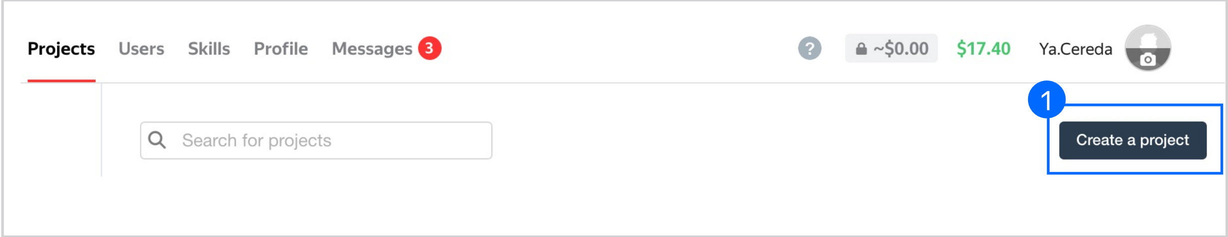
Toloka

Search relevance evaluation manual

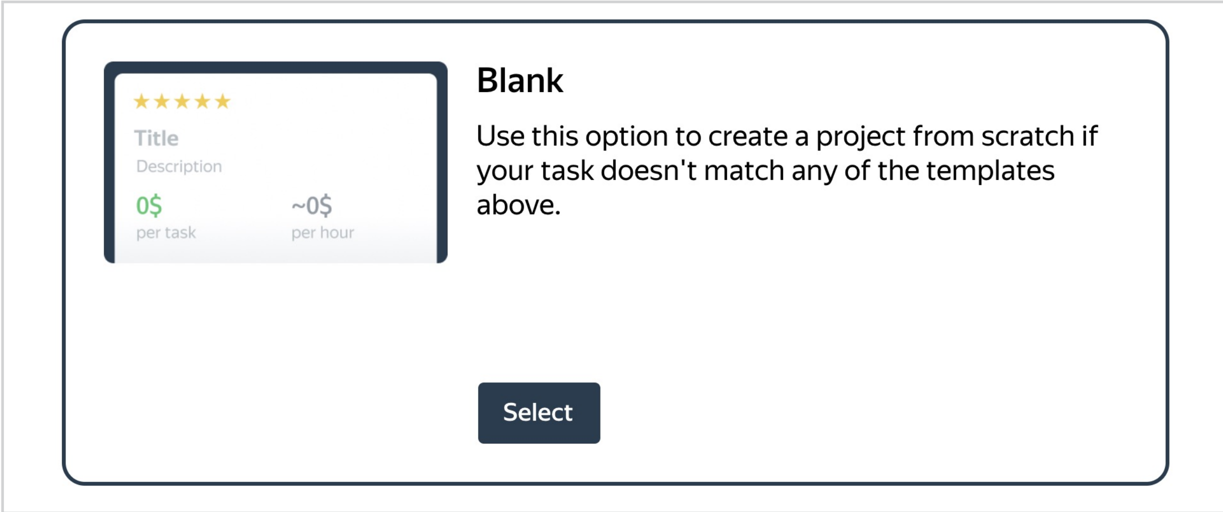
Create a project

1. Click **Create a project.**



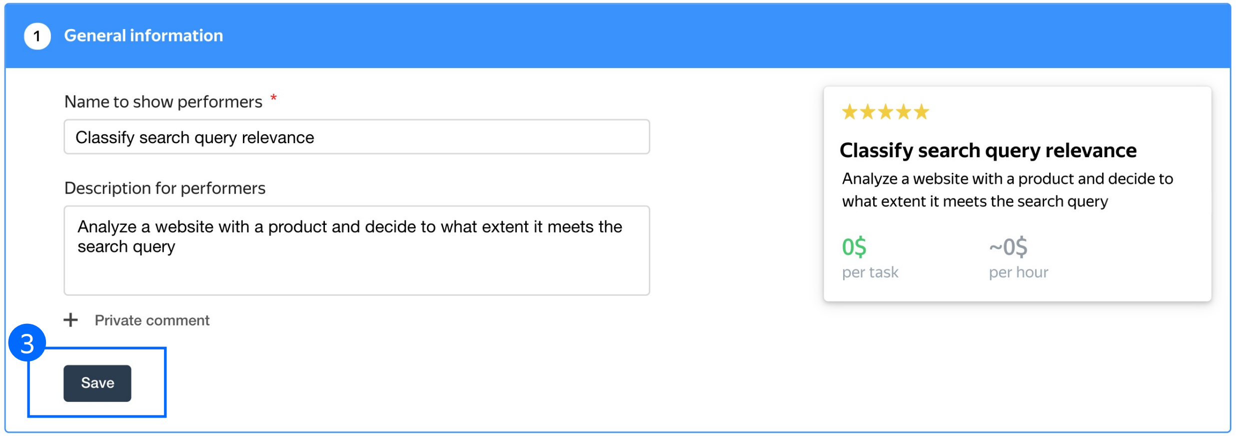
2. Choose the **Blank** template.

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



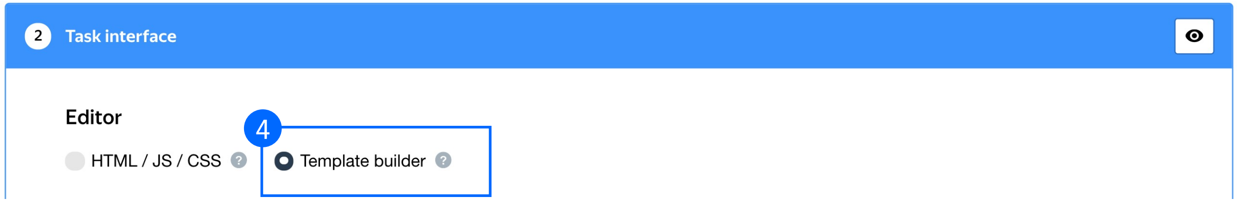
3. Enter a clear project name and description. Click **Save.**

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



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

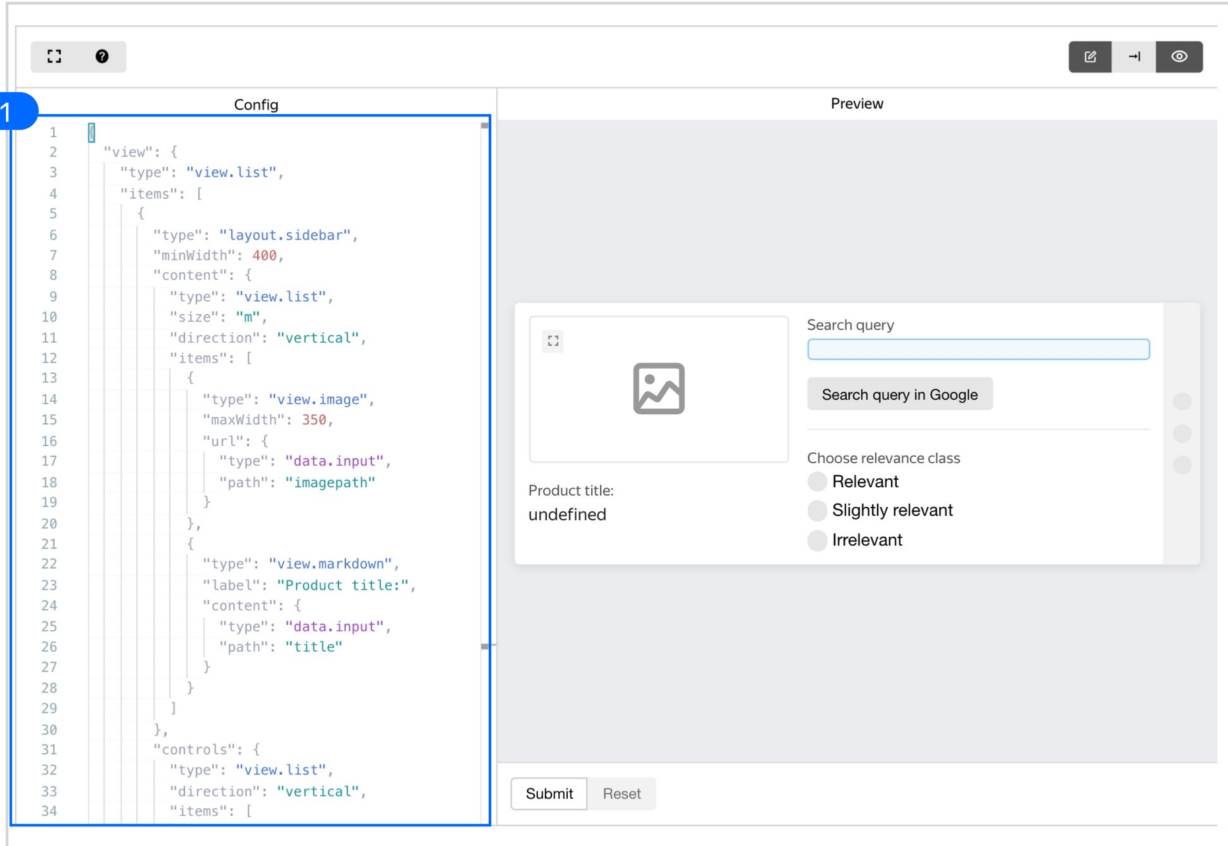
   Read more about the [Template builder](#) in the Requester's Guide.



4.1. You can either update the existing config or delete it and paste the code provided at the end of this manual (in the appendix).

   Note: This interface contains a query, a picture of a product, and its title, which needs to be assessed. There is a button for checking this query in Google, which is handy because the query might not be obvious and performers will often need to look it up. There is also a plugin that checks if a label was really chosen.

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

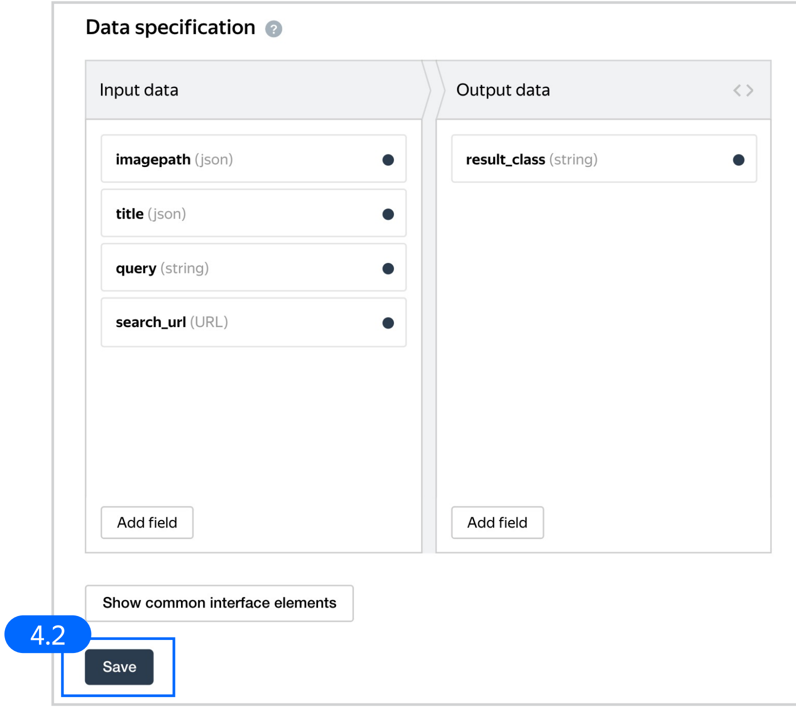


4.2. Make sure the specifications look like this:

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.

Note: We are using screenshots to make this demo more robust against possible webpage changes. Another way is to use an iframe and let the performers assess the whole webpage.

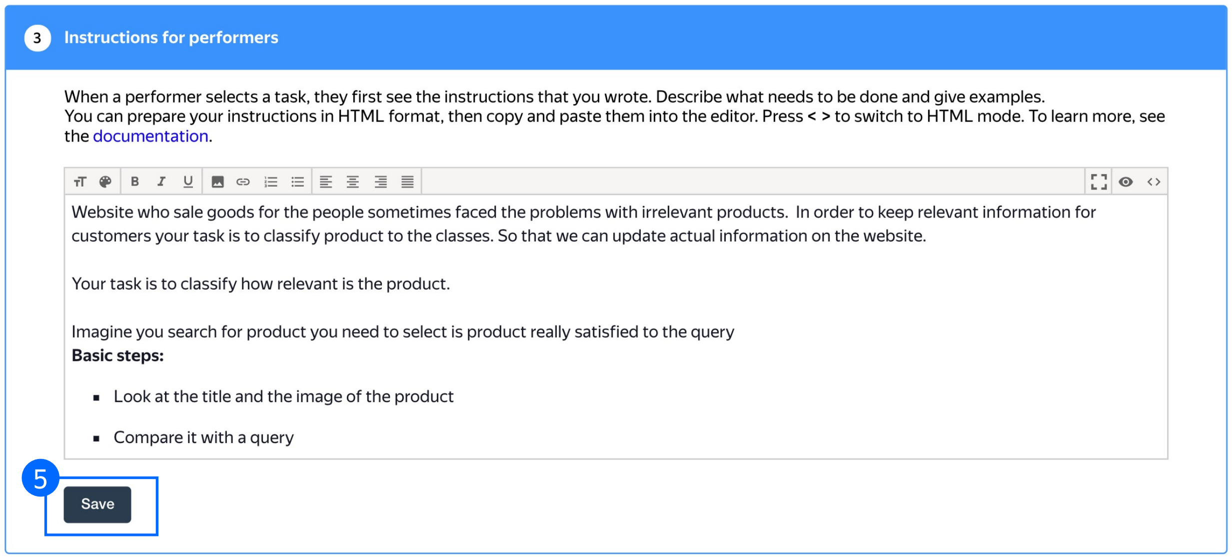
Read more about input and output data specifications in the Requester's Guide.



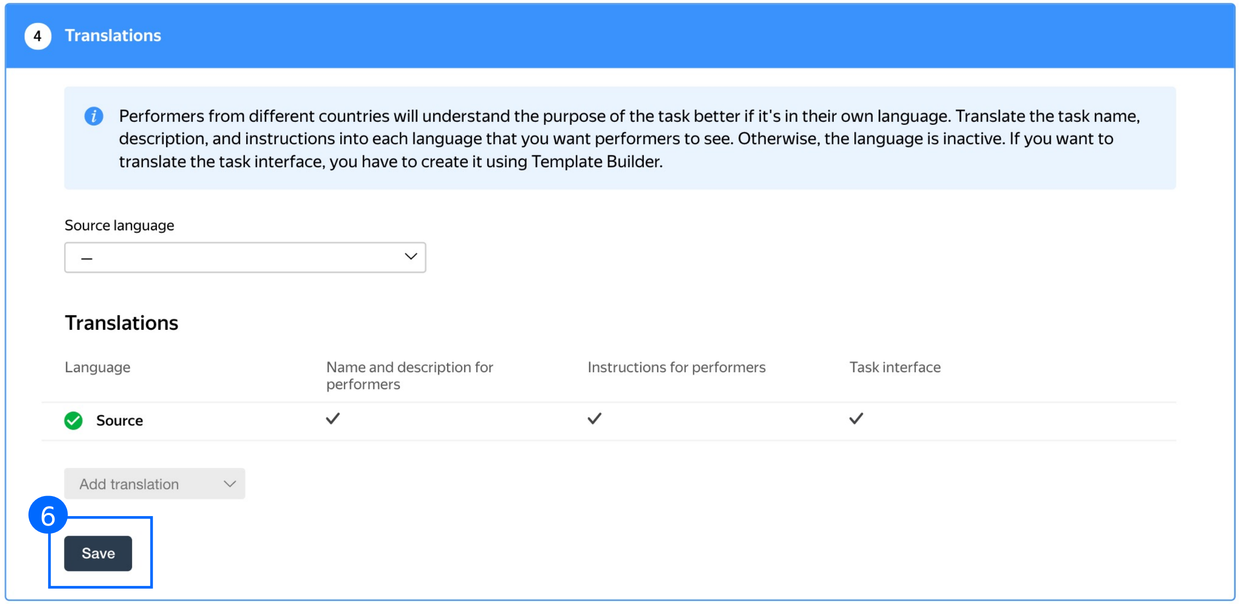
5. Write comprehensive instructions.

