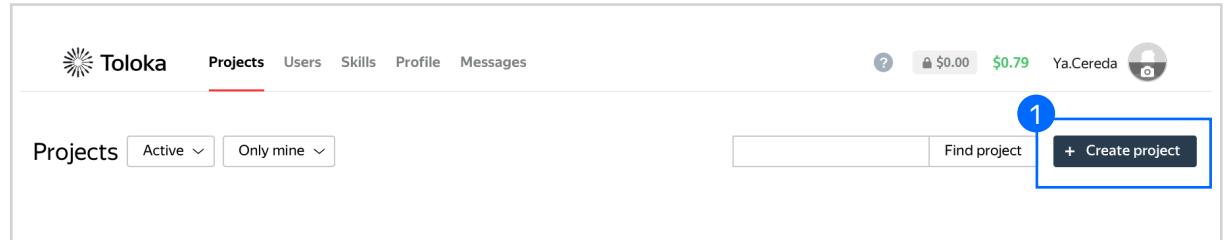




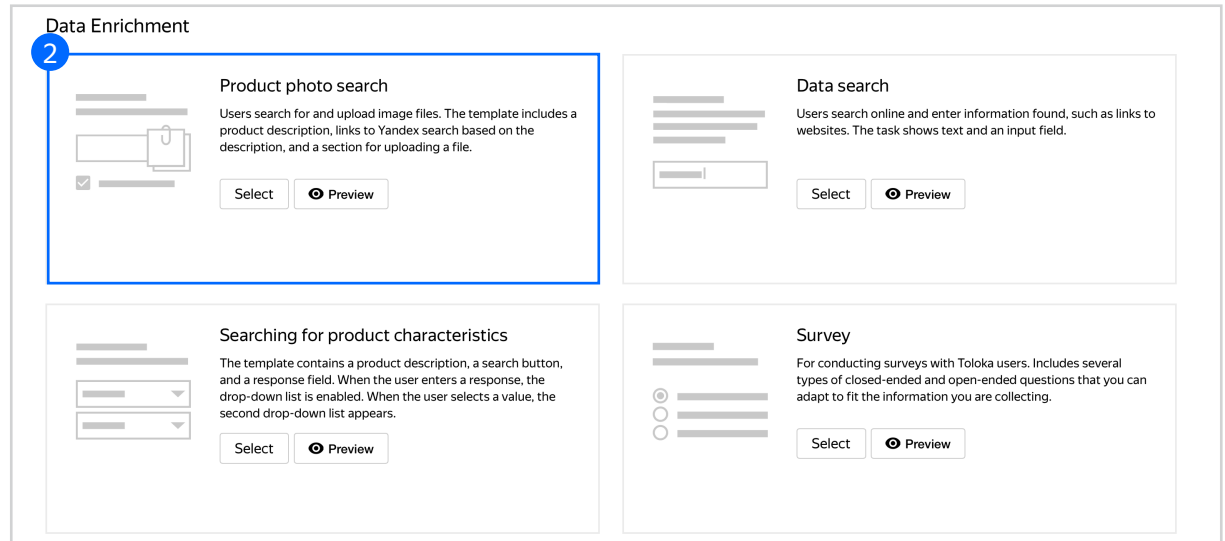
# Image collection manual

# Create a project

1. Click + **Create project**

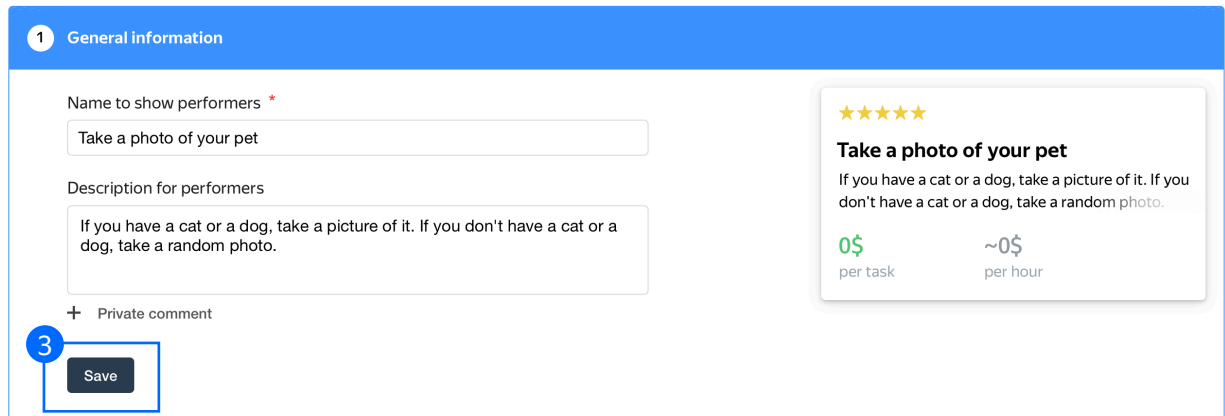


