A plan for PyBossa

How to set up a crowdsourcing project

Jaka Čibej

Centre for Language Resources and Technologies, University of Ljubljana

Ljubljana, 6 December 2018

Univerza *v Ljubljani*

cost
EUROPEAN COOPERATION
IN SCIENCE & TECHNOLOGY

cjvt

Introduction

- During the workshop, we will discuss:
  - how to plan a crowdsourcing project for PyBossa,
  - how to prepare the data,
  - what pitfalls to keep in mind,
  - when to include technical help (and what to expect from it).

A plan for PyBossa

Overview

- Part 1: Is the project idea even suitable for crowdsourcing?

- Part 2: What should the crowdsourcing task look like?

- Part 3: Local PyBossa installation vs. Crowdcrafting.org

- Part 4: Workflow – Preparing the data.

- Part 5: Hands-on experiment: Setting up a task in PyBossa.

- Part 6: Testing and evaluating the task.

Part 1: You get an idea for a crowdsourcing task.

- Is the idea suitable for **crowdsourcing** or is it enough to do a **survey** (e.g. with Google Forms or 1ka)?

Crowdsourcing	Survey
large number of tasks	**limited** number of questions
participants are **not** (necessarily) required to solve **every task**	participants must answer **every question** in a specified order
solve tasks – **take a break** – solve more	answer everything **at once**
no metadata (?) – in general, we don't care who solves what **Exception**: controlled experiments (e.g. only linguists)	systematic collection of **metadata**
(long) open-ended questions **discouraged**	may include (long) **open-ended questions**
no way to measure reliability of responses + unknown participants = unreliable results (?)	less prone to abuse (?)

A plan for PyBossa

Part 1: You get an idea for a crowdsourcing task.

- You decide to go with **crowdsourcing**. Is the task you envisioned suitable?

- **SIMPLICITY**
    - as simple as possible!
    - (ideally) **no typing input** → stick to **clicking**, **drag-and-drop**, etc.
    - short response time!
    - clearcut response options (YES, NO, I DON'T KNOW)

Part 1: You get an idea for a crowdsourcing task.

- Example 1: Is the collocation below to be included in dictionaries?

cjvt

Part 1: You get an idea for a crowdsourcing task.

- Example 2: Which comments below a Facebook post are offensive and to whom?

Does the UK Government need to be taking more drastic measures to bring down immigration?

UK net migration soared by 20% last year

Ni relevantno

yes because our NHS is ruined now	Ni sporni govor ▾	- ▾
Yes for the most umpteenth time. Why is it that the Daily Mail actually supports the Conservatives when they should know that they are not going to do anything?	Ni sporni govor ▾	- ▾
of course they do!	Ni sporni govor ▾	- ▾
of course they should, the sooner the better. then perhaps people born in this country who are on the housing list won't keep being pushed to the back of the queue by all the immigrants.	Ni sporni govor ▾	- ▾

Part 1: You get an idea for a crowdsourcing task.

- Example 3: Which accent is your first choice in modern standard Slovene?
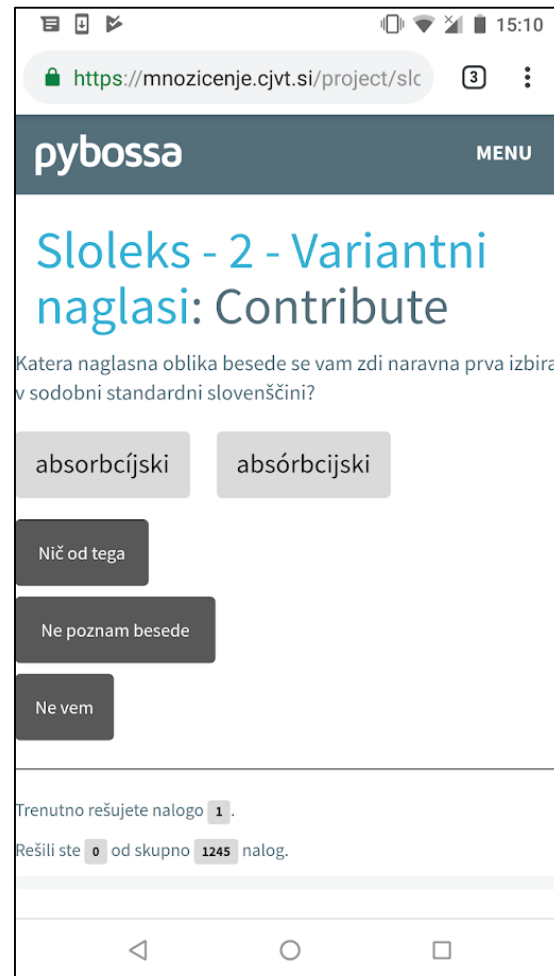
A plan for PyBossa

Part 1: You get an idea for a crowdsourcing task.

- **DATA**
  - What exactly is the input data?
  - What do you want the output to be?
  - How are you going to use it?
  - **Plan ahead!**

cjvt

Part 1: You get an idea for a crowdsourcing task.