Note: Instructions are essential for complex tasks like relevance evaluation that are based on a set of rules and various criteria. Make sure to not only describe the general idea, but also go through examples and explain the evaluation logic in each case. We recommend trying to evaluate around two dozen cases yourself to get more insights for the instructions.

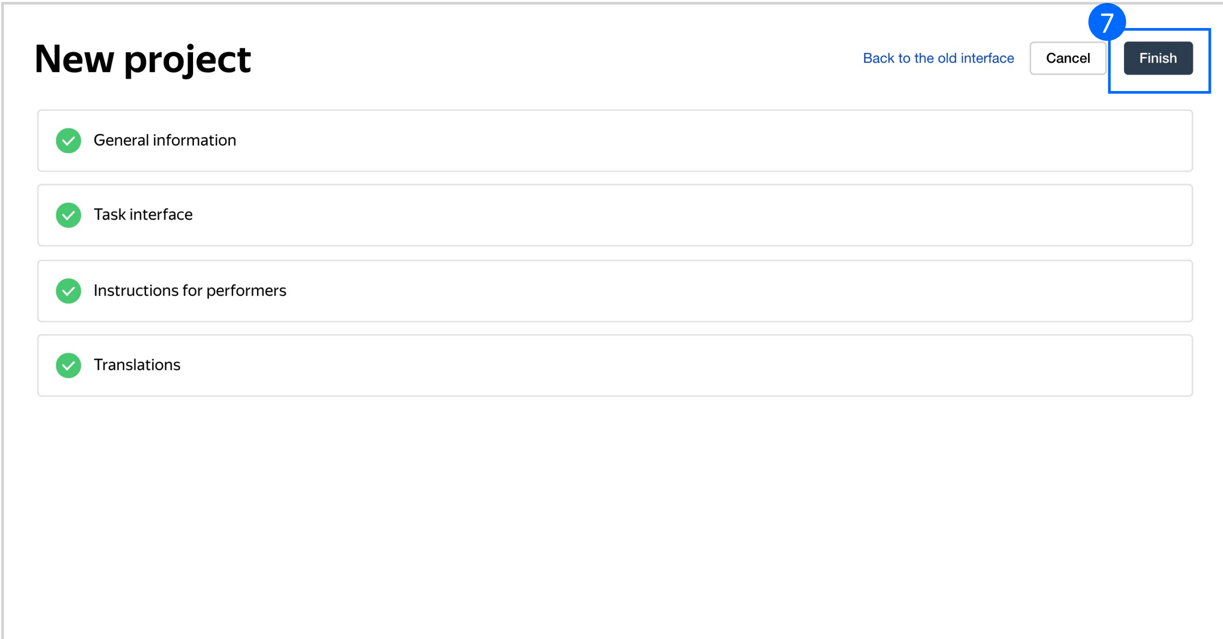
Get more tips on designing instructions in our Knowledge Base.



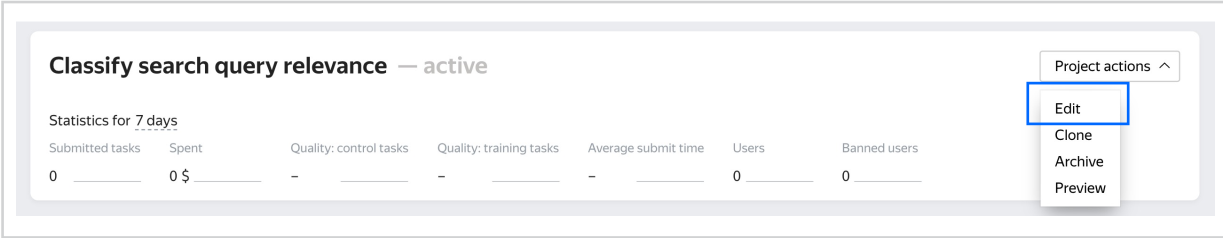
6. Leave the **Translations** block as default and click **Save.**



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.



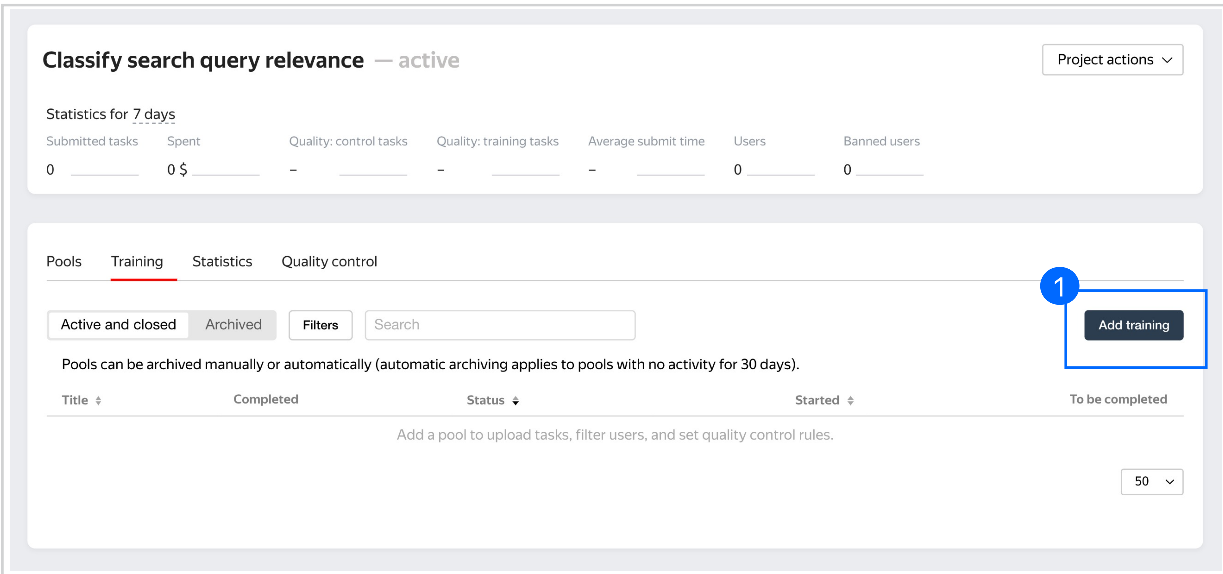
Create a training pool

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

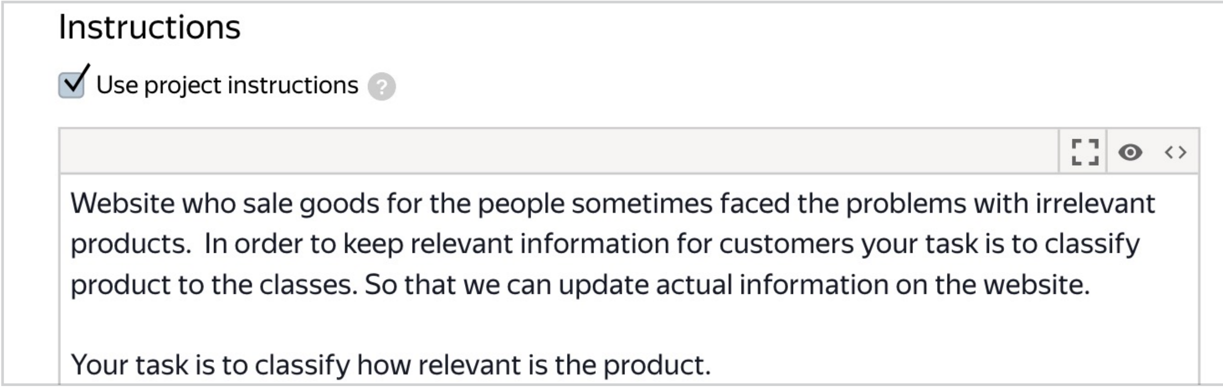
   Note: Since relevance evaluation is based on rules, not just common sense or certain skills, we recommend investing some time on learning how to explain all the rules. Training needs to involve both common and extreme cases. The comments should explain the underlying logic rather than just state the correct answers.

   Note: A well-grounded training exercise 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.



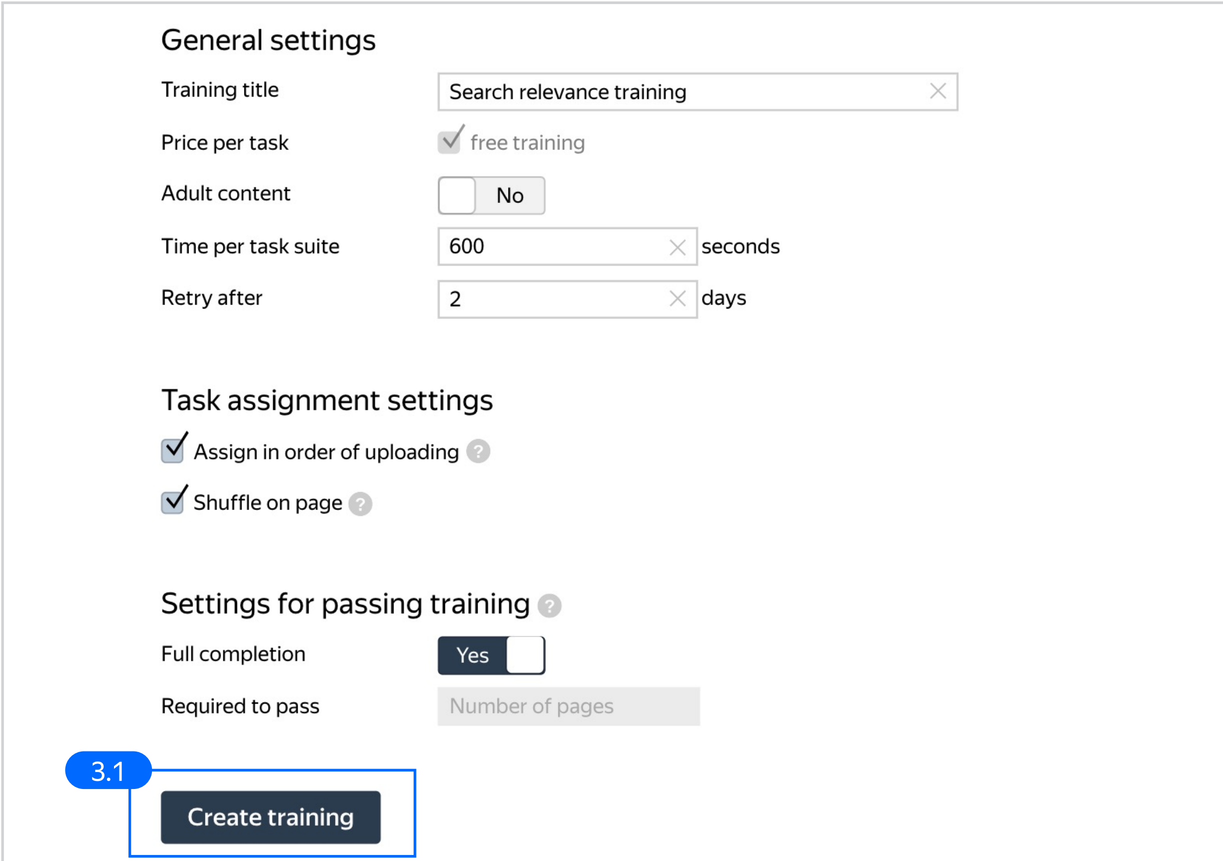
2. Use the existing project instructions.



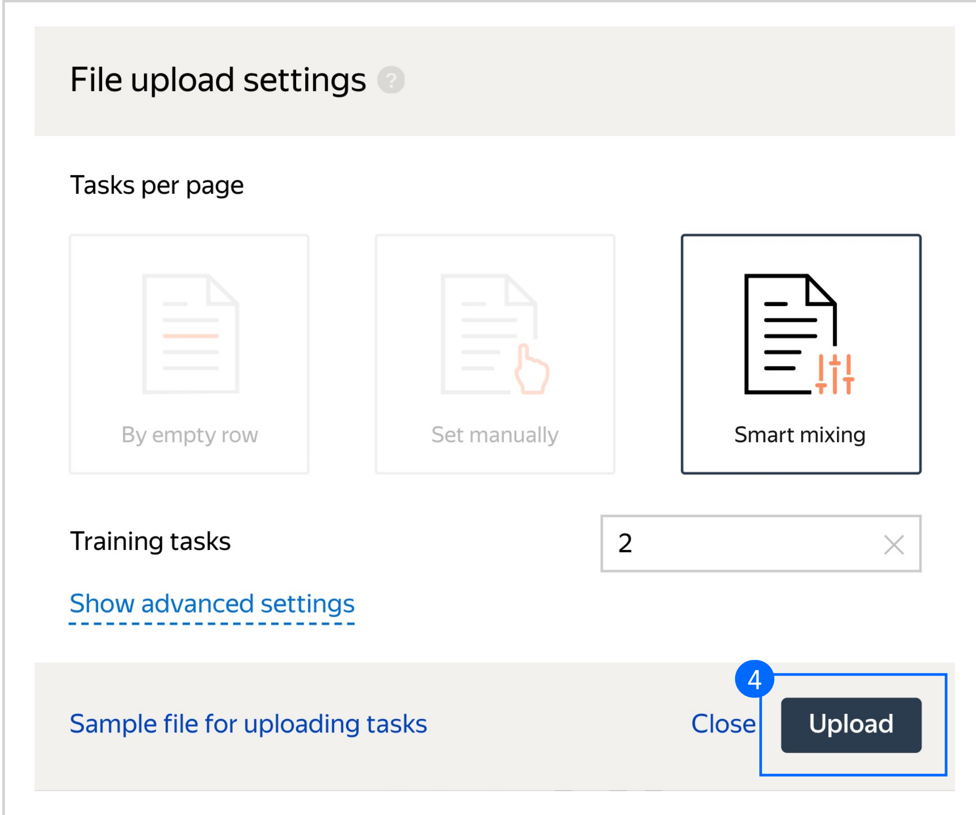
3. Specify the training pool settings:

3.1. Click **Create training.**

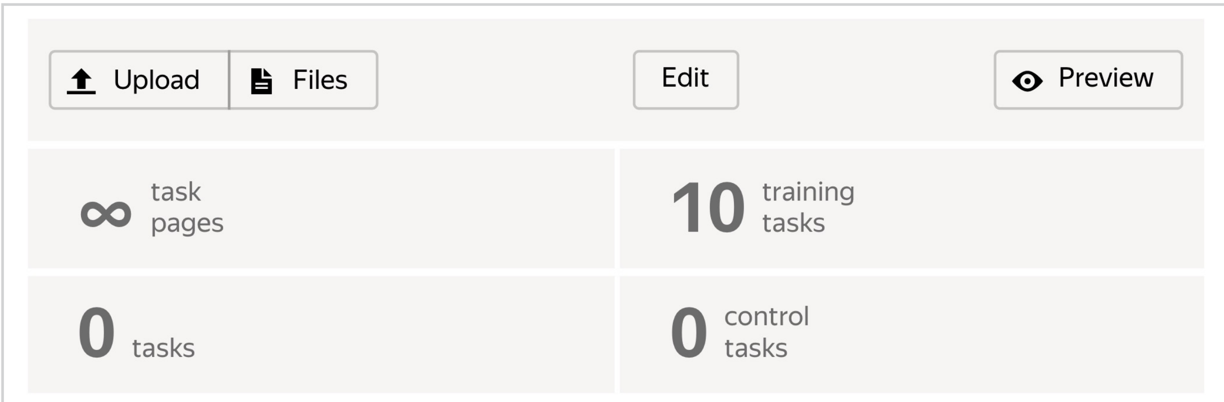
Read more about training pools in our Requester's Guide.



