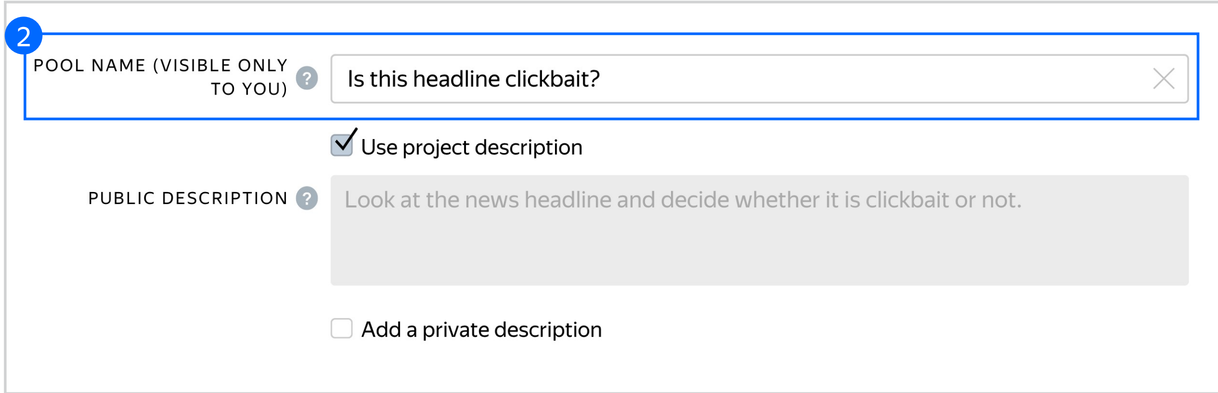
   A pool is a set of paid tasks grouped into task pages. These tasks are sent out for completion at the same time.

   Note: All tasks within a pool have the same settings (price, quality control, etc.)



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

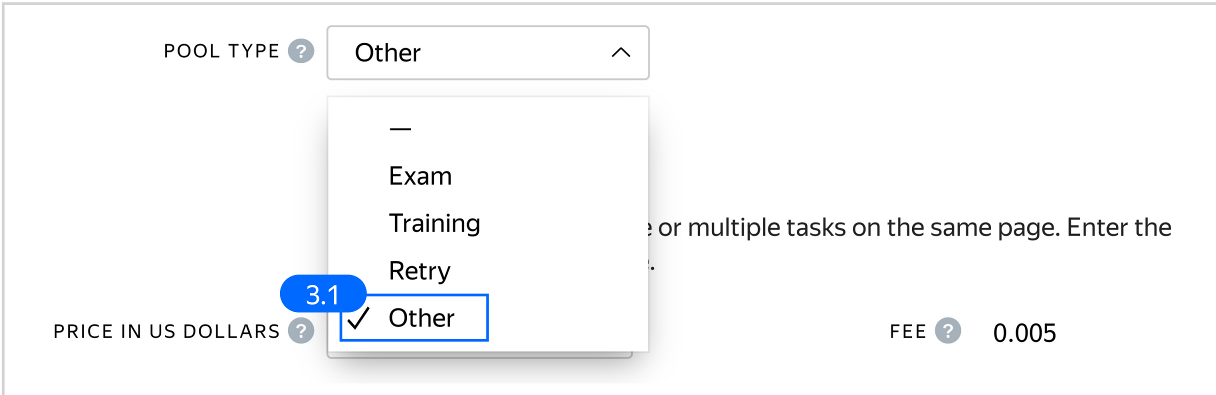
   The description can be either public or private. Choose the option you prefer.



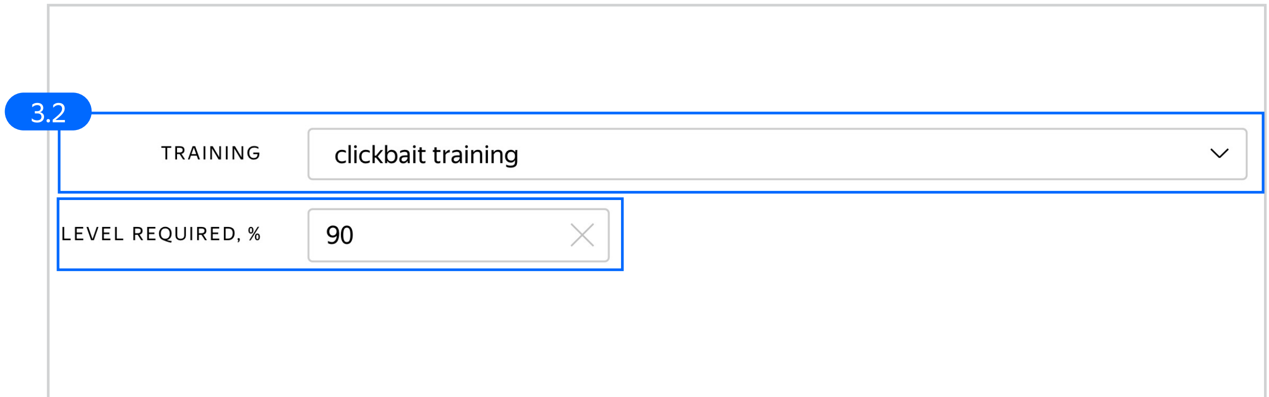
3. Specify pool parameters:

3.1. Select pool type. Choose **Other.**

Read more about pool types in the Requester's Guide.



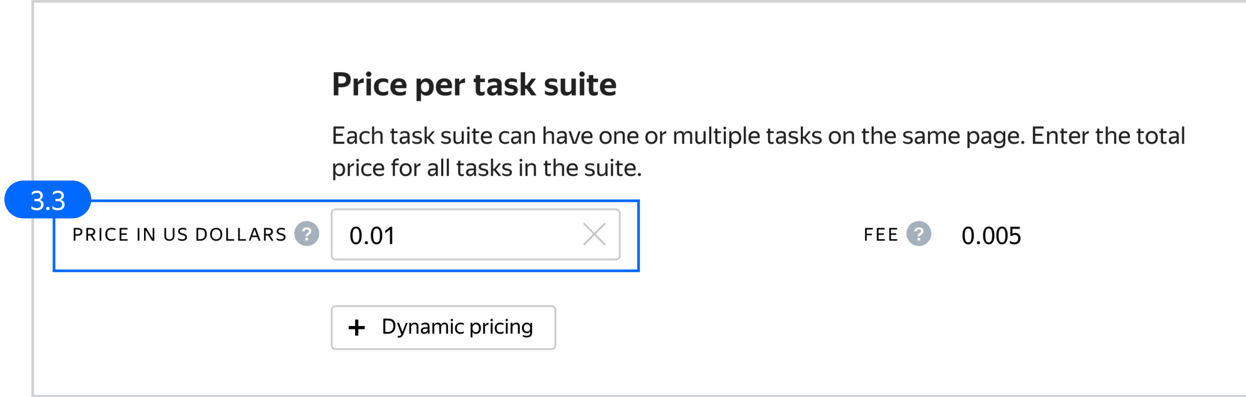
3.2. Attach the training you created earlier and select the accuracy level that is required to reach the main pool. This means that Tolokers who got less than 90% accuracy will not see this pool.