- **DATA**
  - What exactly is the input data?
  - What do you want the output to be?
  - How are you going to use it?
  - **Plan ahead!**

  - **Example:** ranking verbs and adverbs
    - input data: expressions
      - J K L M
    - output data: expressions with ranks
      - J → 2, K → 1, L → 3, M → 2
    - use:
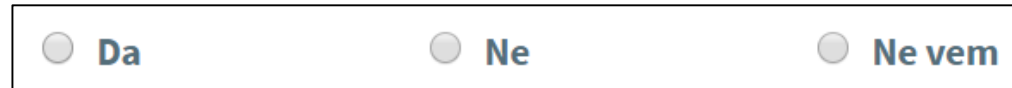   - add up the ranks to create a linear scale

Part 2: What should the task look like?
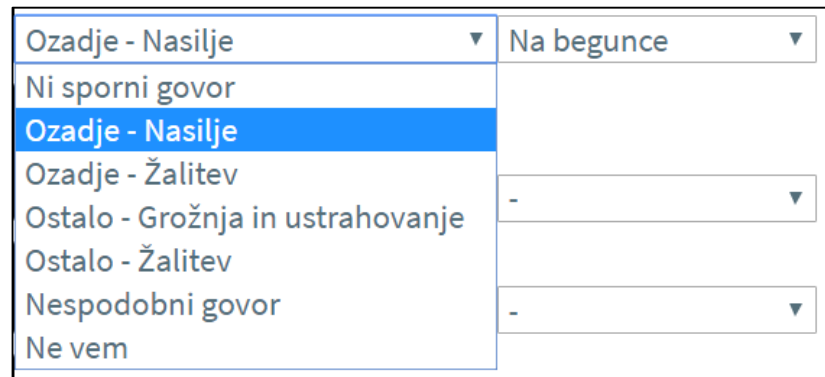
- **VISUALIZATION**
    - How should the task be visualized?
    - What elements to show?
    - Do you use buttons, radio buttons, dropdown lists …?
        - good choices reduce the number of clicks!

A plan for PyBossa

Part 2: What should the task look like?

- **VARIABLE vs. INVARIABLE PARTS OF THE TASK**
  - invariable parts → FIXED position on the screen!
  - variable parts → should not interfere with invariable parts!

Part 2: What should the task look like?

- **PHONE COMPATIBILITY**
  - is the interface too wide/long for the telephone screen?
  - reduce the need for additional scrolling!

cjvt

A plan for PyBossa
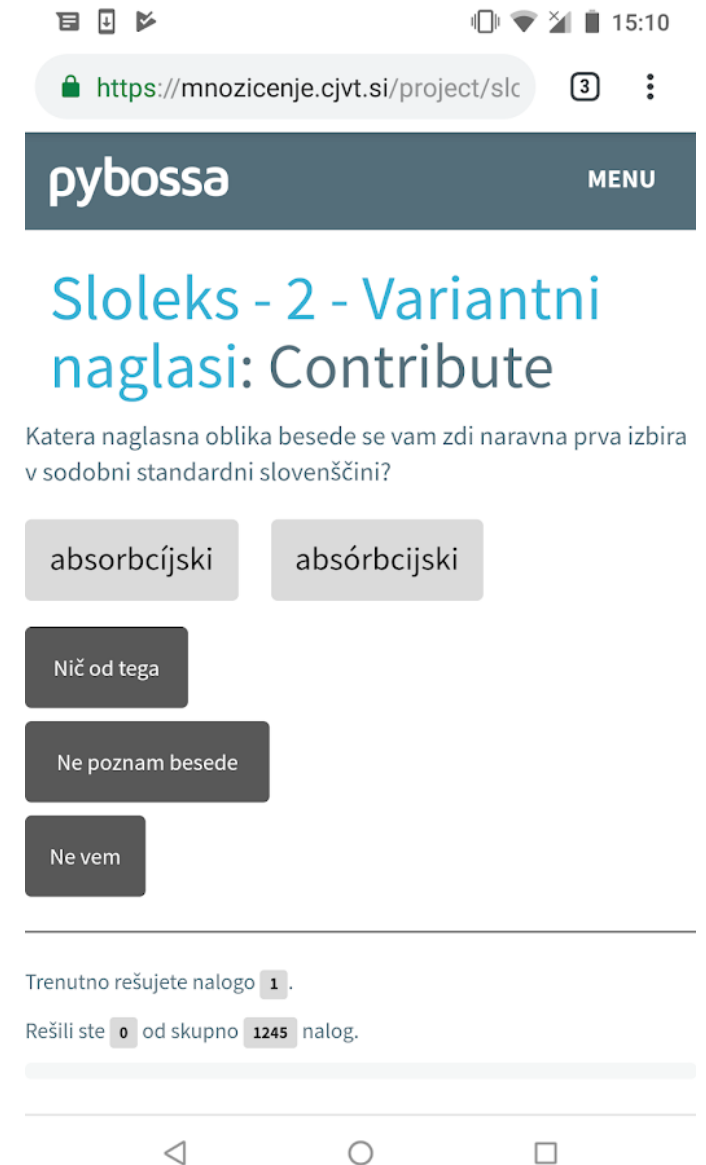
Part 2: What should the task look like?