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

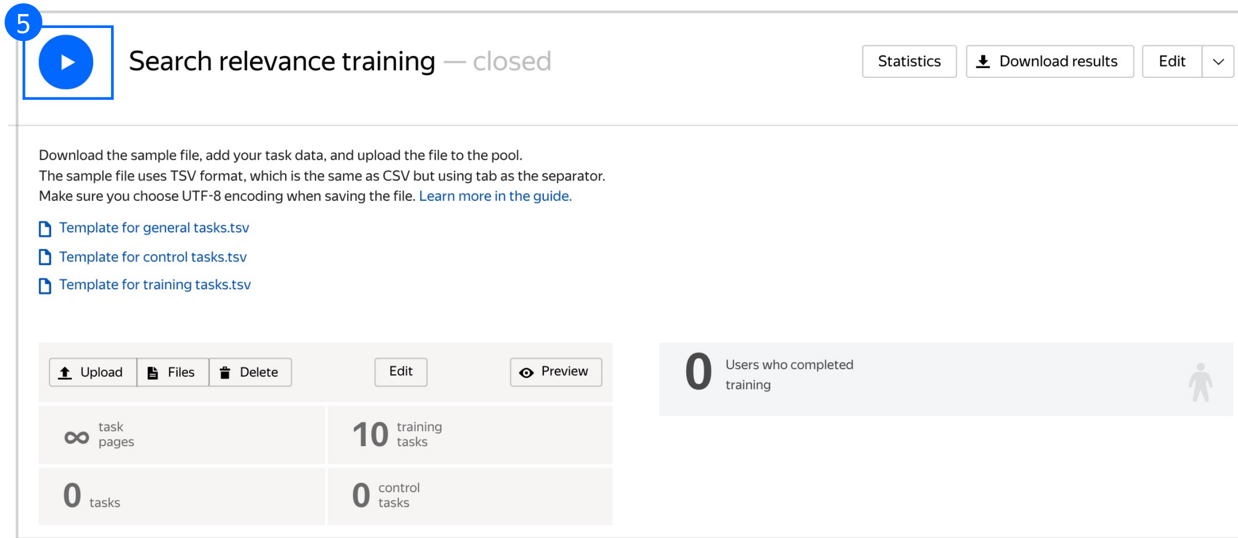


If the upload is successful, you will see the updated task counter.



5. Run the training pool.

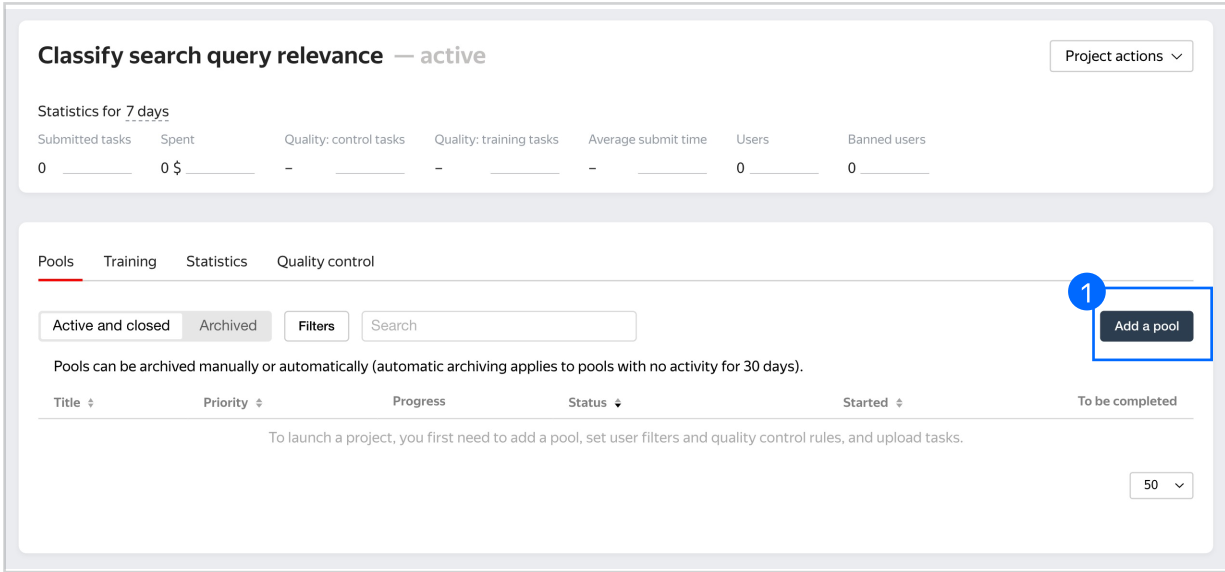
   Note: We recommend opening the training pool along with the main pool. Otherwise Tolokers will spend their time on training but get no access to real tasks, which is frustrating. Also, do not forget to close the training pool when there are no main tasks available anymore.



Create an exam pool

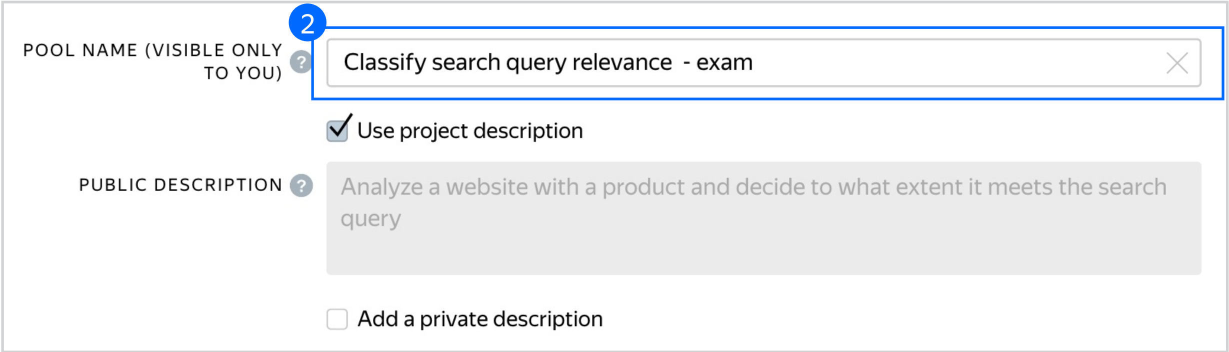
1. Click **Add a pool.**

   Note: We recommend adding an exam pool along with the training because relevance evaluation projects are usually more complicated than most crowdsourcing projects, and it takes a certain effort to master all the guidelines. The more guidelines there are, the greater will be the need to check if the performers have really learned them.



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

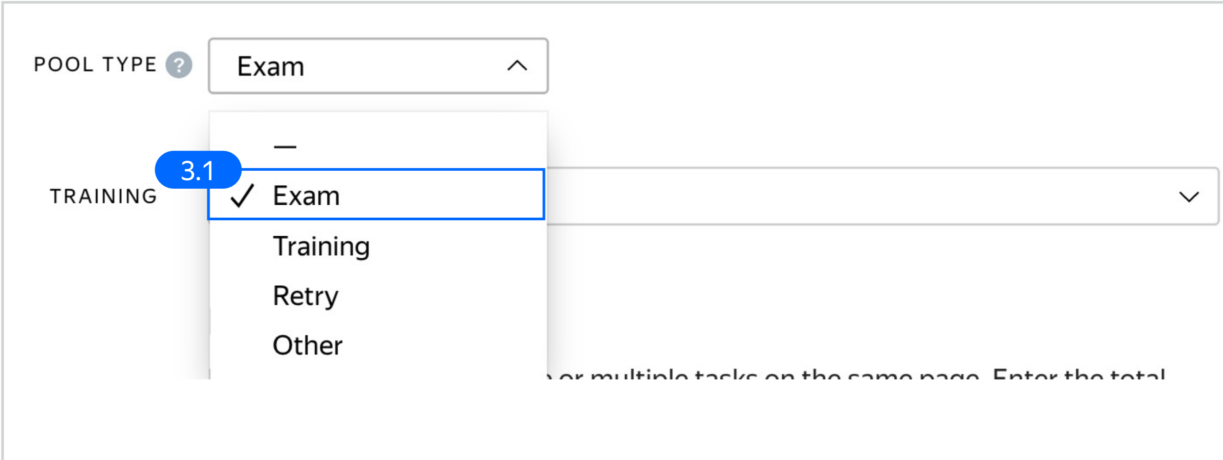
   The description can be either public or private. Choose the option you prefer.



3. Specify pool parameters:

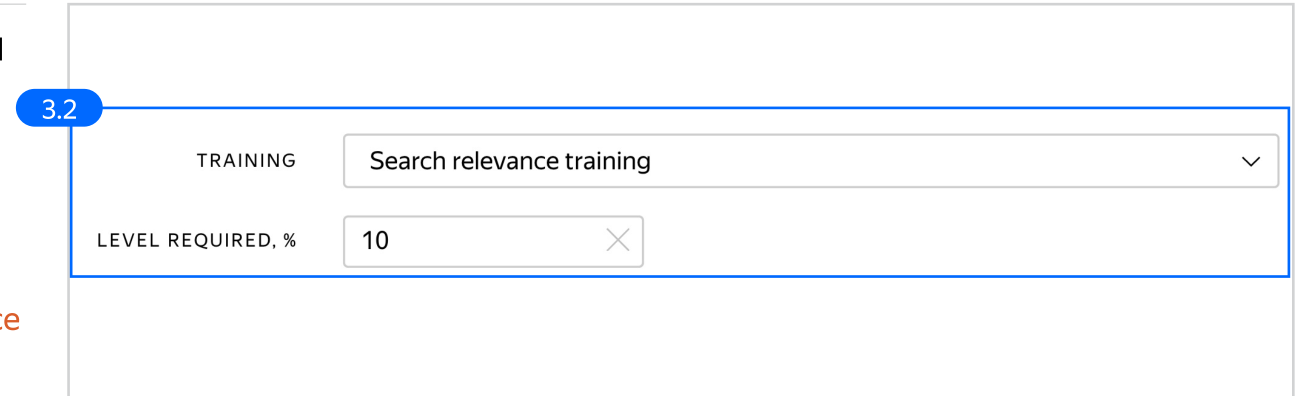
3.1. Select pool type. Choose **Exam.**

Note: The **"Exam"** type will allow you to launch these tasks with a price of zero. You can set a regular price, though. Setting a zero price might be reasonable if the exam is not very long but still complicated and you expect a low pass rate. That way you won't spend too much of your budget on the exam.



3.2. Attach the training pool you created earlier.

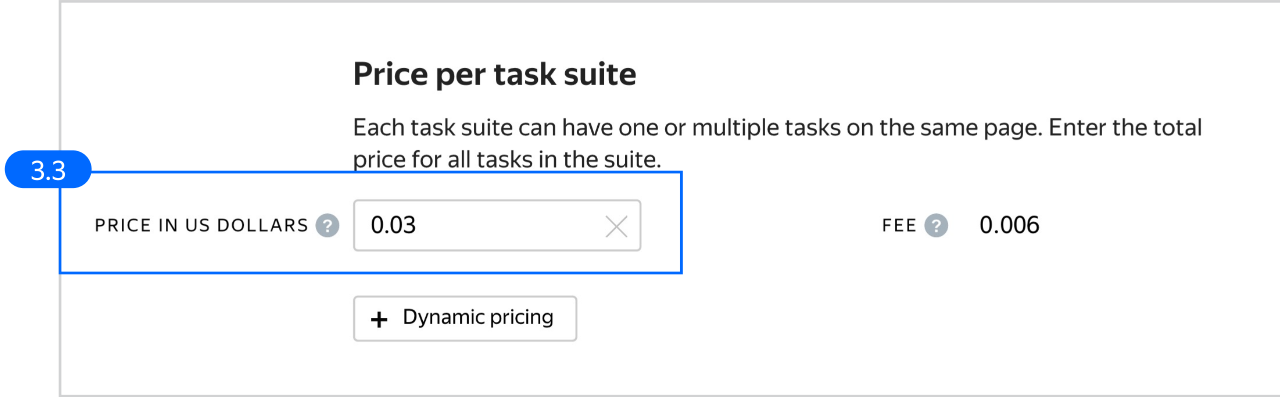
Note: The quality of the training can be low to just filter out potential deception, because we expect performers to make mistakes and learn from them (yet again, relevance is a complicated task type).



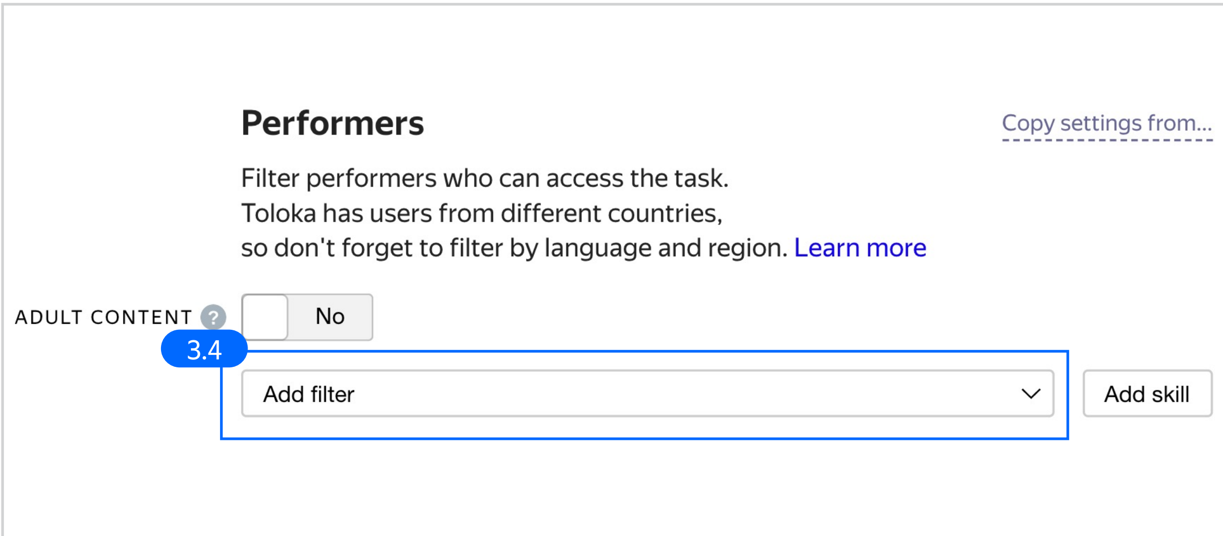
3.3. Set the price per task suite (for example, $0.03).

Note: You can use a zero price as well. However, if the exam is time-consuming, a zero price might be unfair, as the performers will spend a lot of time completing it.

Read more about [pricing principles]() in our Knowledge Base.