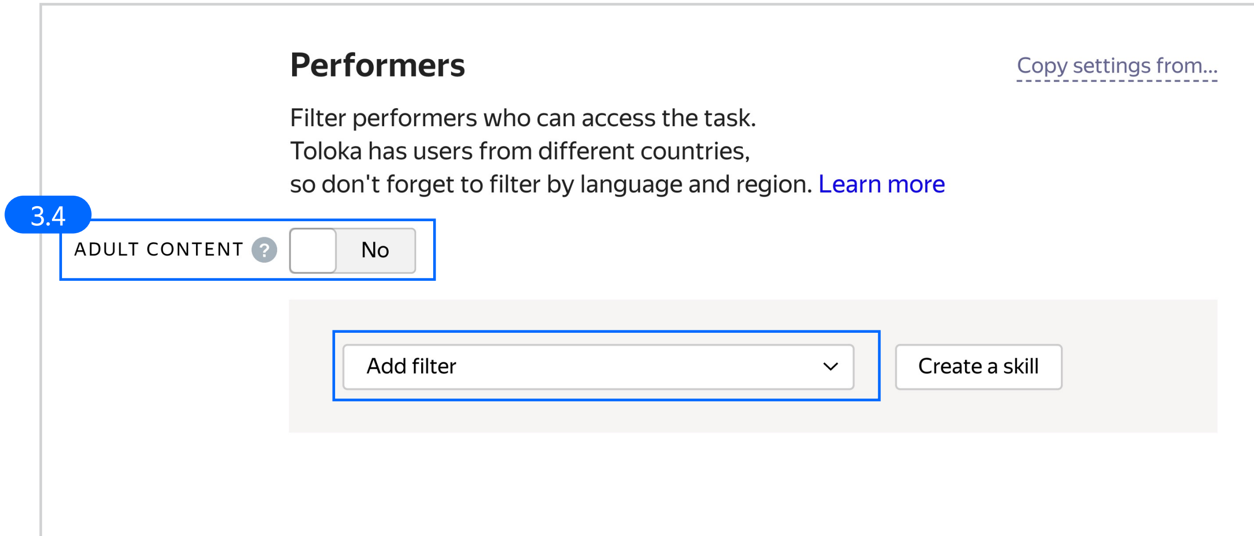
3.3. Set the price per task suite (for example, $0.01).

Text classification tasks are normally paid as basic tasks because these tasks do not take much time.

Read more about pricing principles in our Knowledge Base.

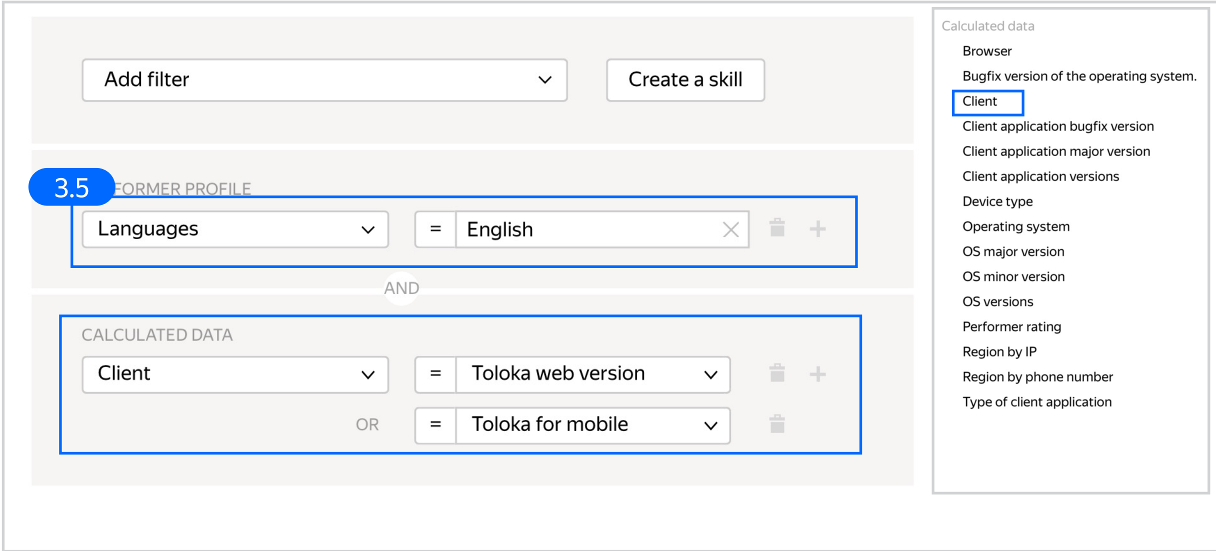


3.4. Filter performers who can access the task. Choose “**No**” in the **Adult content** block. Click **Add filter** to choose the **Languages** and **Client** options in the list.



3.5. Choose **Languages = English** as your first filter. This way, performers who speak English will be invited to complete this task. Then choose **Toloka web version** and **Toloka for mobile** clients.

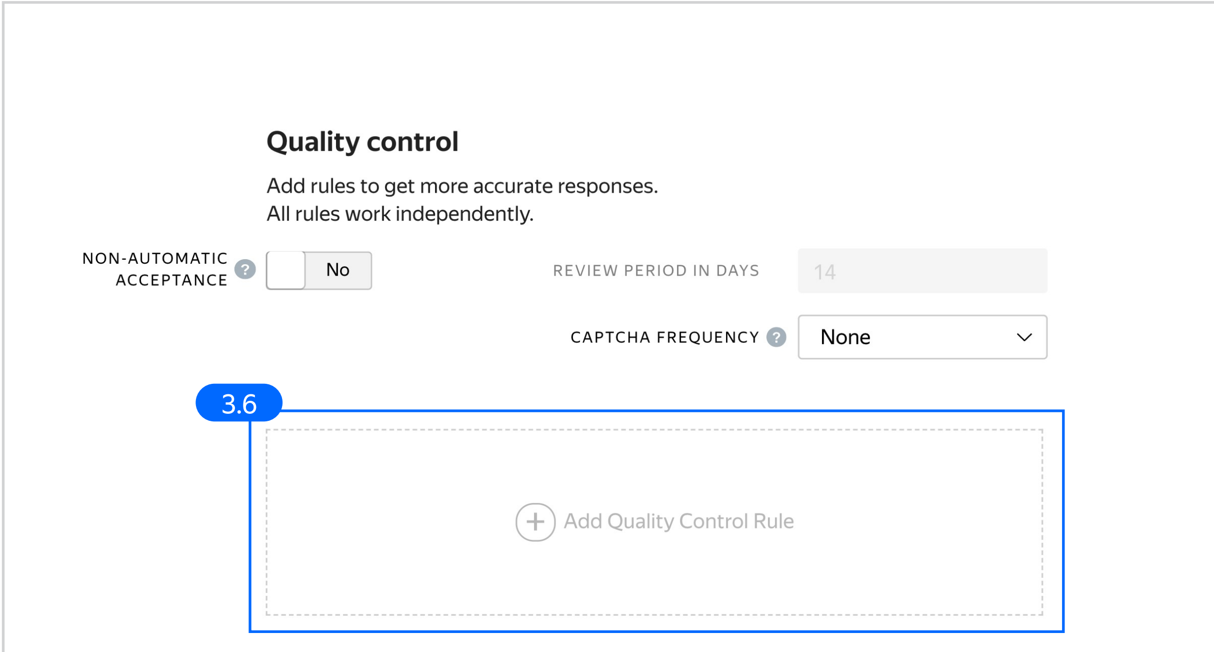
These filters will make it possible for performers to complete your task on their computers or mobile devices.