2. Choose the **Product photo search** template

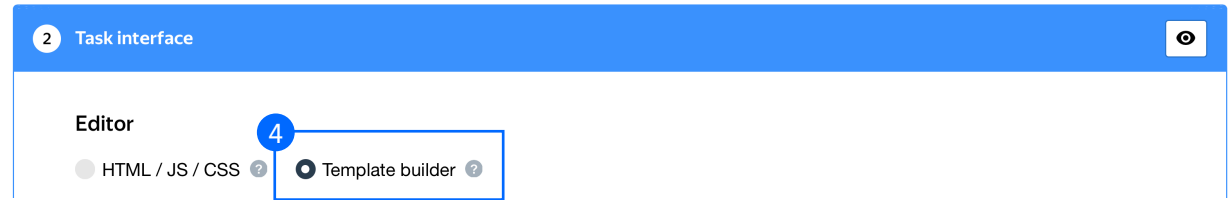


3. Enter a clear project name and description. Click **Save**

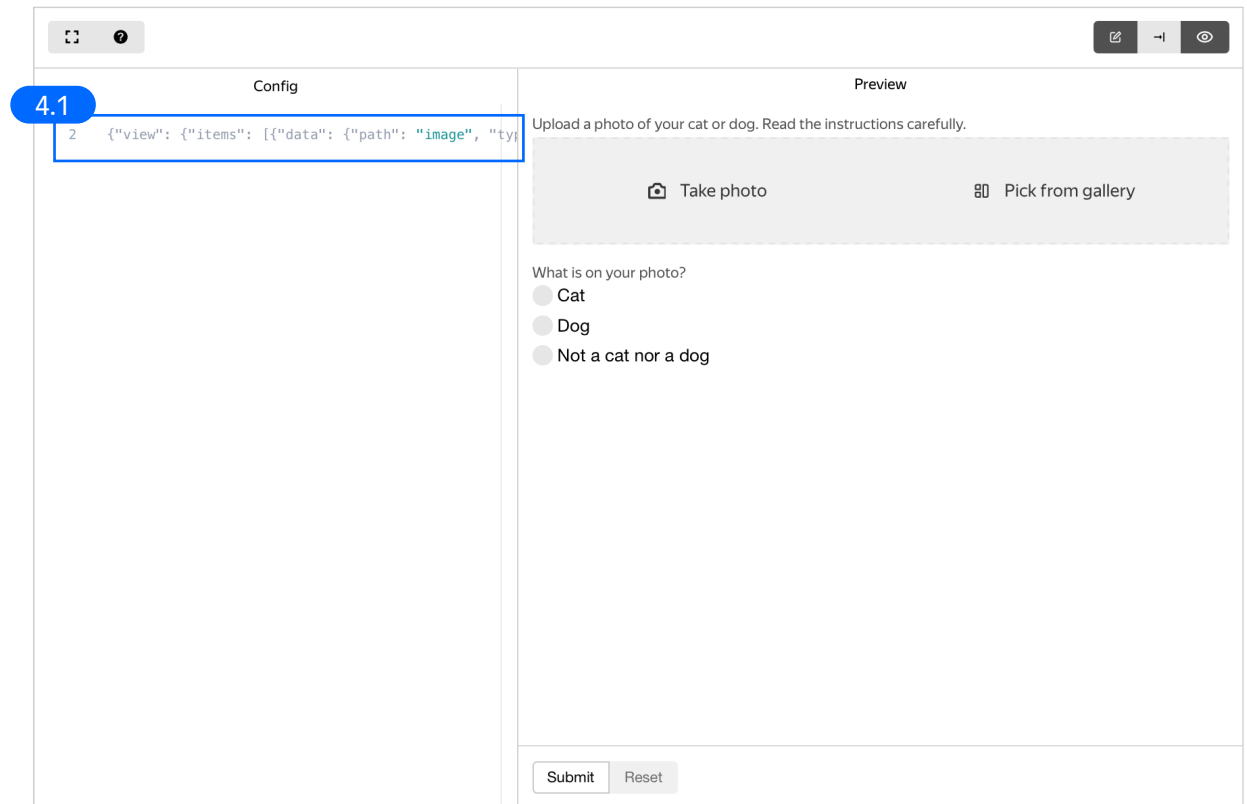
**Note:** The project will be visible to others