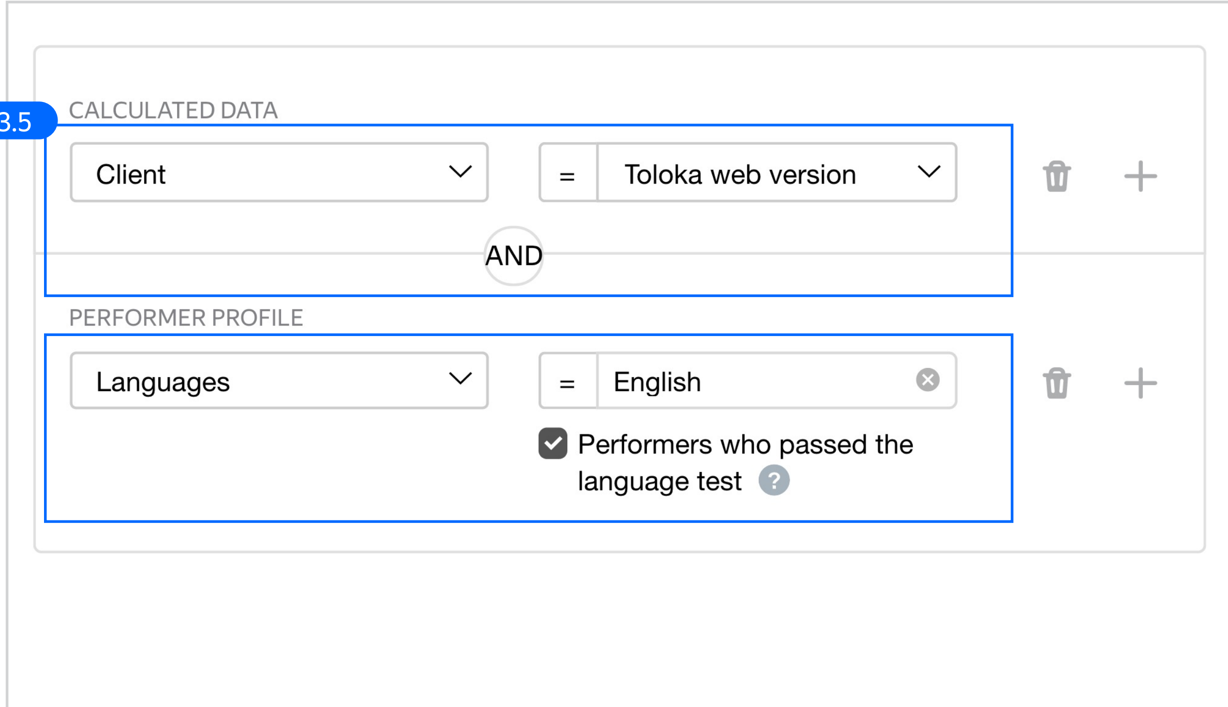


3.4. [Filter]() performers who can access the task. Choose “**No**” in the **Adult content** block. Click **Add filter** to choose the **Languages** and **Client** options in the list.



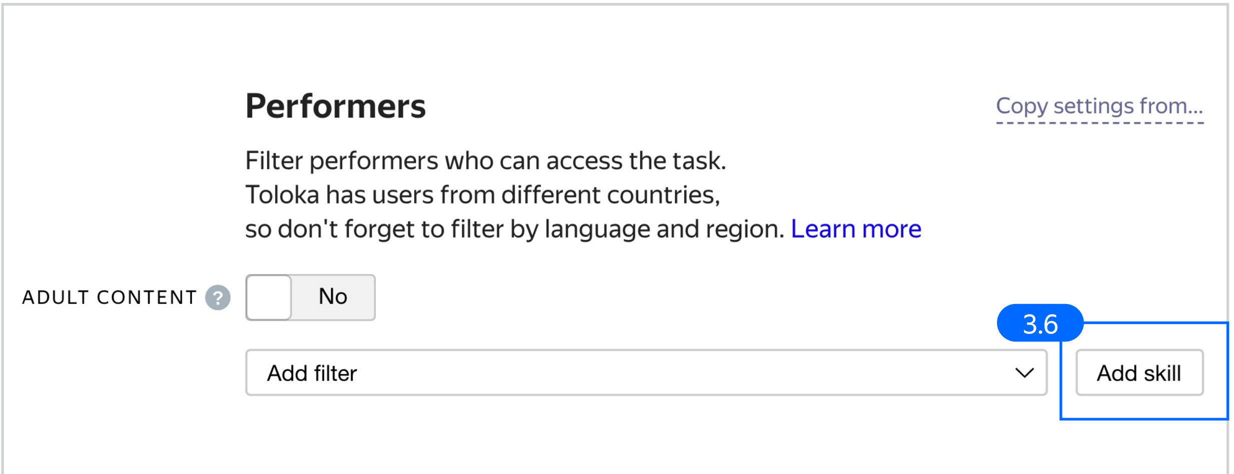
3.5. Choose **Client = Toloka web version.**

Then choose **Languages = English**. This way, performers who speak English will be invited to complete this task.

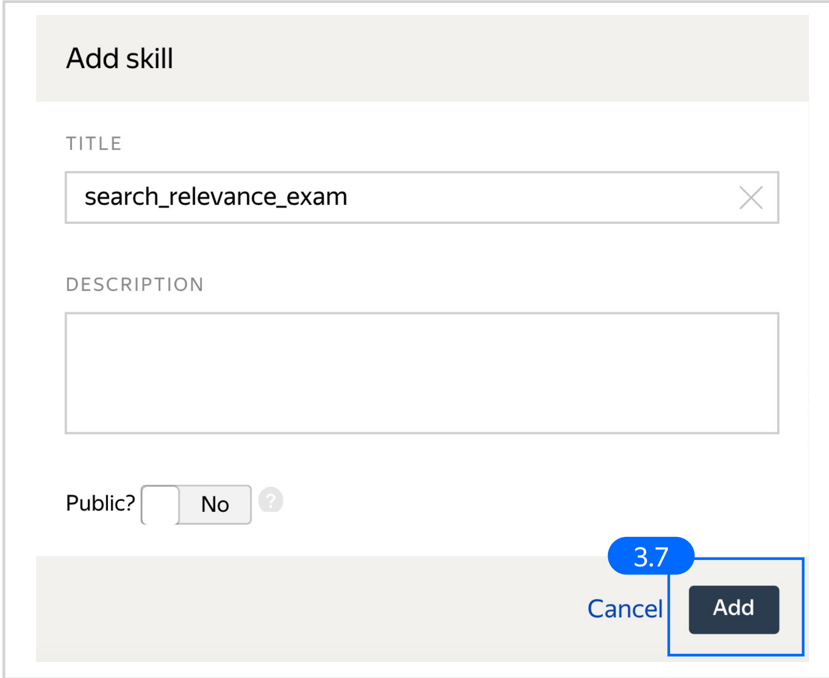


3.6. Set up exam quality calculation.

Click **Add skill.**



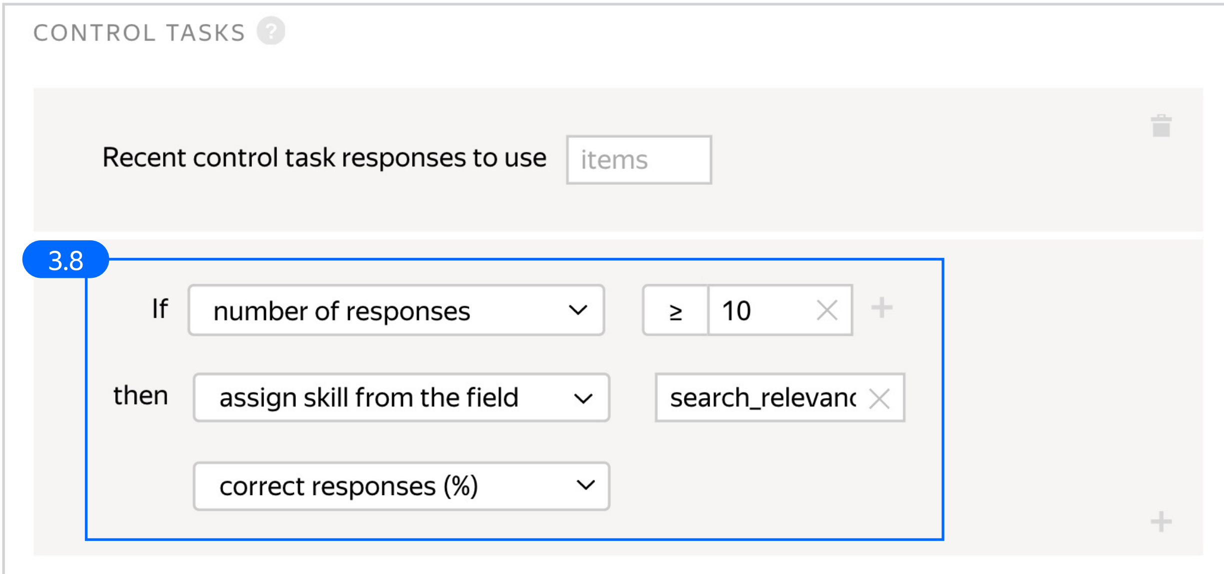
3.7. Fill in the skill title. Add a description if necessary. Click **Add**.



3.8. Scroll down to **Quality Control**. Add a **Control tasks** rule.

We will have 10 tasks in the exam pool, so the quality will be calculated after the whole exam has been passed.

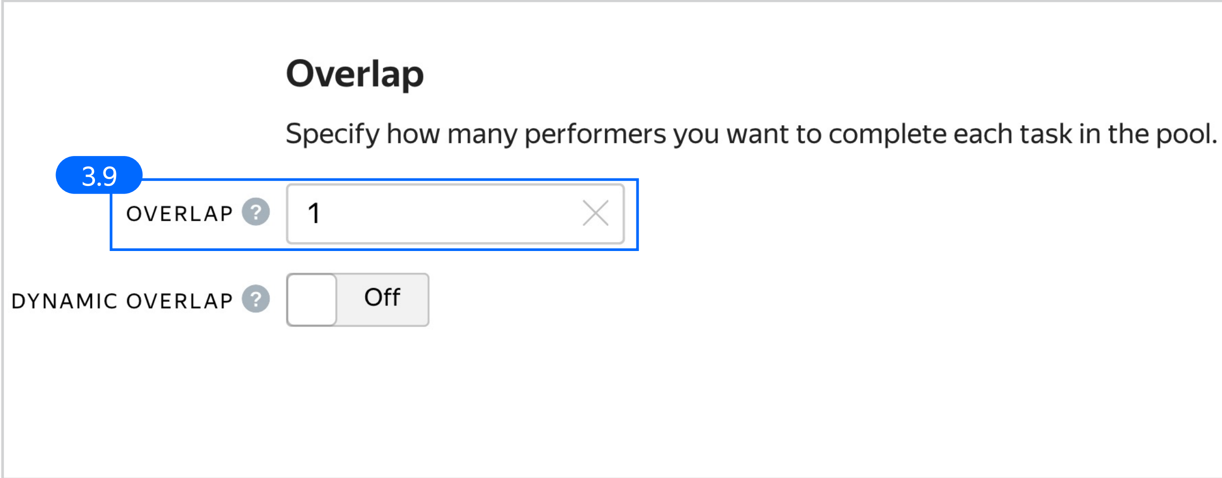
Note: We will then use this parameter as an entry filter for the main pool.



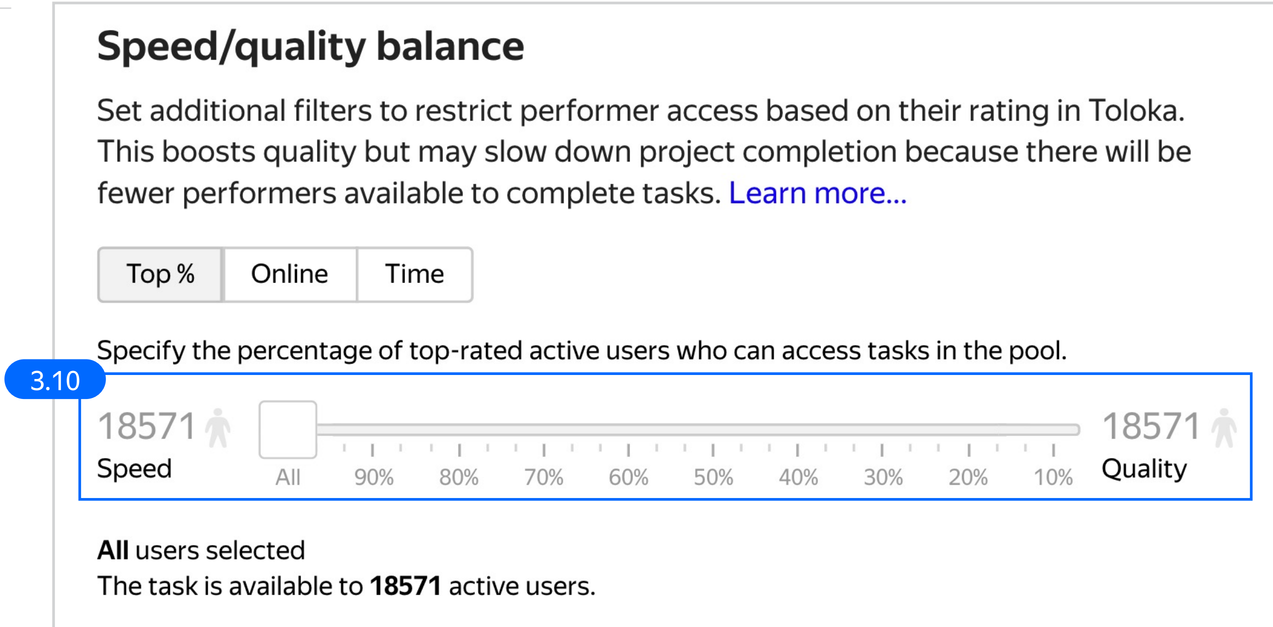
3.9. **Overlap**. This is the number of users who will complete the same task.

Note: If you are using the **"Exam"** task type, the overlap will be infinite by default, so you don't need to change that field.