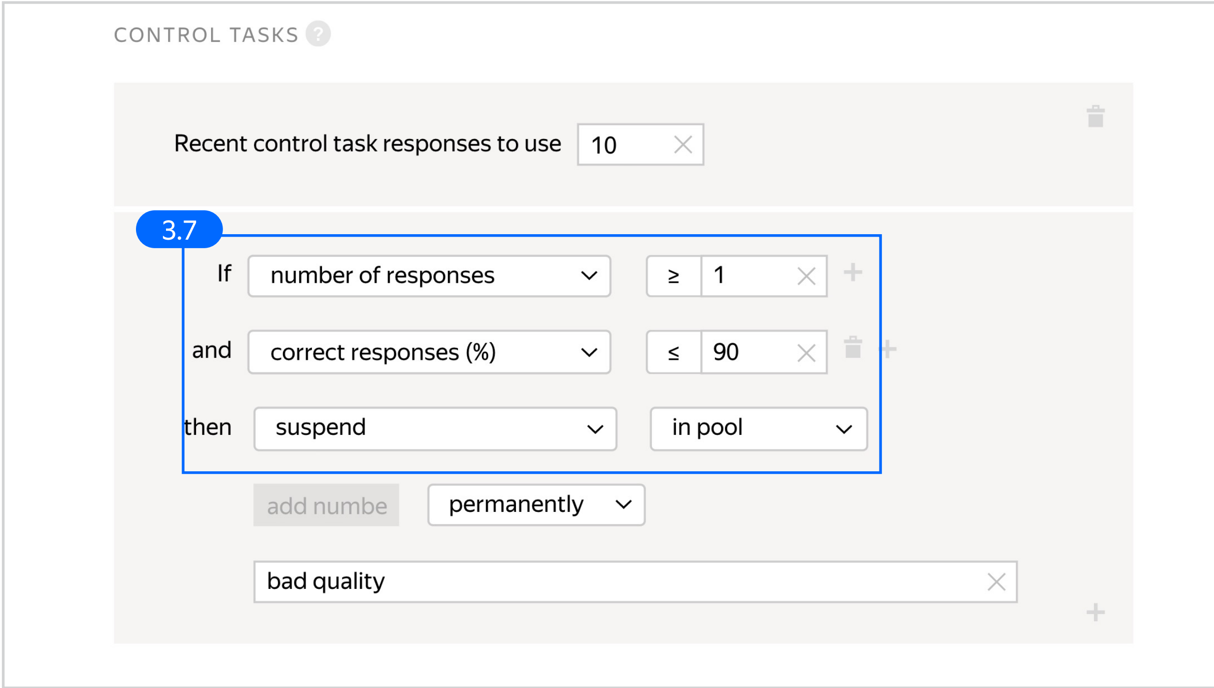
3.6. Set up Quality control. Ban performers who give incorrect responses to control tasks. Click **+ Add Quality Control Rule**

Since tasks such as these have an answer that can be used as ground truth, we can use standard quality control rules like golden sets.

Read more about quality control principles in our Knowledge Base or check out control tasks settings in the Requester's Guide.

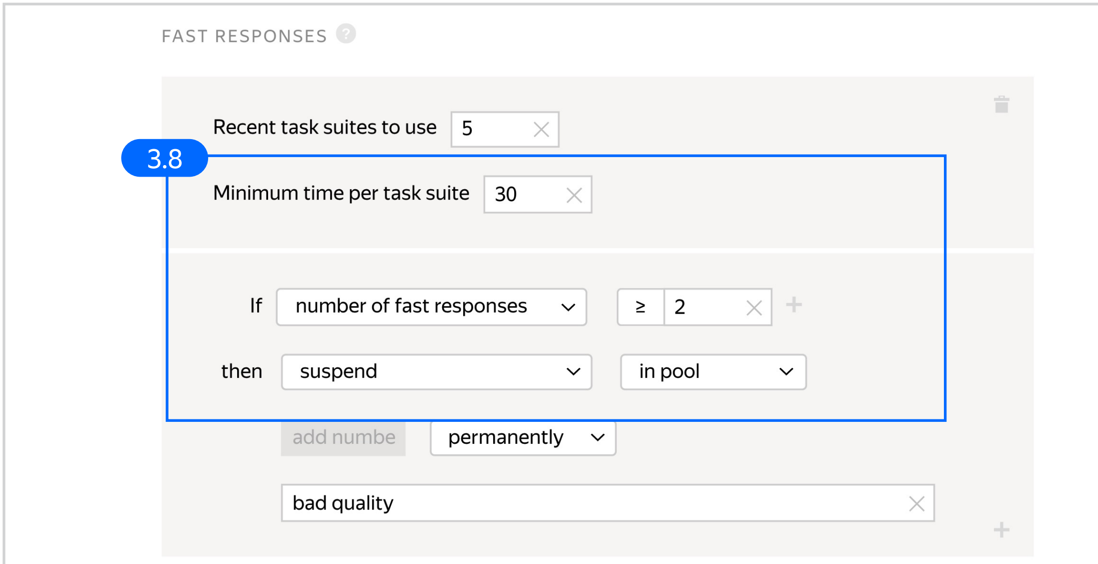


3.7. Click **Control tasks**. Set the number of responses and the percentage of correct responses.



3.8. Set up the up the [Fast responses rule](#).

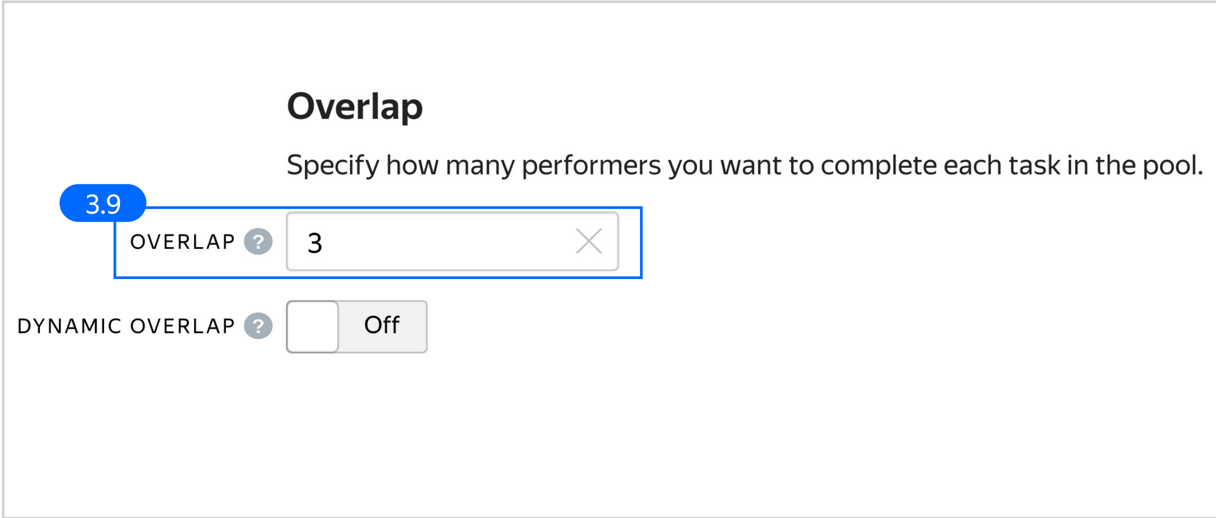
This rule allows you to ban performers who submit tasks at a suspiciously high speed.