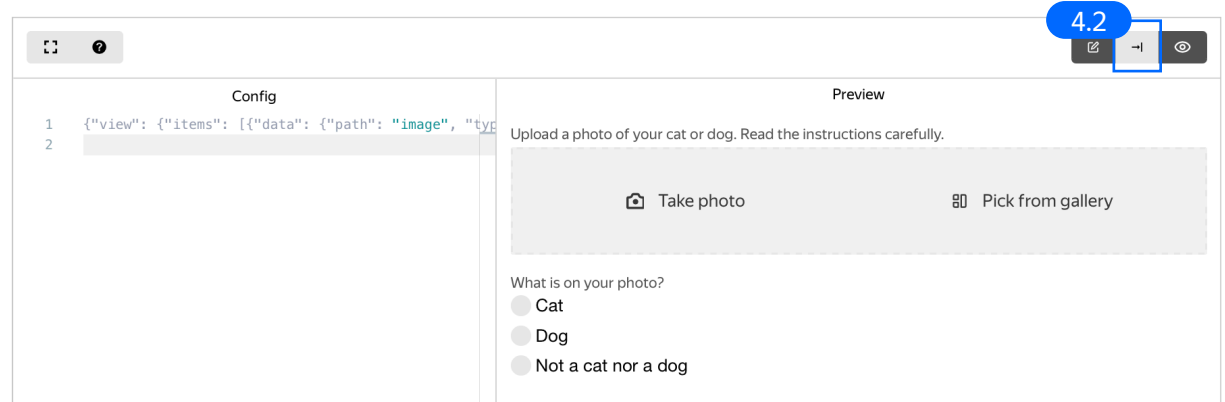
#### 4. Make sure **Template builder** is chosen



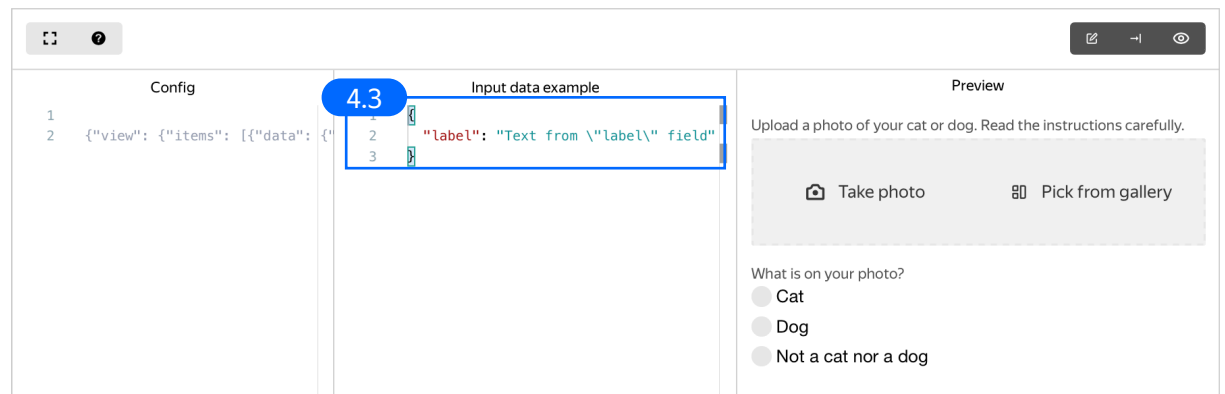
##### 4.1. Delete the existing config and paste the code provided at the end of this manual (in the appendix)