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

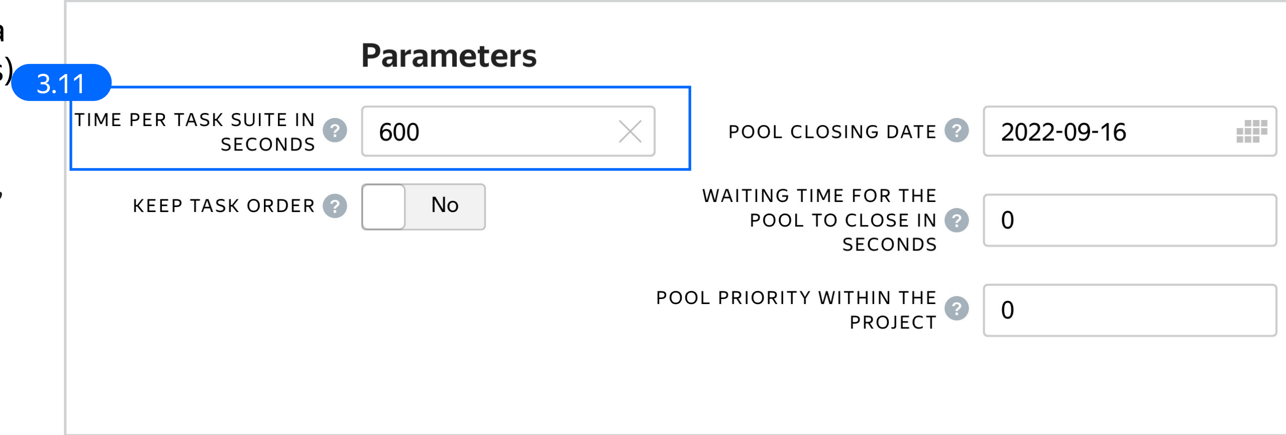


3.10. As we are assessing people in this pool the **Speed/quality** balance can be left as default.

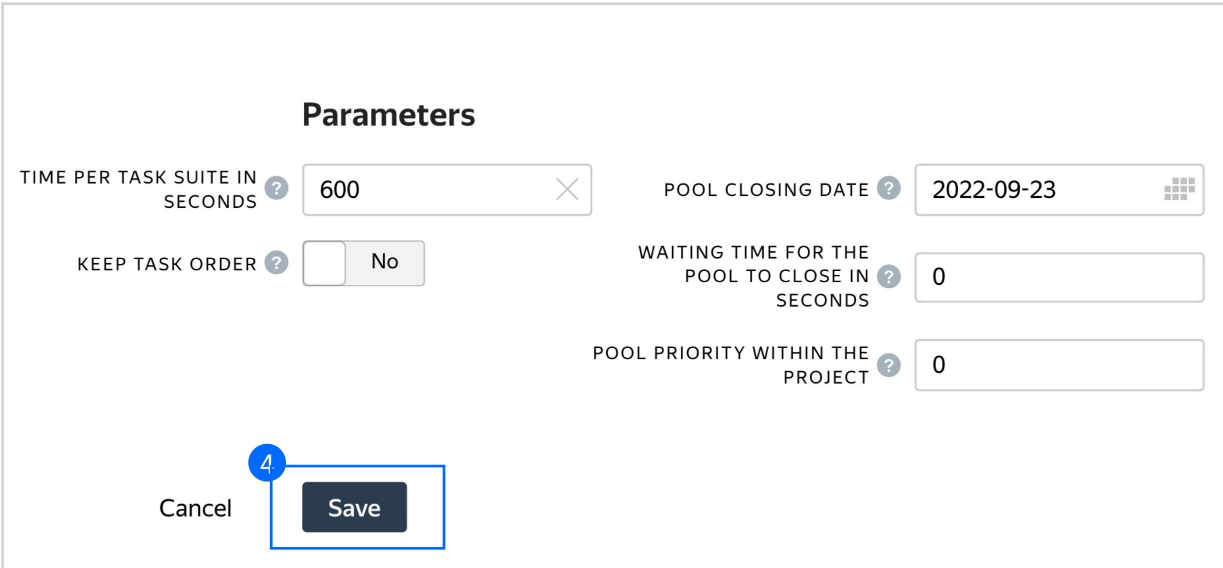


3.11. Specify the time given to complete a task suite (for example, 600 seconds).

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



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

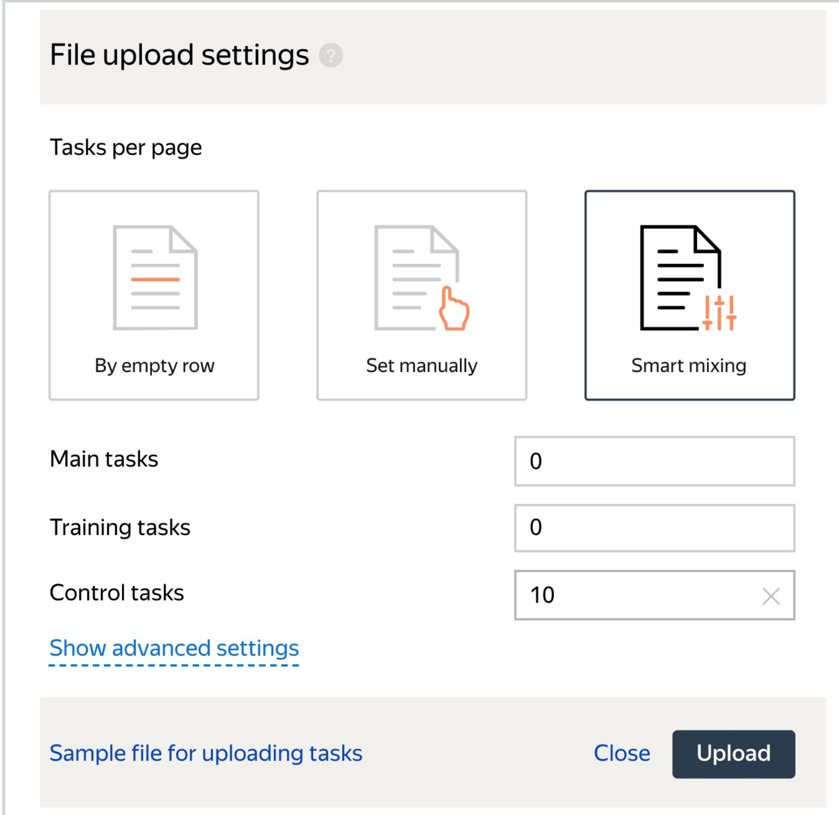


5. Upload exam tasks to the pool. You can download a sample file with exam tasks here:
Origin
License: PDDL

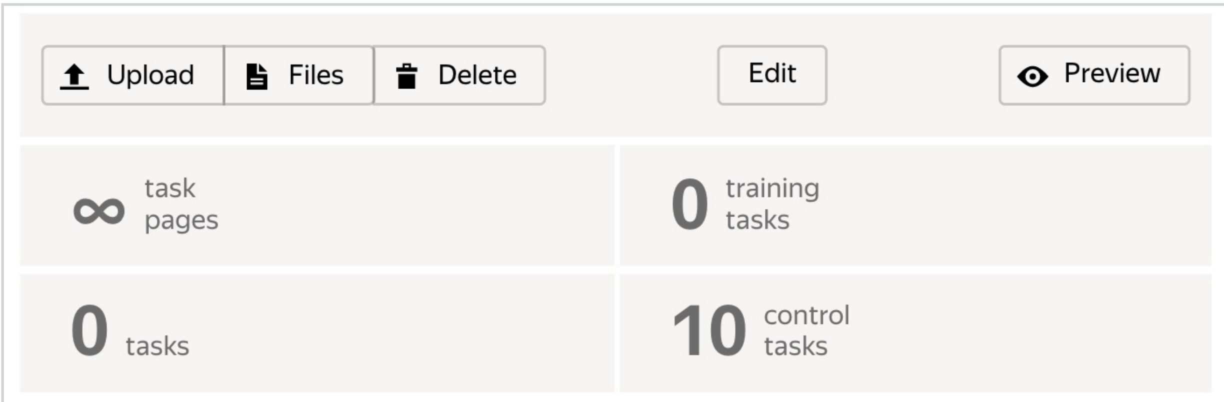
Note. You can create the exam tasks locally and upload them in a file like this one, or upload regular tasks and create controls via the UI.

Read more about this in the Requester's Guide.

Set 10 tasks per page.

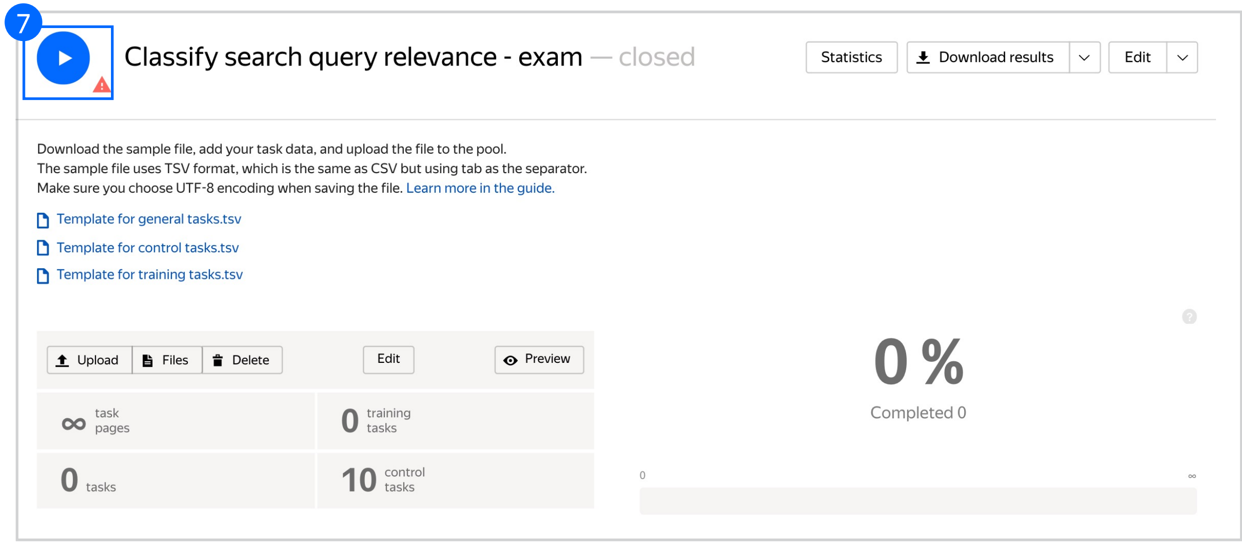


6. If the upload is successful, you will see the updated task counter.



7. Run the exam pool.

Note: We recommend opening the exam pool along with the main pool or just before. Otherwise Tolokers will spend their time on the exam pool but get no access to real tasks and might not return to the project later.

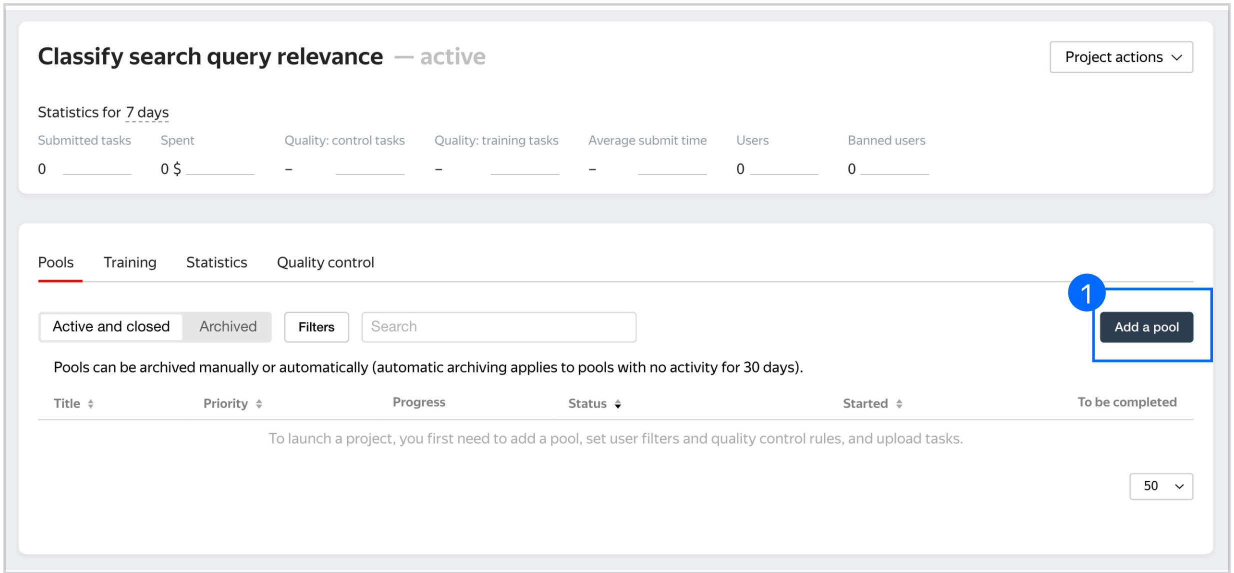


Create the main pool

1. Click **Add a pool.**

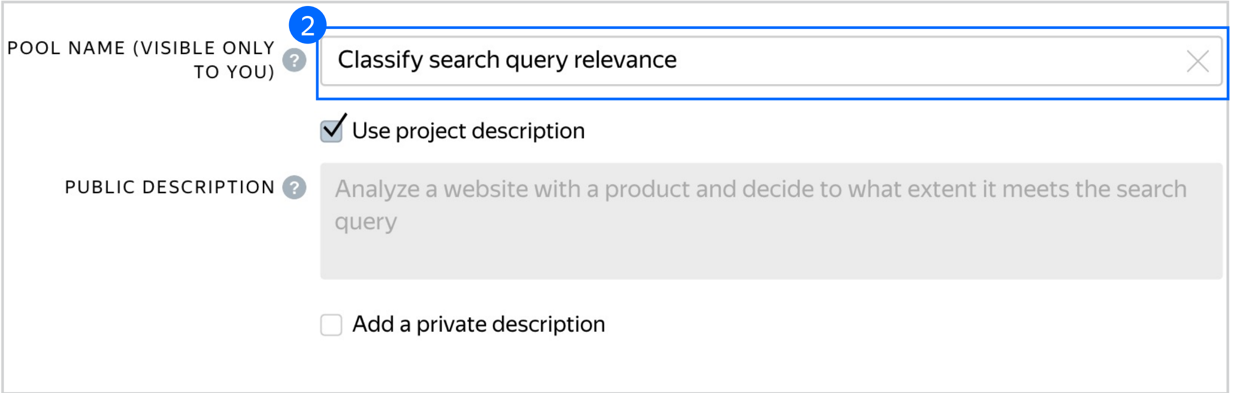
A pool is a set of paid tasks grouped into task pages. These tasks are sent out for completion at the same time.

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.

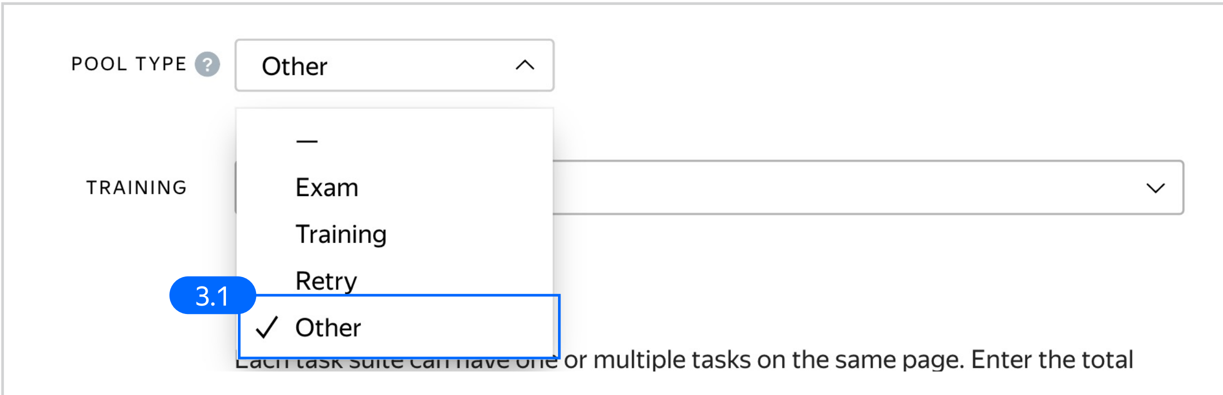
   The description can be either public or private. Choose the option you prefer.



3. Specify [pool parameters](#):

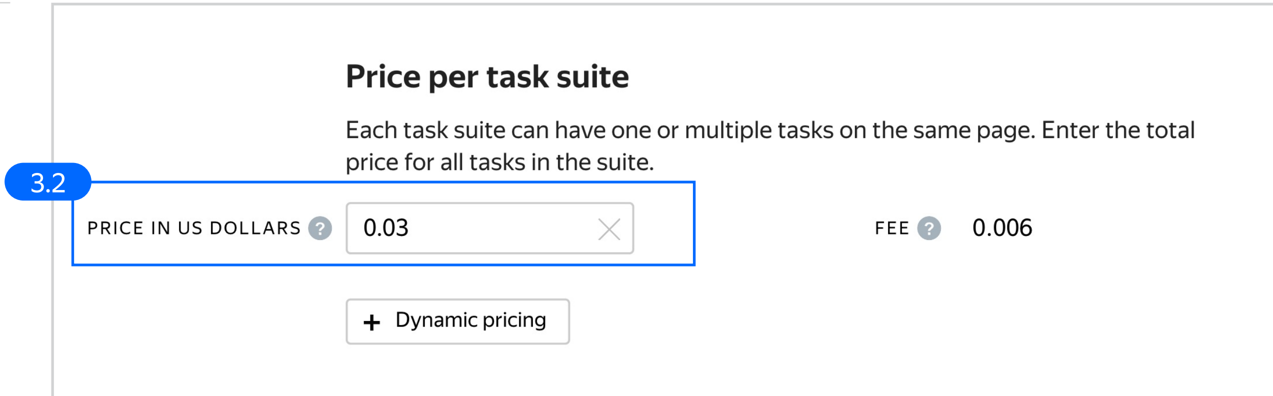
3.1. Select pool type. Choose **Other.**

Read more about [pool types](#) in the Requester's Guide.