3.9. Overlap. This is the number of users who will complete the same task.

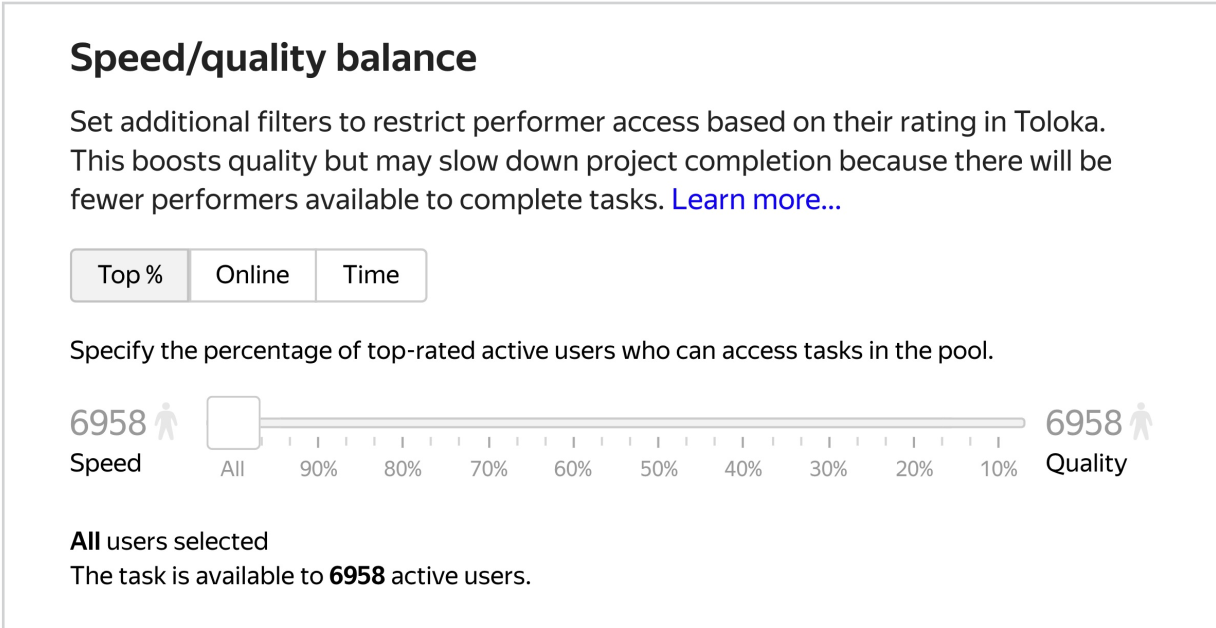
Set an overlap of 3 to get a more confident final label.

To understand how this rule works, go to the Requester's Guide.



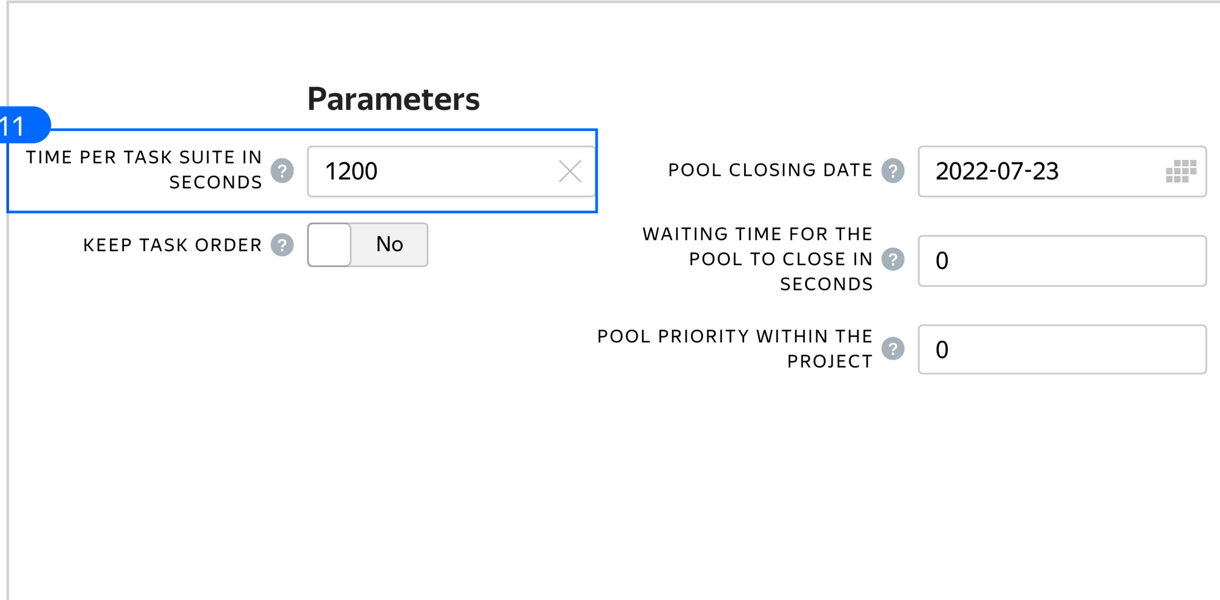
3.10. Optionally, specify the percentage of top-rated performers in the Speed / quality balance.

Note: This can slow down pool completion.

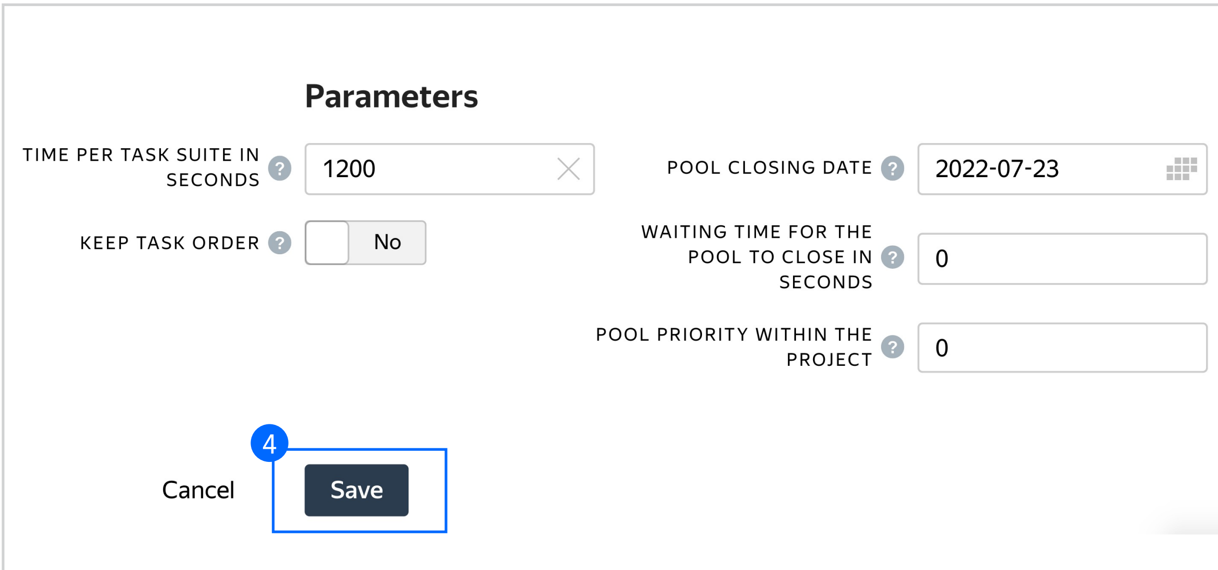


3.11. Specify the time given to complete a task suite (for example, 1200 seconds).

To understand how much time it should take to complete a task suite, try doing it yourself.