#### 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:  
And click **Save**

### Data specification ?

Input data

label (string) ☐

Add field

Output data <>

image (file) ☒

label (string) ☒

Add field

Show common interface elements

4.4

Save

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

**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](#).

Take a picture of your pet if it is a cat or a dog and select the appropriate label.

If you don't have a cat or dog, take a photo of anything and select "Not a cat nor a dog". There should be exactly one animal in the photo, clearly visible, not cropped. The animal can be photographed from any side and in any position. You can take a picture of a pet in your arms.

It should be clearly visible which animal is depicted: do not photograph your pet's back in the dark.

**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

—

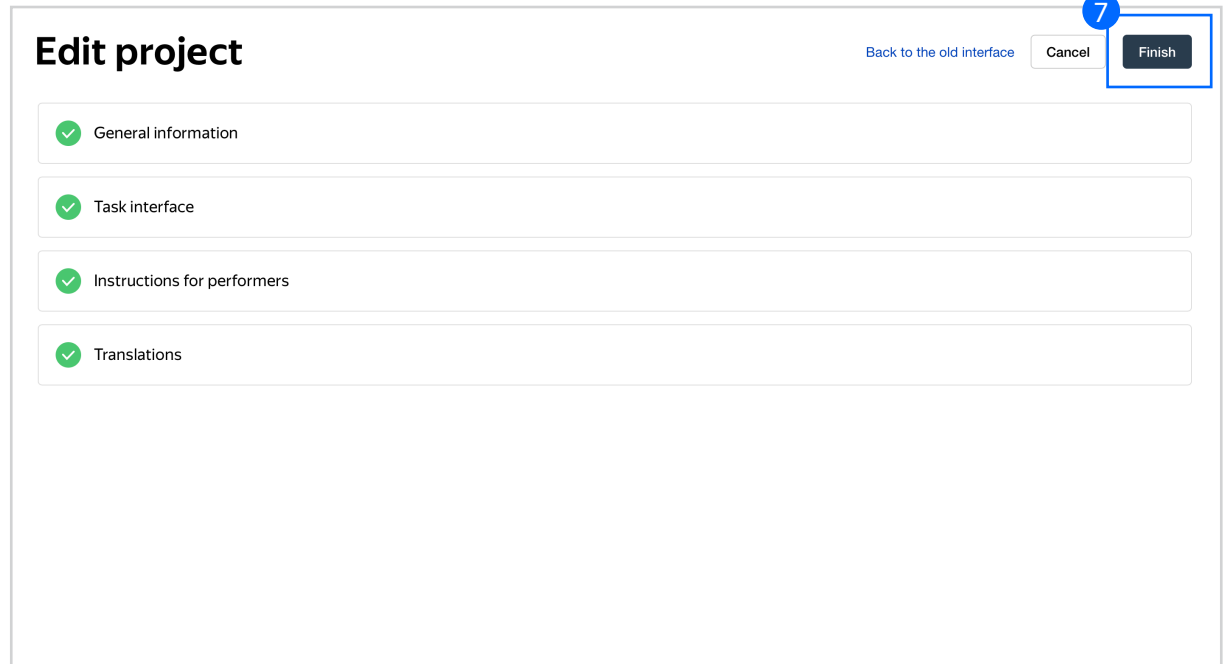
**Translations**

Language	Name and description for performers	Instructions for performers
✓ <b>Source</b>	✓	✓