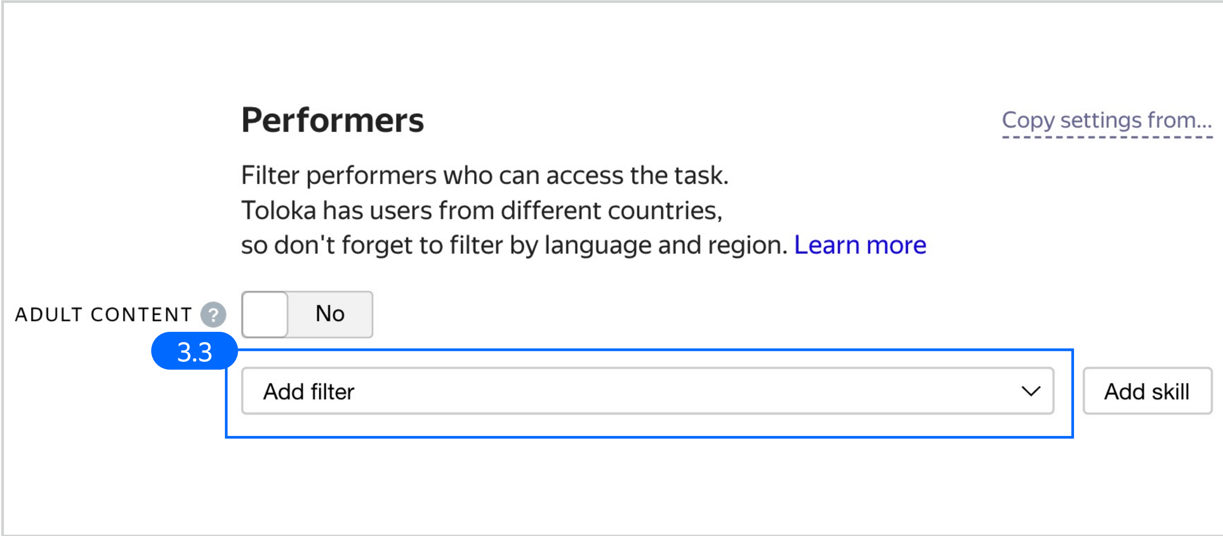


3.2. Set the price per task suite (for example, $0.03).

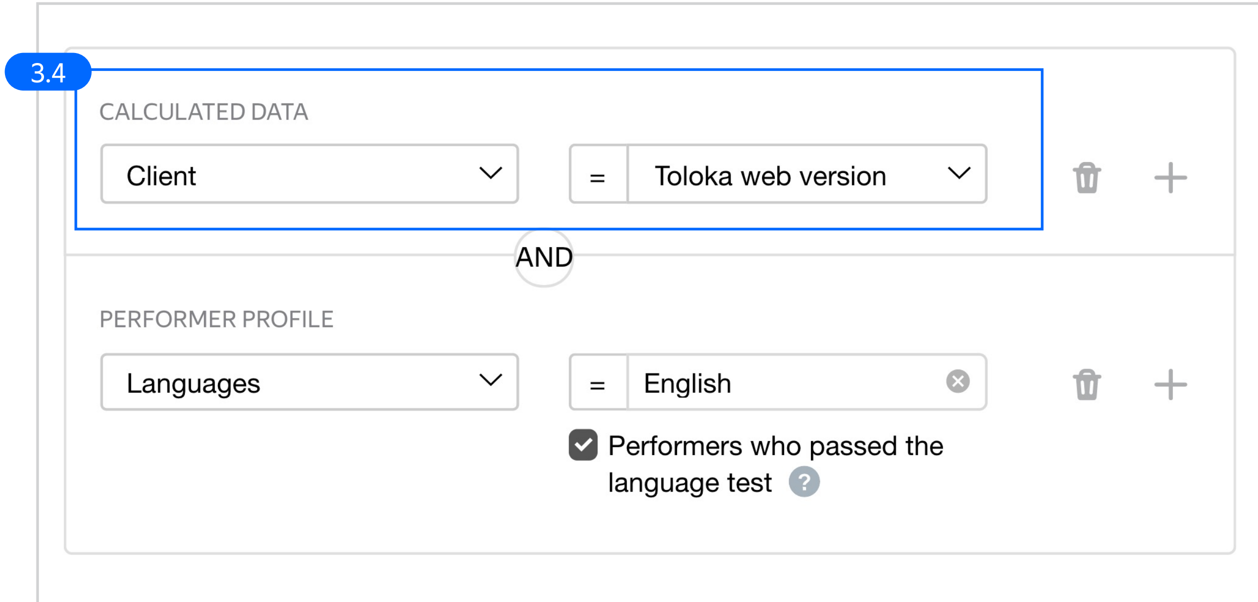
Read more about [pricing principles](#) in our Knowledge Base.



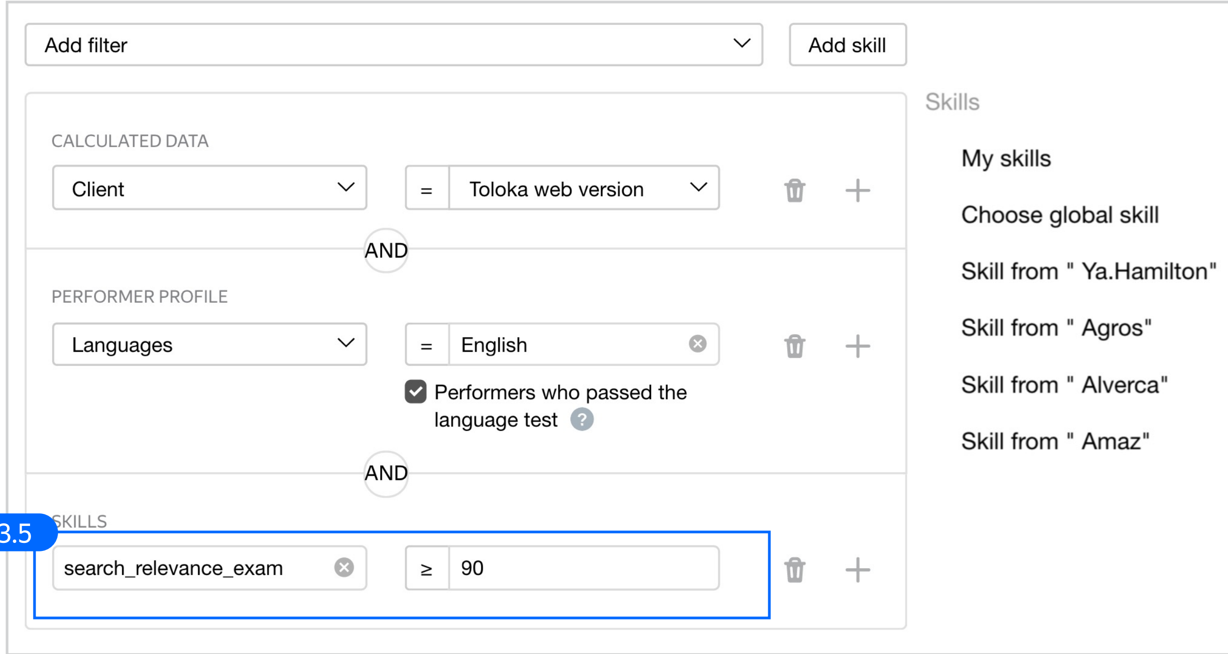
3.3. [Filter](#) performers who can access the task. Choose "**No**" in the **Adult content** block.



3.4. Add a language filter and limit the access to PC users only. The task involves a lot of actions (examining the image, analyzing search results for the query) which are not very mobile-friendly.

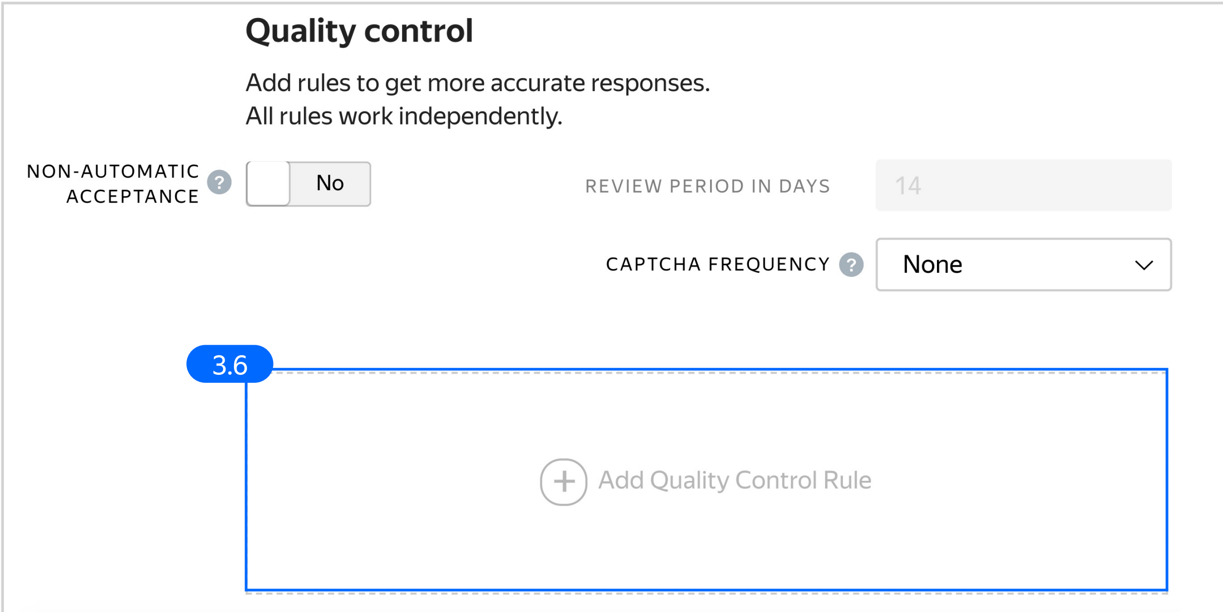


3.5. Add filtering by exam quality. Select **My skills** and find a skill where you recorded exam quality. Use it as a filter. The higher the access threshold is, the better the quality will be.



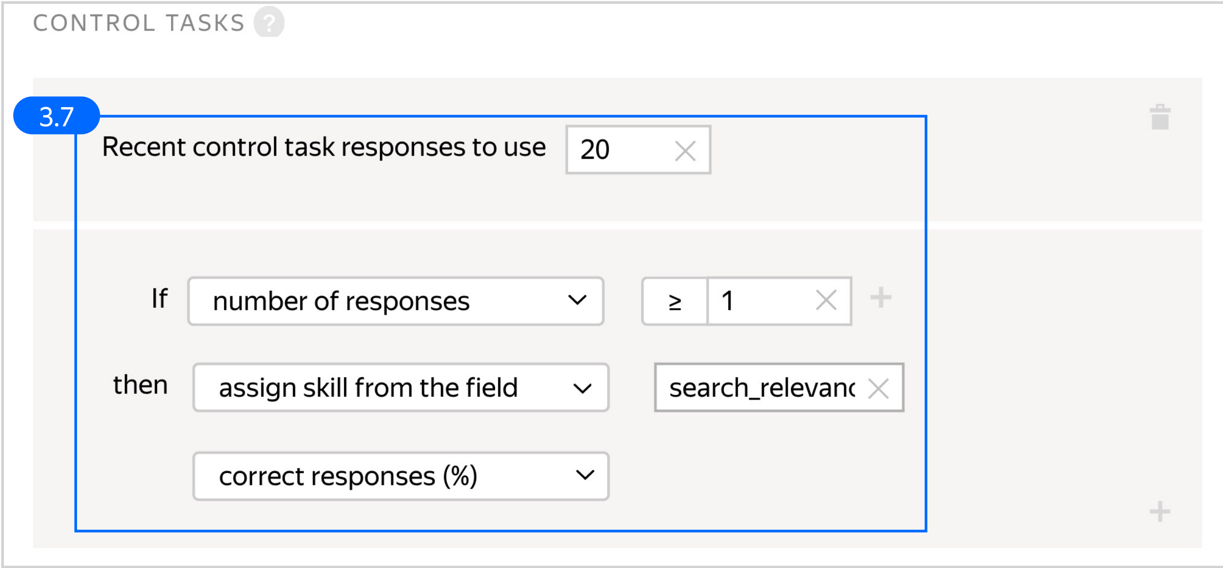
3.6. Set up Quality control.

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.

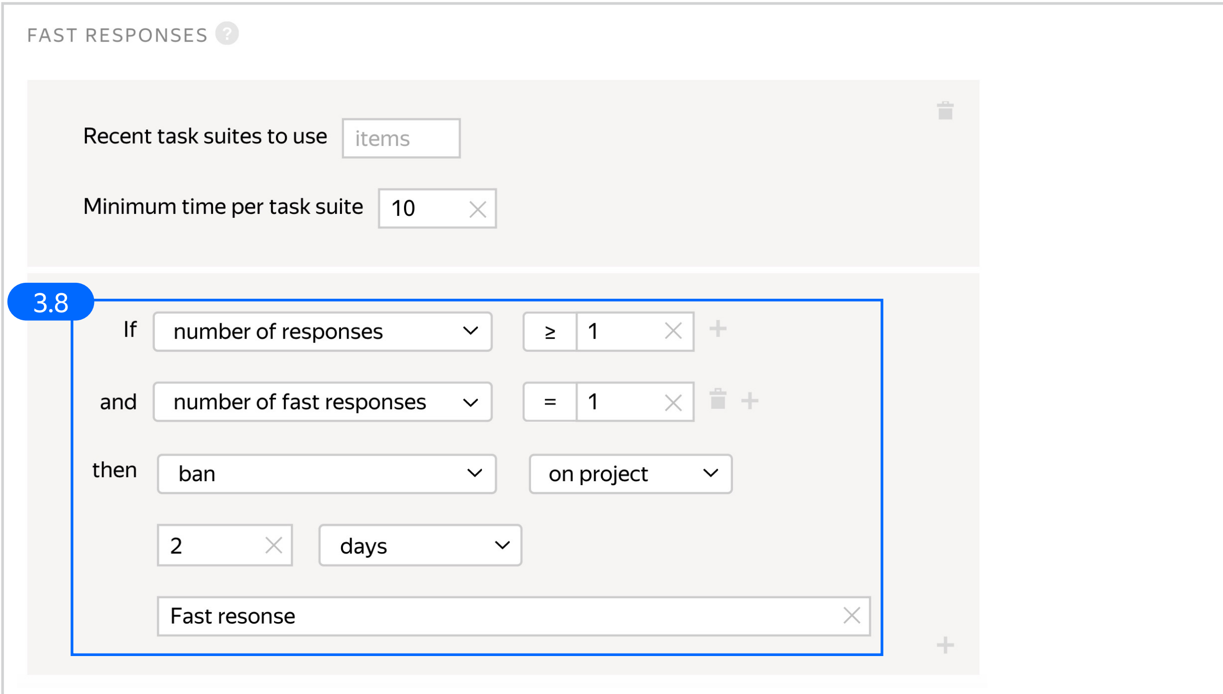
Note: The skill will start calculating after we get at least 3 responses, and will use a history of the 10 last responses. Based on this history, we will record a quality parameter in the same skill we used as a quality filter. If you want to record exam results and current quality in different skills, create a new one, use it for this rule, and also use it as a filter.



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

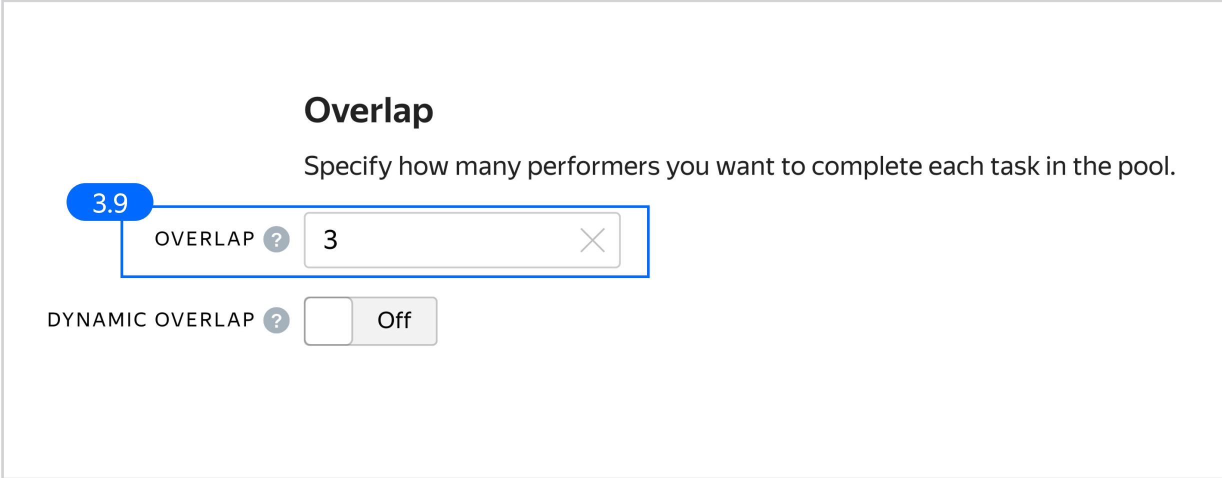
This rule allows you to ban performers who submit tasks at a suspiciously high speed.