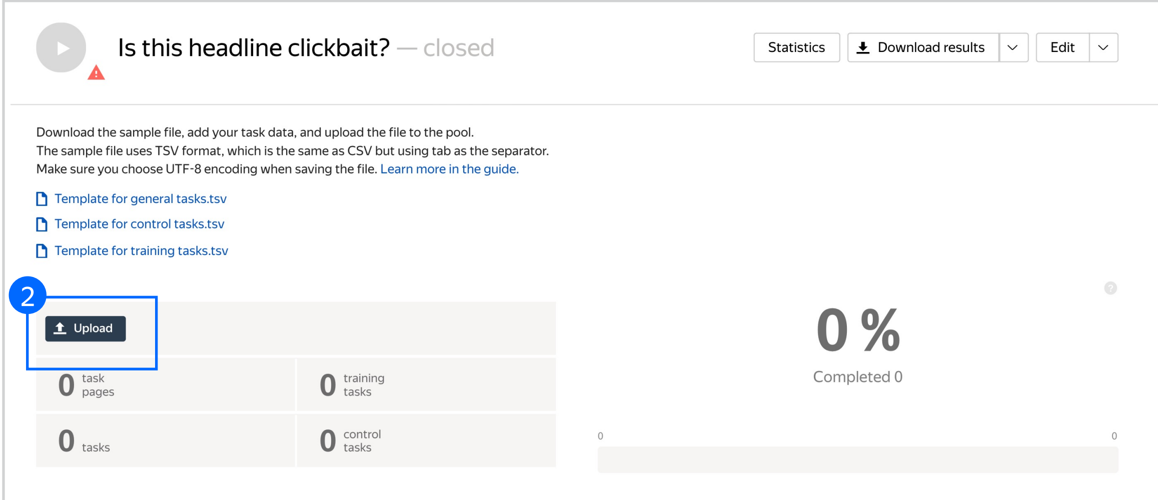
4. Click **Save** to save Pool parameters.



Prepare and upload a file with tasks

1. Prepare a TSV file with tasks as shown in our example.
   Origin
   License: MIT

2. Upload pool tasks from this file.



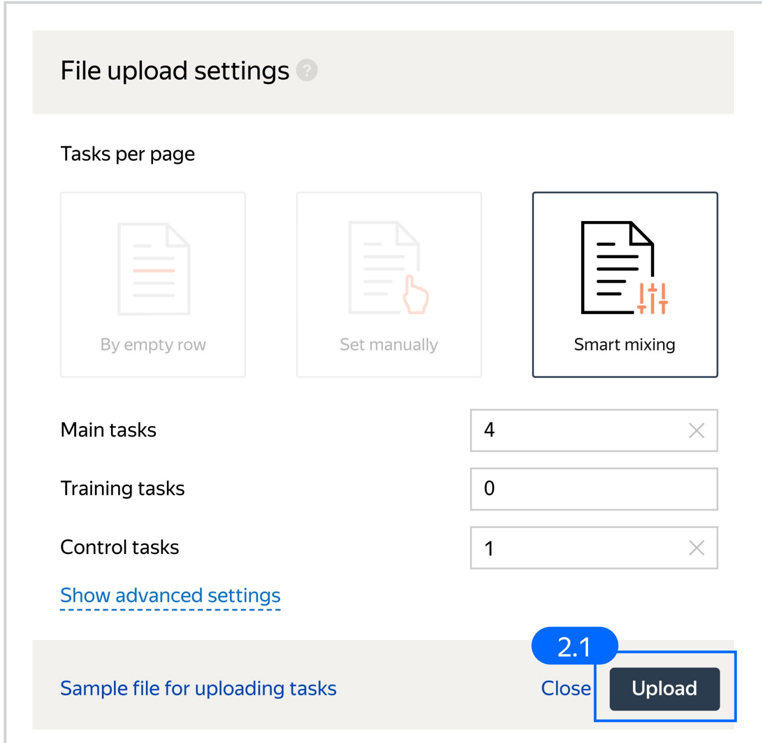
2.1. Select Smart mixing in **File upload settings** and specify the number of tasks of each type per page.

We recommend putting as many tasks on one page as a performer can complete in 1 to 5 minutes. That way, performers are less likely to get tired, and they won't lose a significant amount of data if a technical issue occurs.

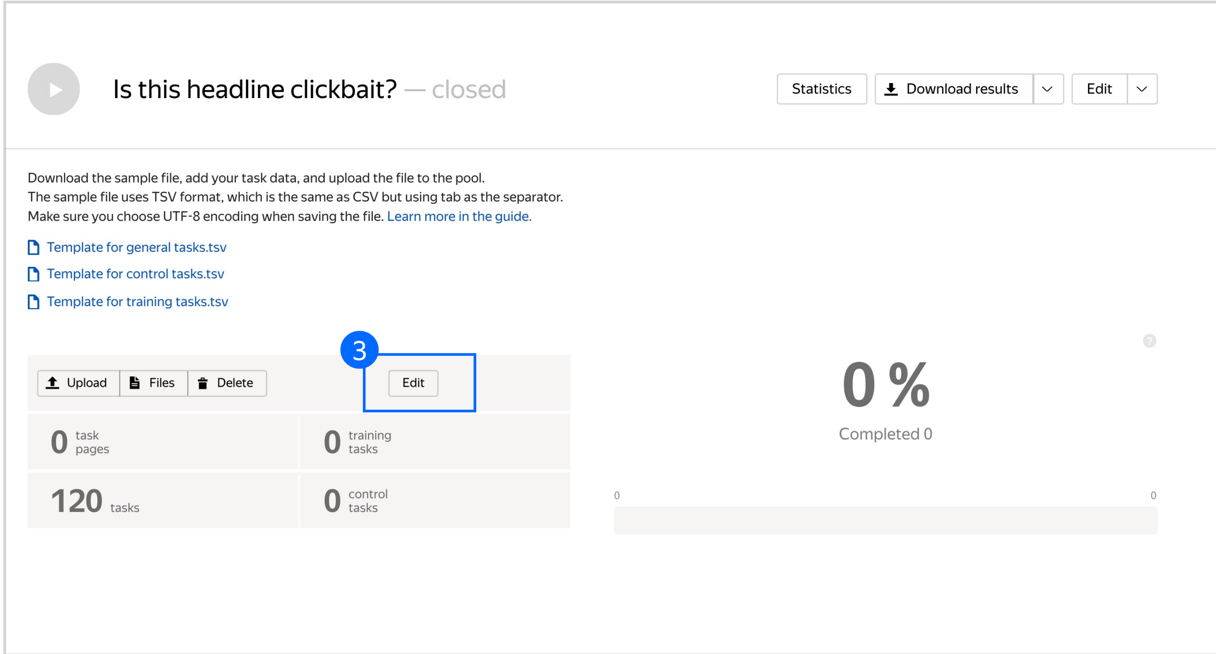
To learn more about grouping tasks into suites, read the Requester's Guide.