Add translation

**6** **Save**

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

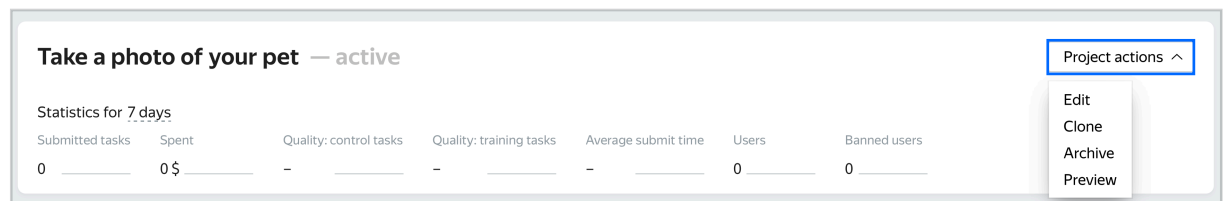


### Edit project

[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



### Take a photo of your pet — active

Statistics for 7 days

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

Project actions ^

- Edit
- Clone
- Archive
- Preview

# Pool creation

1. Click **Add a pool**

Take a photo of your pet — 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

2

POOL NAME (VISIBLE ONLY TO YOU) ?

Take a photo of your pet

☒ Use project description

PUBLIC DESCRIPTION ?

If you have a cat or a dog, take a picture of it. If you don't have a cat or a dog, take a random photo.

☐ Add a private description

- 
3. Specify [pool parameters](#):

- 3.1. Set the price per task suite (for example, \$0.05)

3.1

PRICE IN US DOLLARS ?

0.05

FEE ?

0.01

+ Dynamic pricing

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

**3.2**

### Performers

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](#)

[Copy settings from...](#)

ADULT CONTENT ? ☐ No

Add filter Create a skill

OS major version  
OS minor version  
OS versions  
Performer rating  
Region by IP  
Region by phone number  
Type of client application  
User agent minor version

Performer profile

Adult content  
Citizenship  
City  
Country  
Date of birth  
Education  
Gender  
**Languages**  
Verified

Skills

My skills  
Add filter

3.3. Select English-speaking performers using the **Language = English** filter

### Performers

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](#)

[Copy settings from...](#)

ADULT CONTENT ? ☐ No

Add filter Create a skill

**3.3**

PERFORMER PROFILE

Languages = English X trash +

- 3.4. Click **Add filter** to choose the **Client** option in the list.  
Make sure the task is displayed only to mobile users: use the **Client=Toloka for mobile** filter.

The screenshot shows the 'PERFORMER PROFILE' and 'CALCULATED DATA' sections of the Toloka skill configuration interface. In the 'PERFORMER PROFILE' section, the 'Languages' filter is set to 'English'. In the 'CALCULATED DATA' section, the 'Client' filter is set to 'Toloka for mobile'. A blue callout bubble with the number '3.4' points to the 'Client' filter dropdown. To the right, a list of available filters is shown, with 'Client' highlighted. The list includes: Browser, Bugfix version of the operating system, Client, Client application bugfix version, Client application major version, Client application versions, Device type, Operating system, OS major version, OS minor version, OS versions, Performer rating, Region by IP, Region by phone number, and Type of client application.

- 3.5. Create a skill. It will be assigned to users after they complete the pool tasks. Click **Create a skill**

The screenshot shows the 'Performers' section of the Toloka skill configuration interface. It includes a link 'Copy settings from...' and a description: '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](#)'. Below this, there is an 'ADULT CONTENT' toggle set to 'No'. At the bottom, there is an 'Add filter' dropdown and a 'Create a skill' button. A blue callout bubble with the number '3.5' points to the 'Create a skill' button.

- 3.6. Enter the skill name and add a description if needed. You are the only one who will see it. Leave the skill private (as default) and click **Add**

so don't forget to filter by language and region. [Learn more](#)

No Add skill

TITLE

3.6 pet\_photo\_received

DESCRIPTION

PERF

La

CALC

Click

Public? ☐ No ?