Note: In this configuration, a performer will be banned for two days if they give an answer within less than 10 seconds.



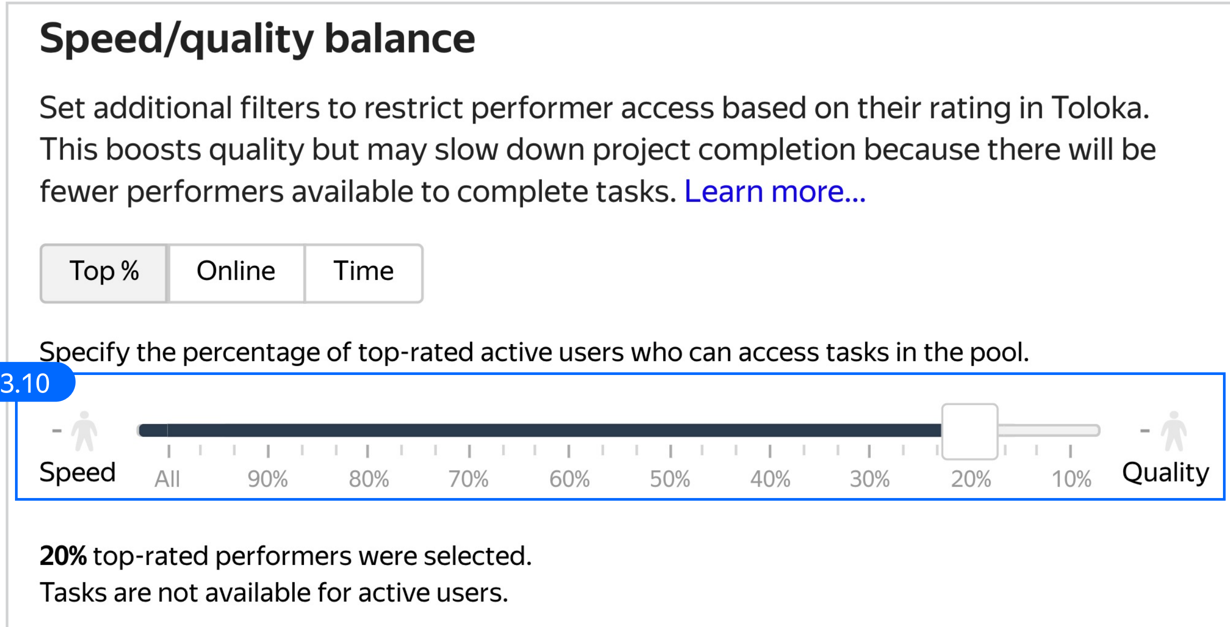
3.9. Set an overlap. This is the number of users who will complete the same task. For example, 3. We will aggregate the results after the pool is completed.

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



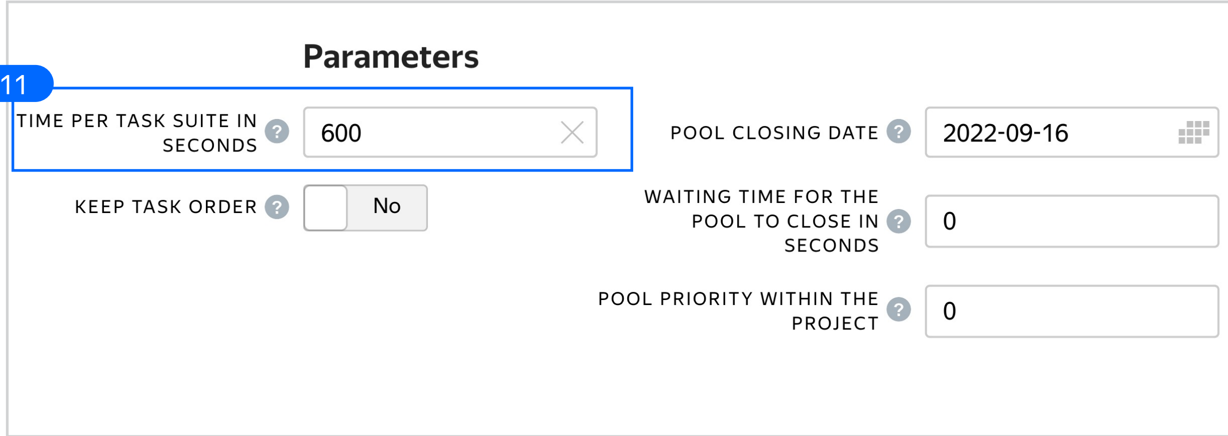
3.10. Optionally, specify the percentage of top-rated performers in the Speed / quality ratio. This is not really necessary though, since we only grant access to performers who already proved their competence in the exam.

Note: This can slow down pool completion.

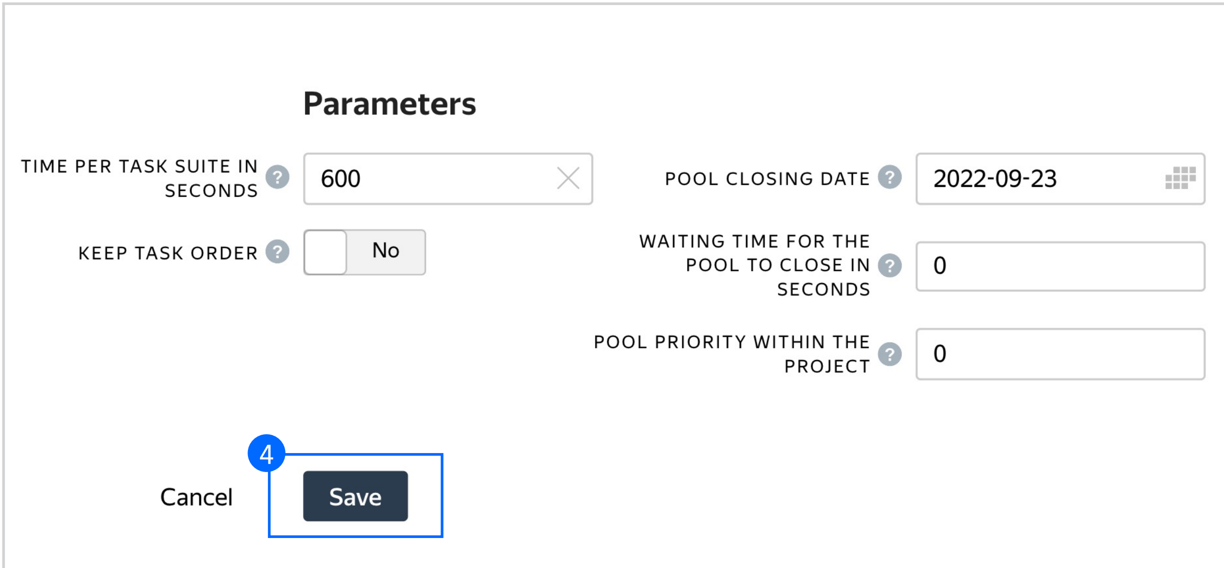


3.11. Specify the time given to complete a task suite (for example, 600 seconds).

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



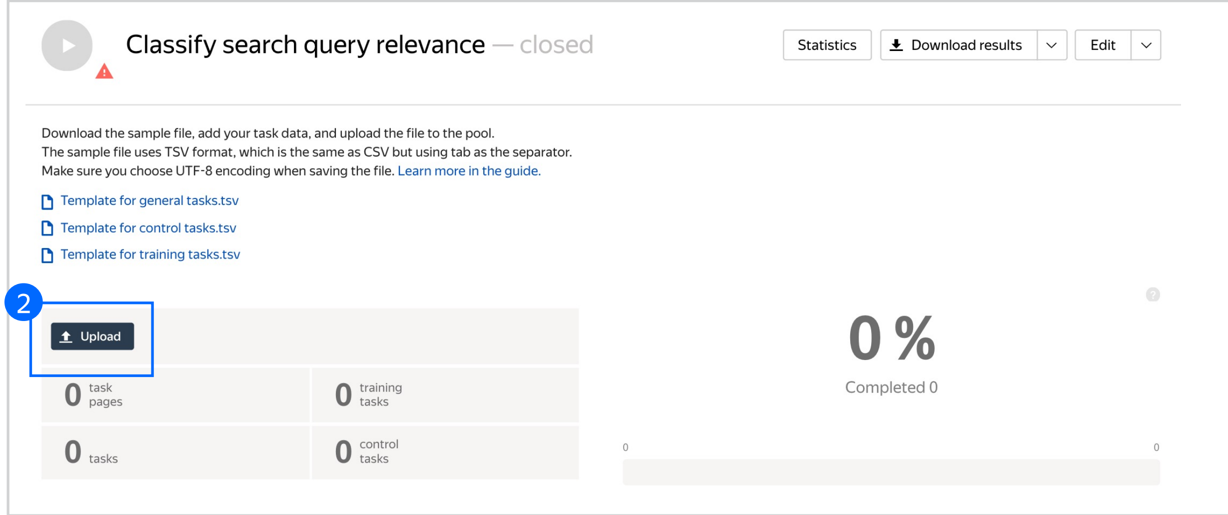
4. Click **Save** to save Pool parameters.



Prepare and upload a file with tasks

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

2. Upload pool tasks from this file.



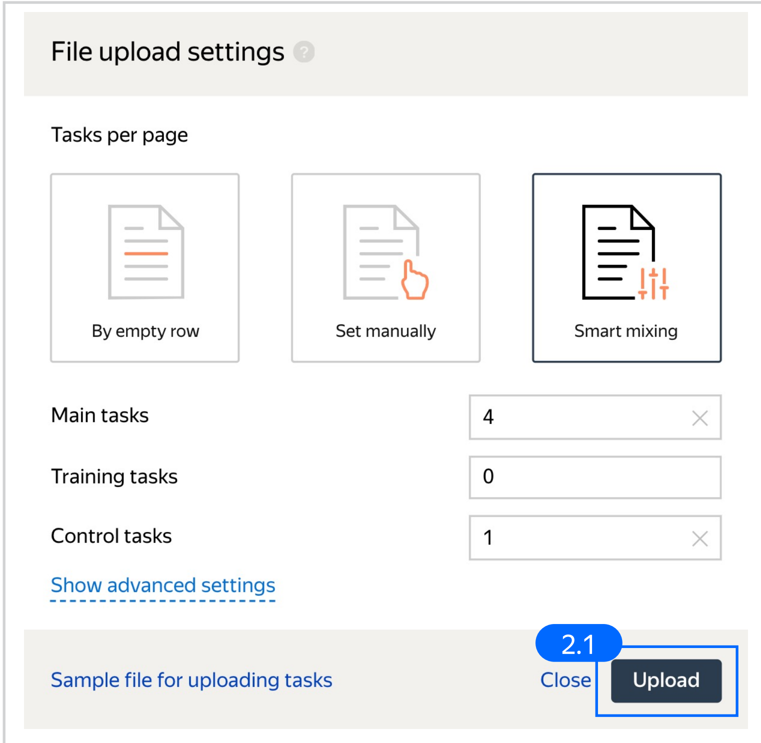
2.1. Select Smart mixing in **File upload settings** and specify the number of tasks of each type per page.

Note: 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.

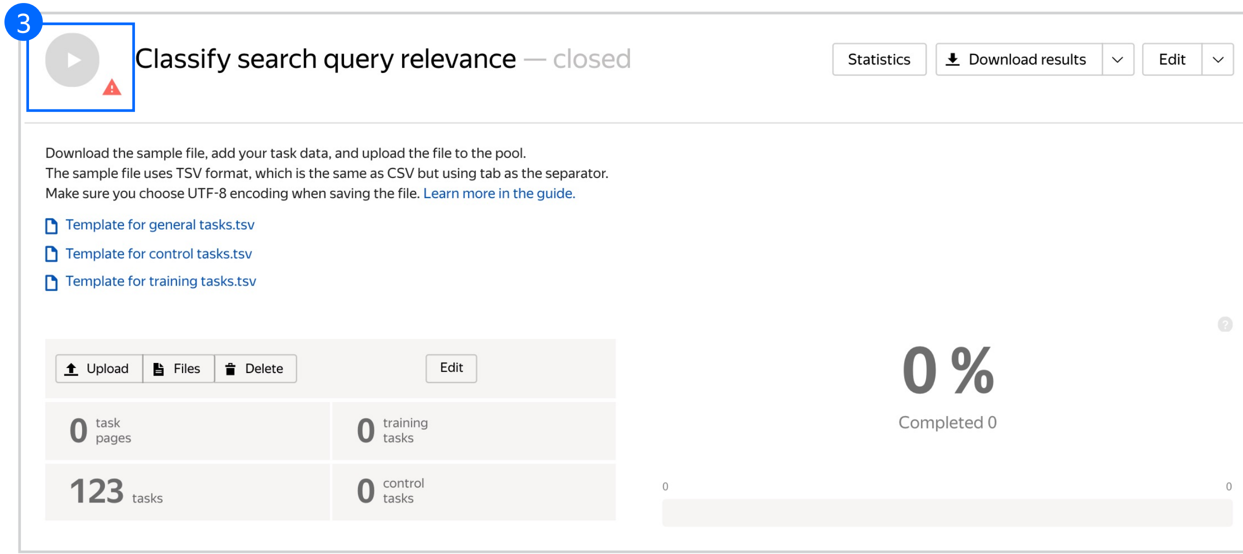
To learn more about grouping tasks into suites, read the Requester's Guide.

Click **Upload** again.

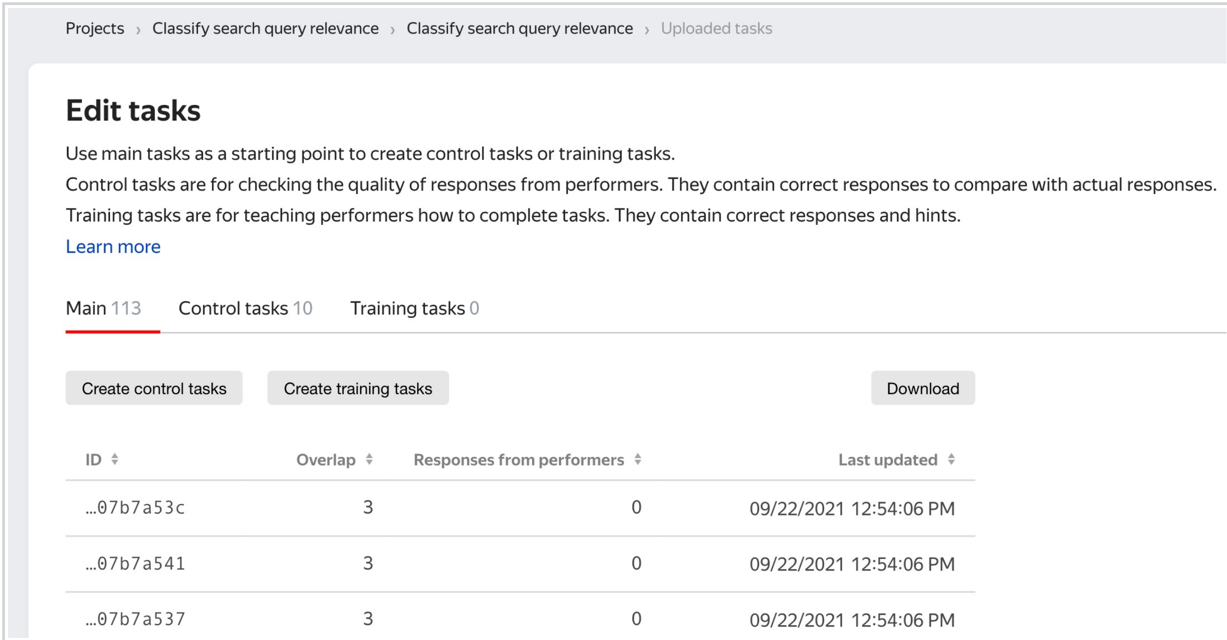
Note: If you changed the name of the input field, change it in the file as well.