Click **Upload** again.

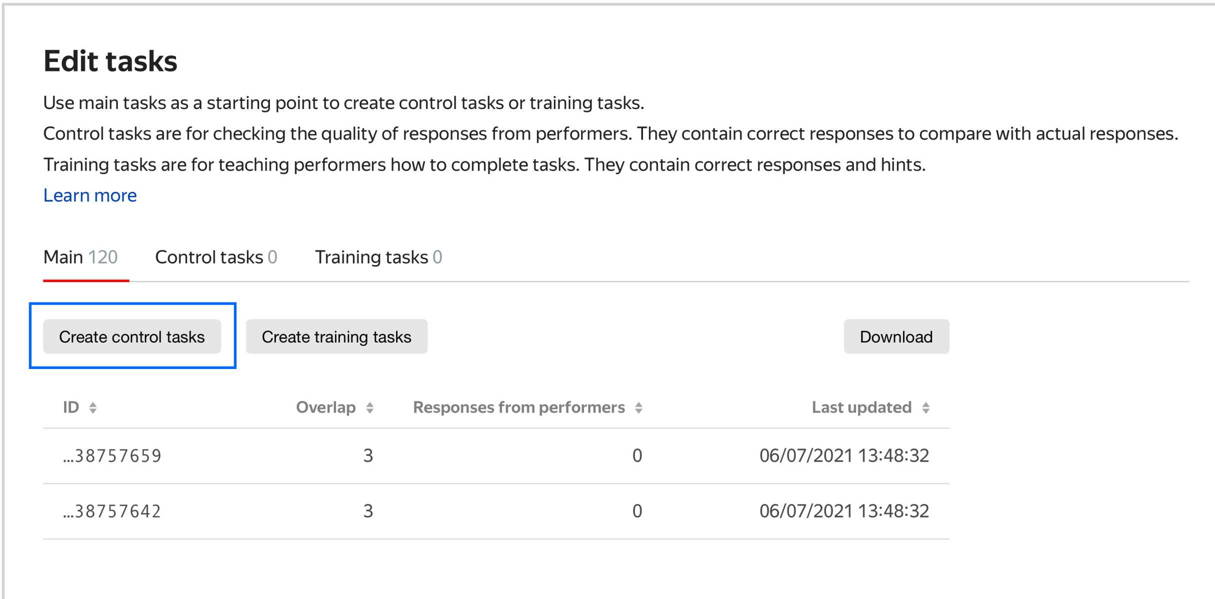
Note: If you changed the name of the input field, change it in the file as well.



3. [Create control tasks](#).
Click **Edit** → **Create control tasks.**



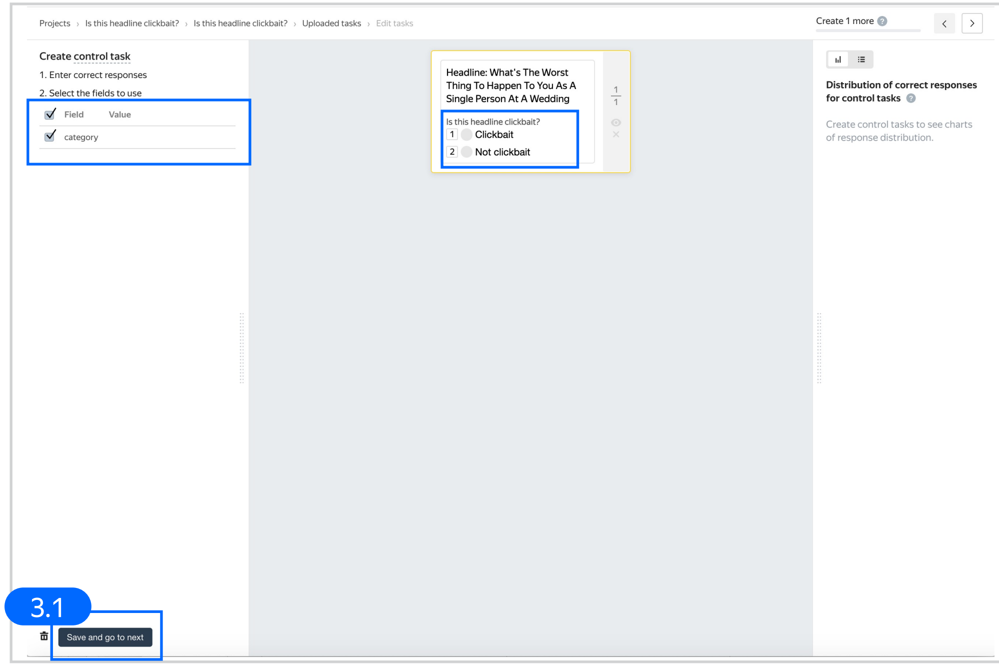
Note: Control tasks are tasks that already contain the correct response. They are used for checking the quality of responses from performers. The performer's response is compared to the response you provided. If they match, it means the performer answered correctly.



3.1. Enter correct responses for your control tasks. Check the result output field, which compares the user's response to the control task. Look at the headline, select the response, click **Save and go to next**. Repeat until you have 20 control tasks.

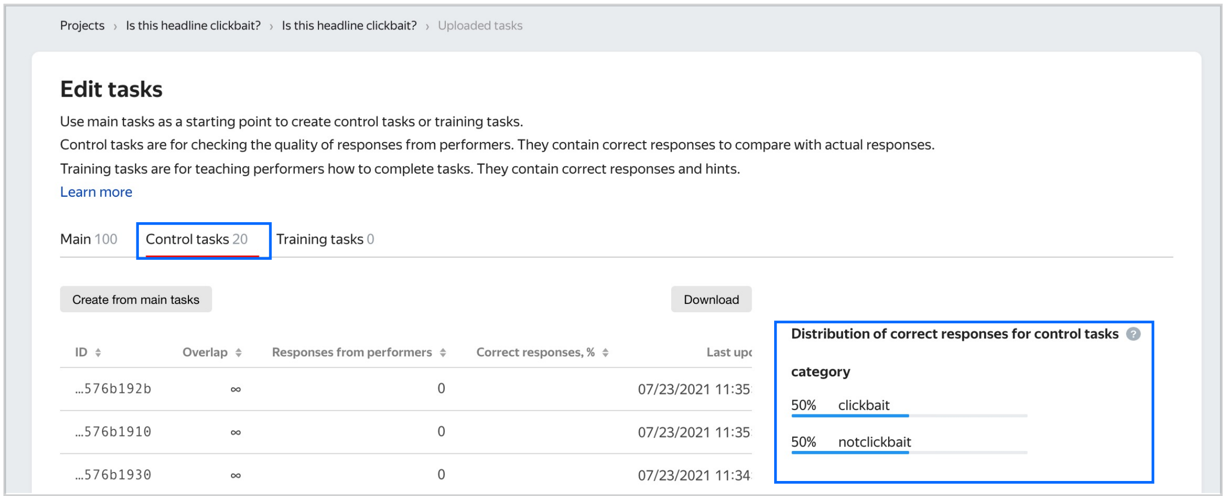
You need to create 20 tasks, because in small pools, control tasks should account for 10–20% of all tasks.