Cancel Add

Quality center

3.7. Set up [Quality control](#). 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 ? None

3.7

+ Add Quality Control Rule

3.8. Set up the **Submitted responses**  
quality control rule. When a person  
submits 1 or more tasks, they are  
assigned the skill created in the step  
above

### SUBMITTED RESPONSES ?

3.8

If submitted task suites > 0 +

then assign skill value pet\_photo\_recei 1 +

3.9. Overlap. This is the number of users who will complete the same task. For example, 1

### Overlap

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

3.9

OVERLAP ? 1

DYNAMIC OVERLAP ? ☐ Off

3.10. Time given to complete a task suite (for example, 600 seconds)

### Parameters

3.10

TIME PER TASK SUITE IN SECONDS ? 600

KEEP TASK ORDER ? ☐ No

POL CLOSING DATE ? 2022-06-03

WAITING TIME FOR THE POOL TO CLOSE IN SECONDS ? 0

POOL PRIORITY WITHIN THE PROJECT ? 0

- Click **Save** to save Pool parameters

### Parameters

TIME PER TASK SUITE IN SECONDS ?	<input type="text" value="600"/>	POOL CLOSING DATE ?	<input type="text" value="2022-06-07"/>
KEEP TASK ORDER ?	<input type="checkbox"/> No	WAITING TIME FOR THE POOL TO CLOSE IN SECONDS ?	<input type="text" value="0"/>
		POOL PRIORITY WITHIN THE PROJECT ?	<input type="text" value="0"/>

Cancel4 Save

- Preview the pool

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

4:02 / \$0.05

Instructions

1 / 1

Upload a photo of your cat or your dog. Read the instructions carefully.

Take photo

Pick from gallery

What is on your photo?

Cat

Dog

Not a cat nor a dog

Submit

## Preparing and uploading a file with tasks

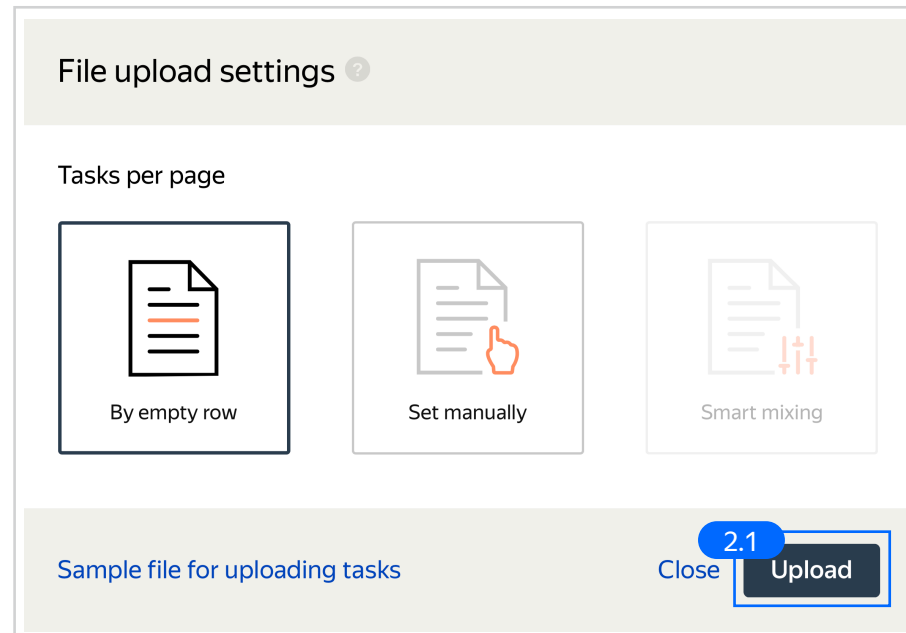
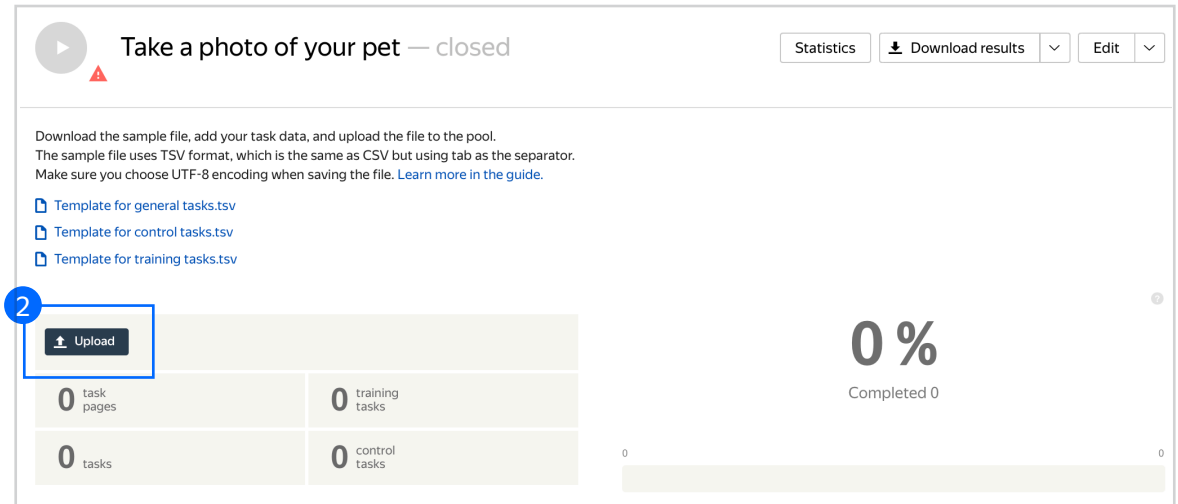
1. Prepare a TSV file for the tasks. The name of the input field should correspond to the data specifications selected in the project - INPUT:label