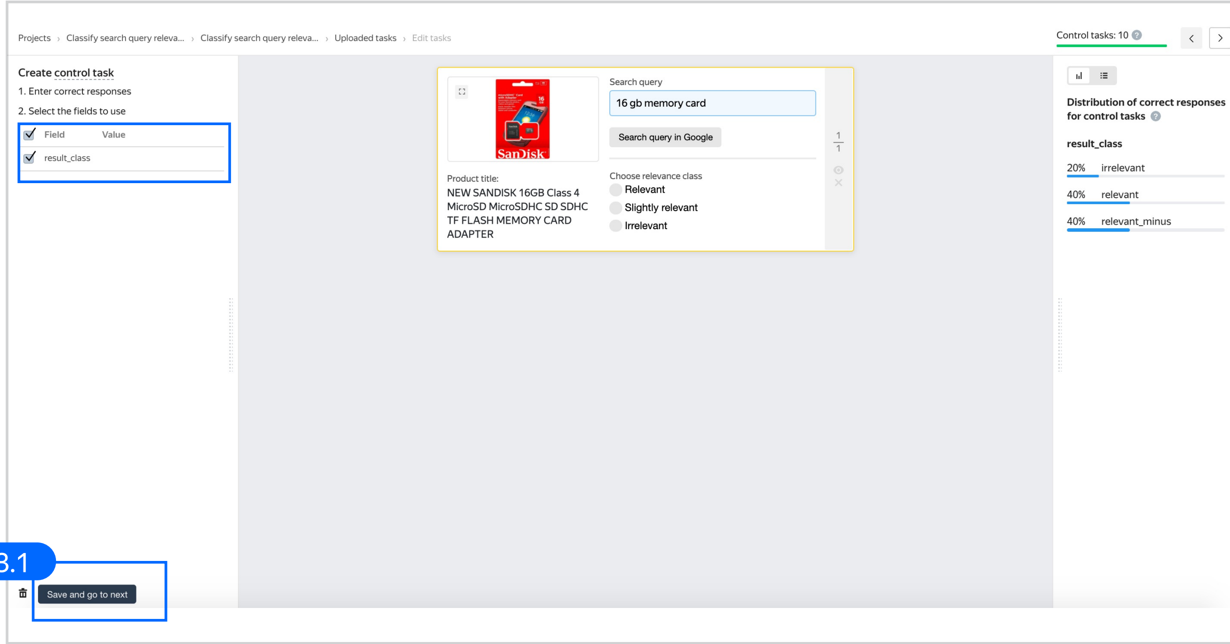
3. [Create control tasks](#).
   Click **Edit → Create control tasks.**



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 to your control tasks. Check the **result_class** output field, which compares the user's response to the control task. Look at the query, select the response, click **Save and go to next**. Repeat until you have ~10 control tasks.



Note: In small pools, control tasks should account for 10-20% of all tasks.

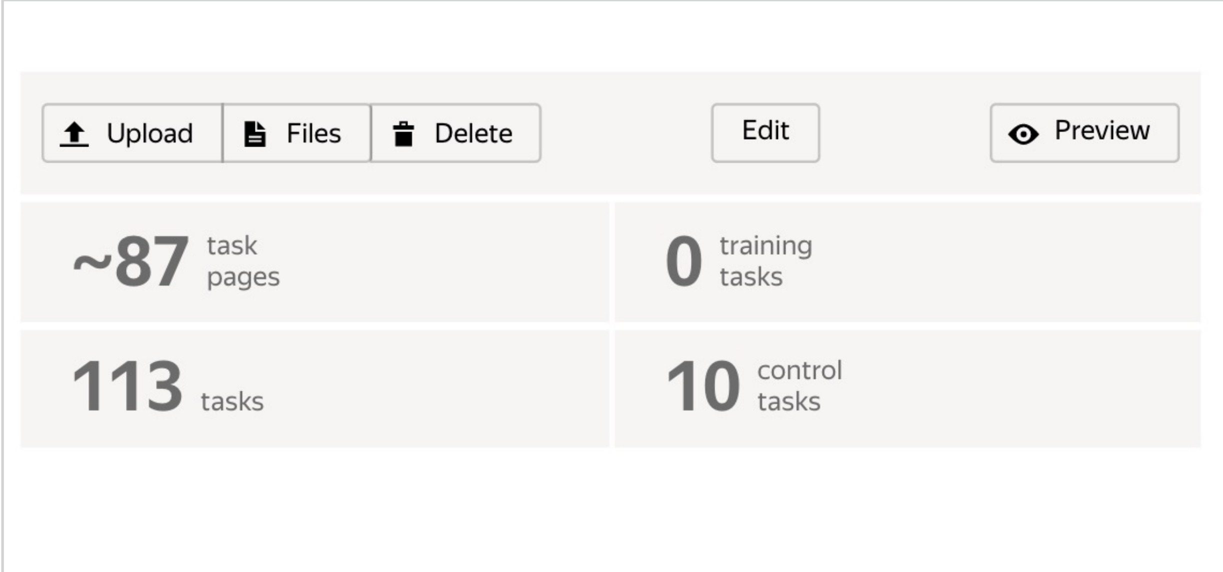
To learn more about [creating control tasks](), go to the Requester's Guide.

Note: You can check the distribution of labels for control tasks in the right panel. We recommend making it even to ensure a fair quality calculation.

Note: Control tasks should be as reliable as possible. It may be a good idea to evaluate control tasks within your team to ensure the guidelines are really clear about these cases, and that all details about the query have been considered.

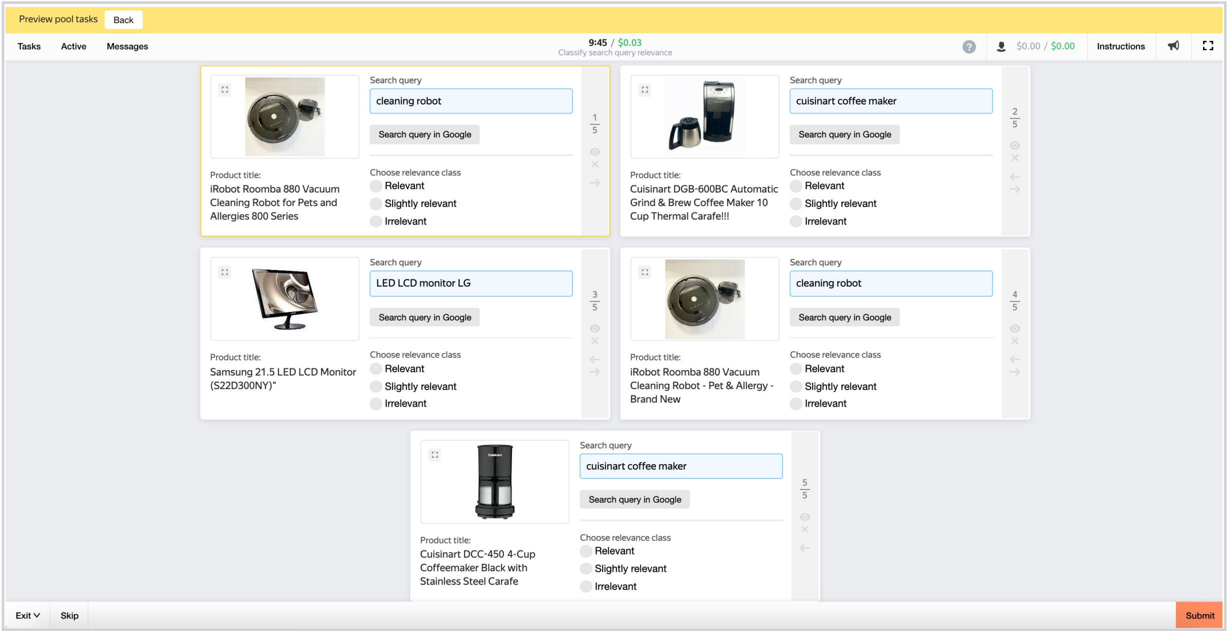
3.2. To check the number of control tasks, go to the **Pool page.**

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.
[Example]
[Origin]
License: PDDL



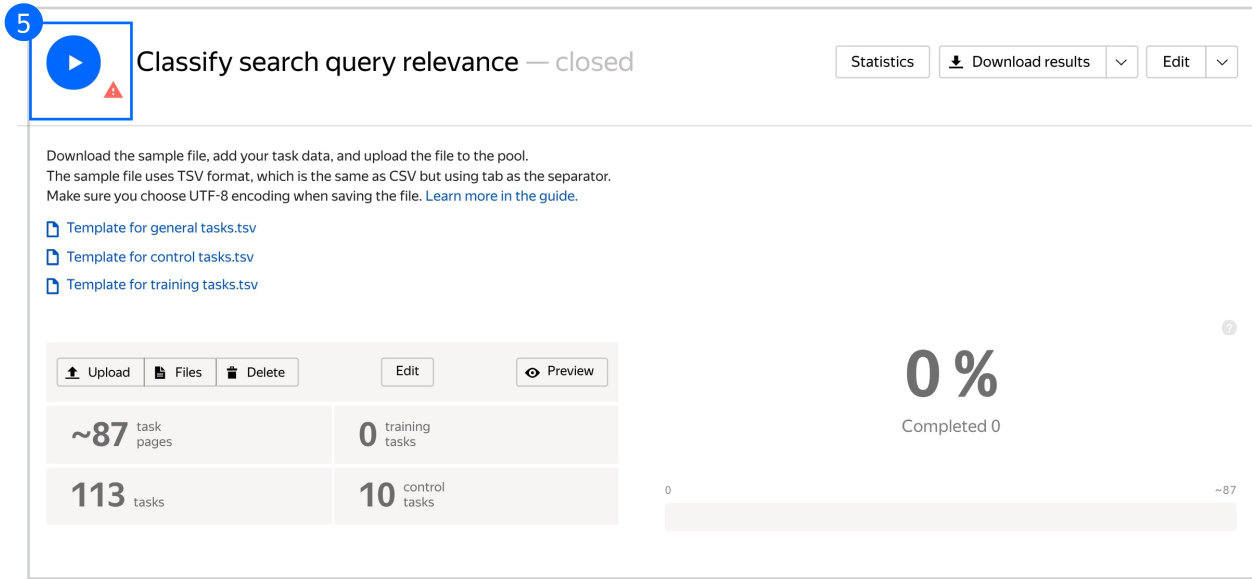
4. Preview the pool.

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.

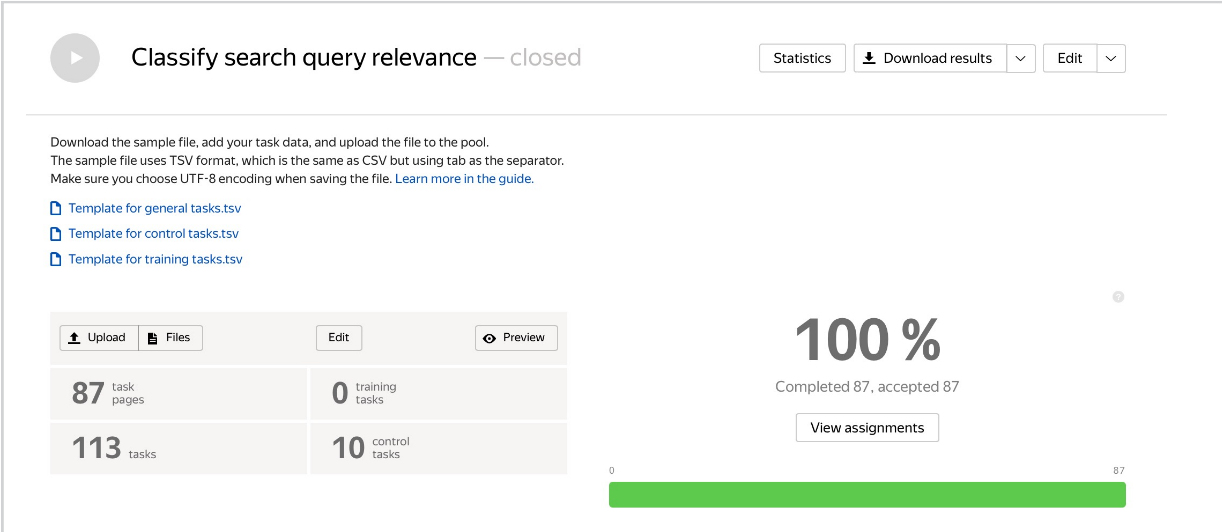
   Make sure you start the training pool and the exam pool before you start the main pool.



Receiving responses

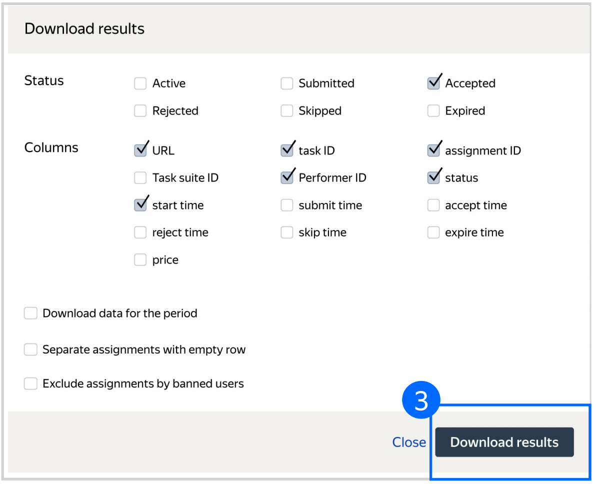
1. Wait until the pool is completed. Refresh the pool page to check progress.

2. Click **Download results.**



Note: Do not forget to close the training pool and the exam pool when there are no main tasks available anymore.

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



Appendix

Interface code
Step 4.1.

```
{
 "view": {
  "type": "view.list",
  "items": [
   {
    "type": "layout.sidebar",
    "minWidth": 400,
    "content": {
     "type": "view.list",
     "size": "m",
     "direction": "vertical",
     "items": [
      {
       "type": "view.image",
       "maxWidth": 350,
       "url": {
        "type": "data.input",
        "path": "imagepath"
       }
      },
      {
       "type": "view.markdown",
       "label": "Product title:",
       "content": {
        "type": "data.input",
        "path": "title"
       }
      }
     ]
    },
    "controls": {
     "type": "view.list",
     "direction": "vertical",
     "items": [
      {
       "type": "view.alert",
       "theme": "info",
       "label": "Search query",
       "content": {
        "type": "view.text",
        "content": {
         "type": "data.input",
         "path": "query"
        }
       }
      },
      {
       "type": "view.action-button",
       "label": "Search query in Google",
       "action": {
        "type": "action.open-link",
        "payload": {
         "type": "data.input",
         "path": "search_url"
        }
       }
      },
      {
       "type": "view.divider"
      },
      {
       "type": "field.radio-group",
       "label": "Choose relevance class",
       "options": [
        {
         "label": "Relevant",
         "value": "relevant"
        },
        {
         "label": "Slightly relevant",
         "value": "relevant_minus"
        },
        {
         "label": "Irrelevant",
         "value": "irrelevant"
        }
       ],
       "data": {
        "type": "data.output",
        "path": "result_class"
       },
       "validation": {
        "type": "condition.required"
       }
      }
     ]
    }
   }
  ]
 },
 "plugins": [
  {
   "type": "plugin.toloka",
   "layout": {
    "kind": "scroll",
    "taskWidth": 600
   }
  }
 ]
}
```