To learn more about creating control tasks, go to the Requester's Guide.

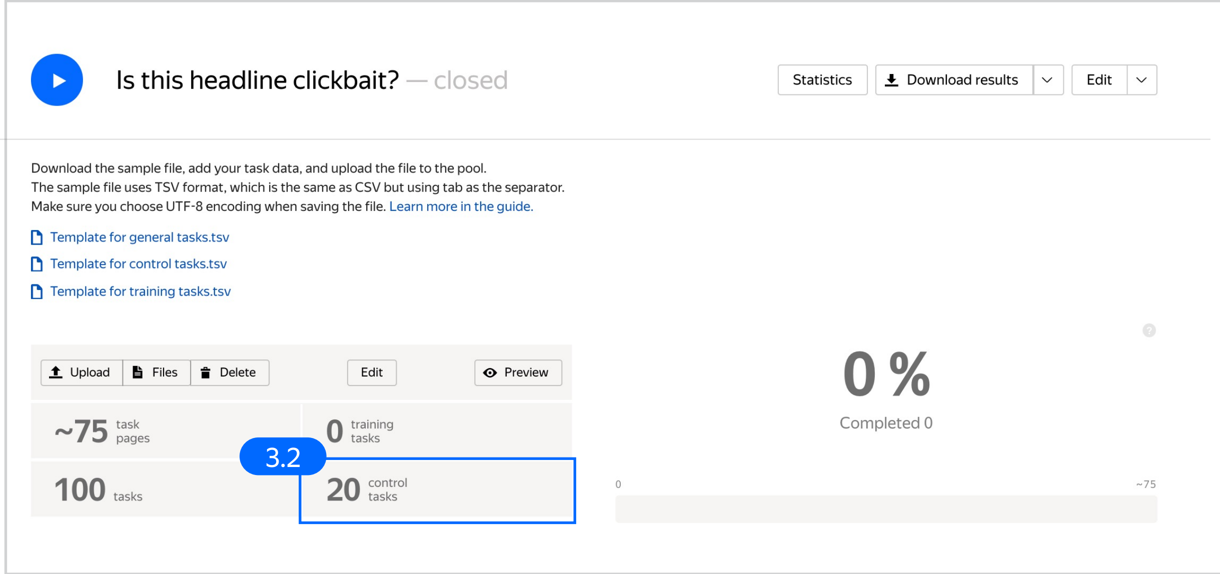


Tip. Make sure to include different variations of correct responses in equal amounts.

Go to **Uploaded tasks → Control tasks → Distribution of correct responses for control tasks** tab.

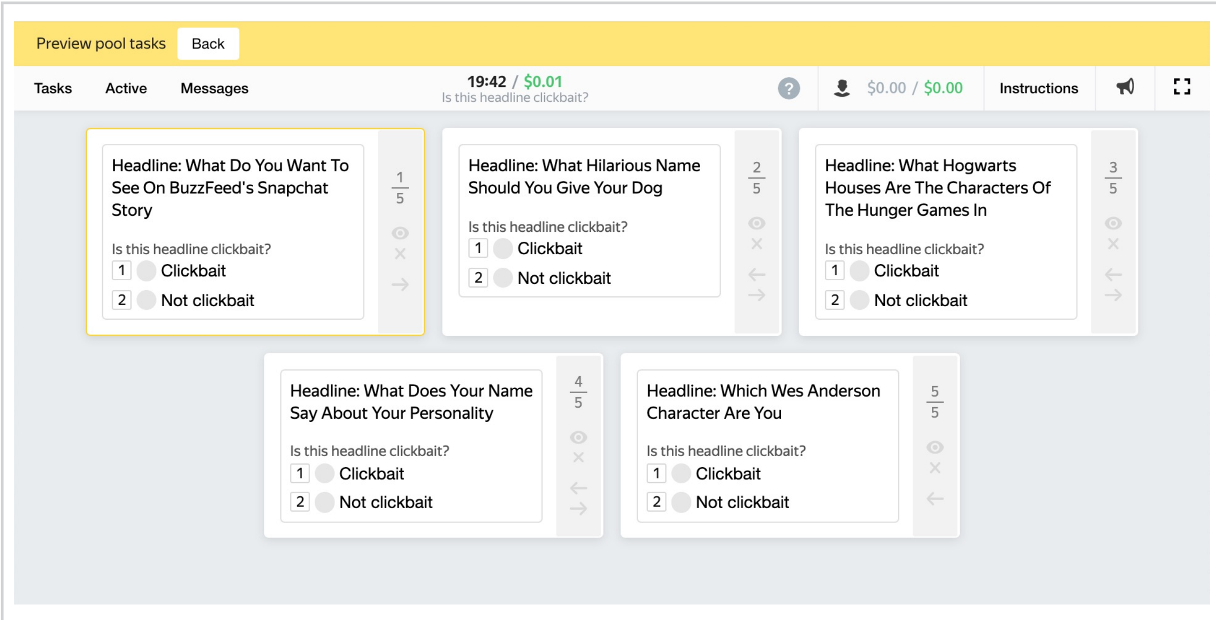


3.2. To check the number of control tasks, go to the **Pool page.**



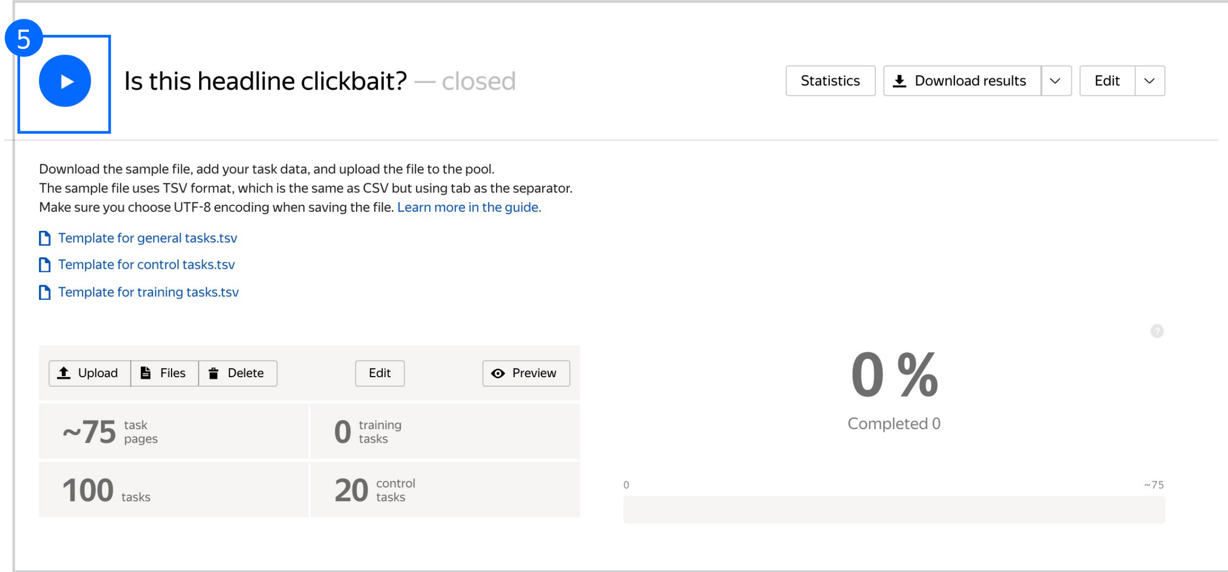
4. Preview the pool.