**Note:** It is necessary to upload a task file even though no input will be shown in the task interface

2. Upload this file

- 2.1. Select [Empty row](#) in **File upload settings**. Click **Upload**

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





2.2. In the pop-up window, click **Add** to add tasks to the pool

Adding tasks to pool (image\_collection.tsv)

TASKS FOR POOL

5 task pages	5 tasks
0 training tasks	0 control tasks

2.2

Cancel Add

3. Start the pool

3

Take a photo of your pet — 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

5 task pages	0 training tasks
5 tasks	0 control tasks

0 %  
Completed 0

0 5

## 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

Take a photo of your pet — 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](#)

Upload Files Preview

5 task pages	0 training tasks
5 tasks	0 control tasks

100 %  
Completed 5, accepted 5

View assignments

Download results

Status

- ☐ Active
- ☐ Submitted
- ☒ Accepted
- ☐ Rejected
- ☐ Skipped
- ☐ Expired

Columns

- ☒ URL
- ☒ assignment ID
- ☐ Task suite ID
- ☒ Performer ID
- ☒ status
- ☒ start time
- ☐ submit time
- ☐ accept time
- ☐ reject time
- ☐ skip time
- ☐ expire time
- ☐ price
- ☐ task suite ID

☐ Download data for the period

☐ Separate assignments with empty row

☐ Exclude assignments by banned users

Close Download results

## Appendix

Interface code

Step 4.1.

```
{"view": {"items": [{"data": {"path": "image", "type":  
"data.output"}, "label": "Upload a photo of your cat or  
dog. Read the instructions carefully.", "validation":  
{"type": "condition.required"}, "accept": {"gallery":  
true, "photo": true}, "multiple": false, "type":  
"field.media-file"}, {"data": {"path": "label", "type":  
"data.output"}, "label": "What is on your photo?",  
"validation": {"type": "condition.required"}, "options":  
[{"label": "Cat", "value": "cat"}, {"label": "Dog",  
"value": "dog"}, {"label": "Not a cat nor a dog",  
"value": "none"}], "type": "field.radio-group"}],  
"type": "view.list"}}
```

Input data example

Step 4.3.

```
{  
"label": "Text from \"label\" field"  
}
```