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.

   Make sure you start the training pool before you start the main pool.



Receiving responses

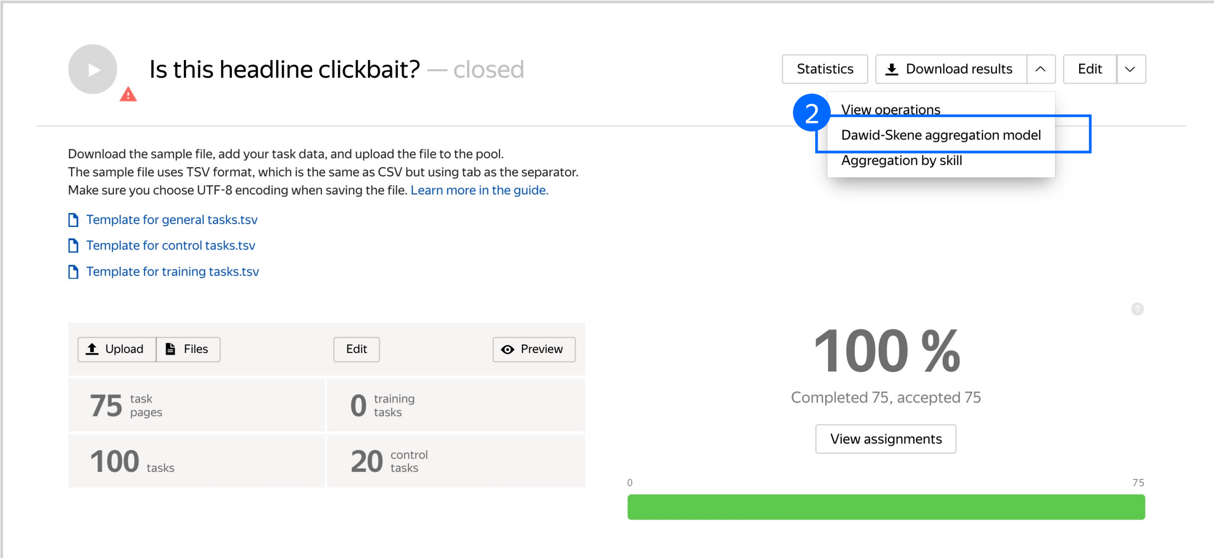
1. Wait until the pool is completed. Refresh the pool page to check progress.

Note: Aggregation takes from 5 to 20 minutes. During this time, you can start configuring your next project. Refresh the Operations page to check progress.

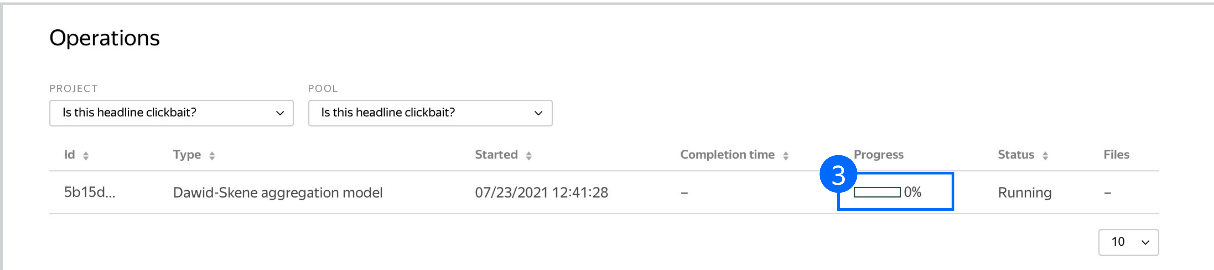
2. Click the **Download results** button and run aggregation using the Dawid-Skene model.

   We use this aggregation model because our questions are of comparable difficulty, and we don't have many control tasks.

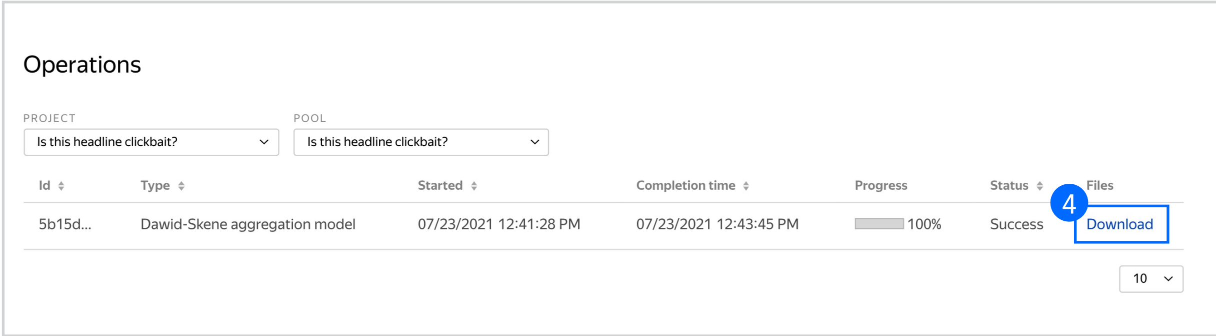
   Read more about the Dawid-Skene model in the Requester's Guide or get at an overview of different aggregation models our Knowledge Base.



3. Go to the list of operations and wait until aggregation is complete.
   Refresh the **Operations** page to check progress.



4. When aggregation is complete, download the TSV file with the results.



Appendix

Interface code
Step 4.1.

```
{
"view": {
"type": "view.list",
"items": [
{
"type": "view.group",
"content": {
"type": "view.list",
"items": [
{
"type": "view.text",
"content": {
"type": "helper.join",
"items": [
"Headline: ",
{
"type": "data.input",
"path": "headline"
}
],
"by": ""
}
},
{
"type": "field.radio-group",
"label": "Is this headline clickbait?",
"options": [
{
"label": "Clickbait",
"value": "clickbait"
},
{
"label": "Not clickbait",
"value": "notclickbait"
}
],
"data": {
"type": "data.output",
"path": "category"
},
"validation": {
"type": "condition.required",
"hint": "you need to select one answer"
}
}
]
}
}
]
},
"plugins": [
{
"type": "plugin.toloka",
"layout": {
"kind": "scroll",
"taskWidth": 300
}
},
{
"1": {
"type": "action.set",
"data": {
"type": "data.output",
"path": "category"
},
"payload": "clickbait"
},
"2": {
"type": "action.set",
"data": {
"type": "data.output",
"path": "category"
},
"payload": "notclickbait"
},
"type": "plugin.hotkeys"
}
]
}
```