- **ONE STRUCTURE PER PROJECT**
    - avoid mixing radically different tasks in the same project
    - Example: collocations:
        - one project for *adjective + noun*
        - one project for *adverb + adjective* …

Part 3: Local PyBossa vs. Crowdcrafting.org

- **Crowdcrafting.org** is a public installation of PyBossa

cjvt
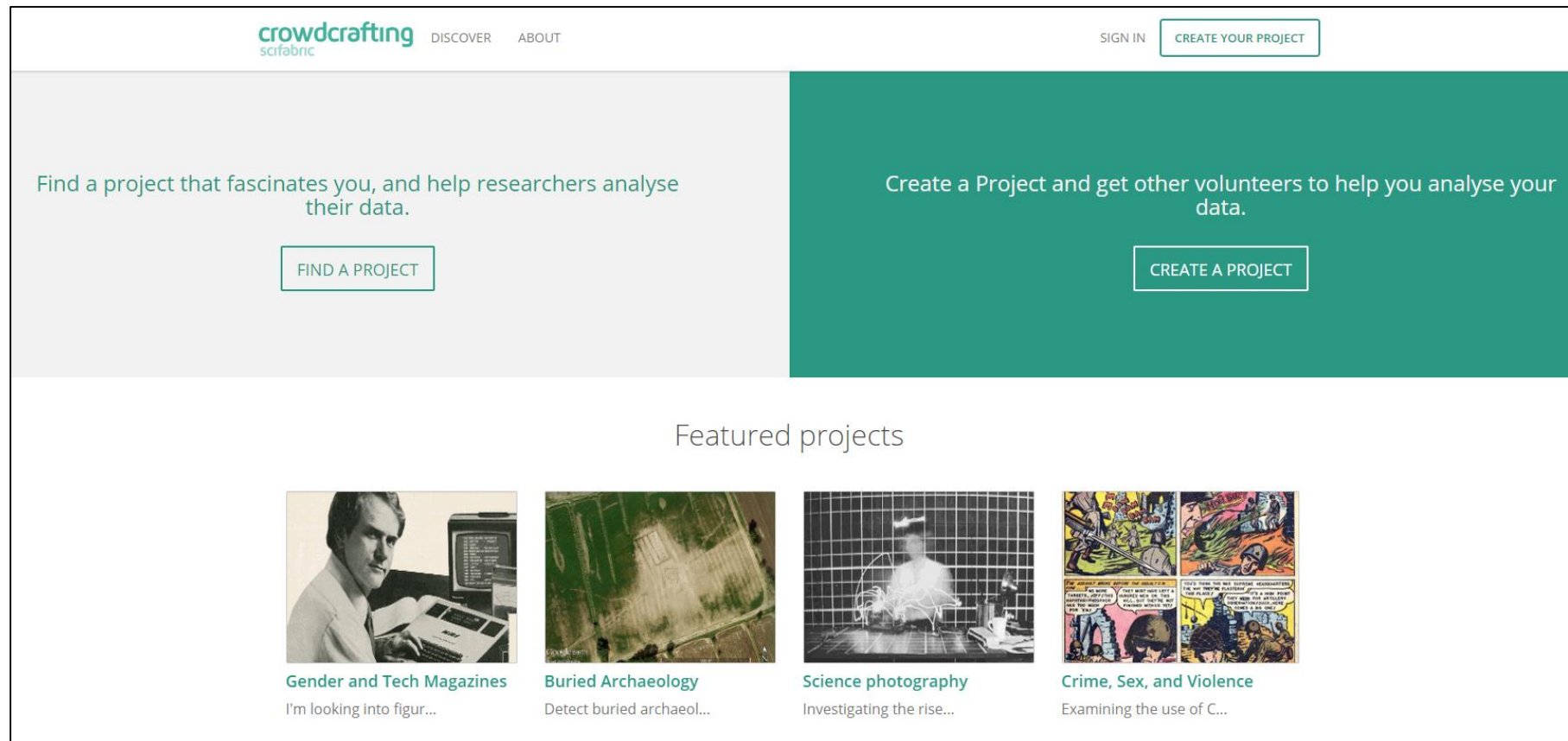
Part 3: Local PyBossa vs. Crowdcrafting.org

- **Local PyBossa vs. Crowdcrafting.org**

Local PyBossa	Crowdcrafting.org
Fully compatible	Partially compatible
More technical help	Less technical help (but still some!)
Participants you disseminate the platform to	Participants from other projects
Projects can be made private	All projects are public (problematic with personal information or intellectual property!)

A plan for PyBossa

Part 4: Workflow

A plan for PyBossa
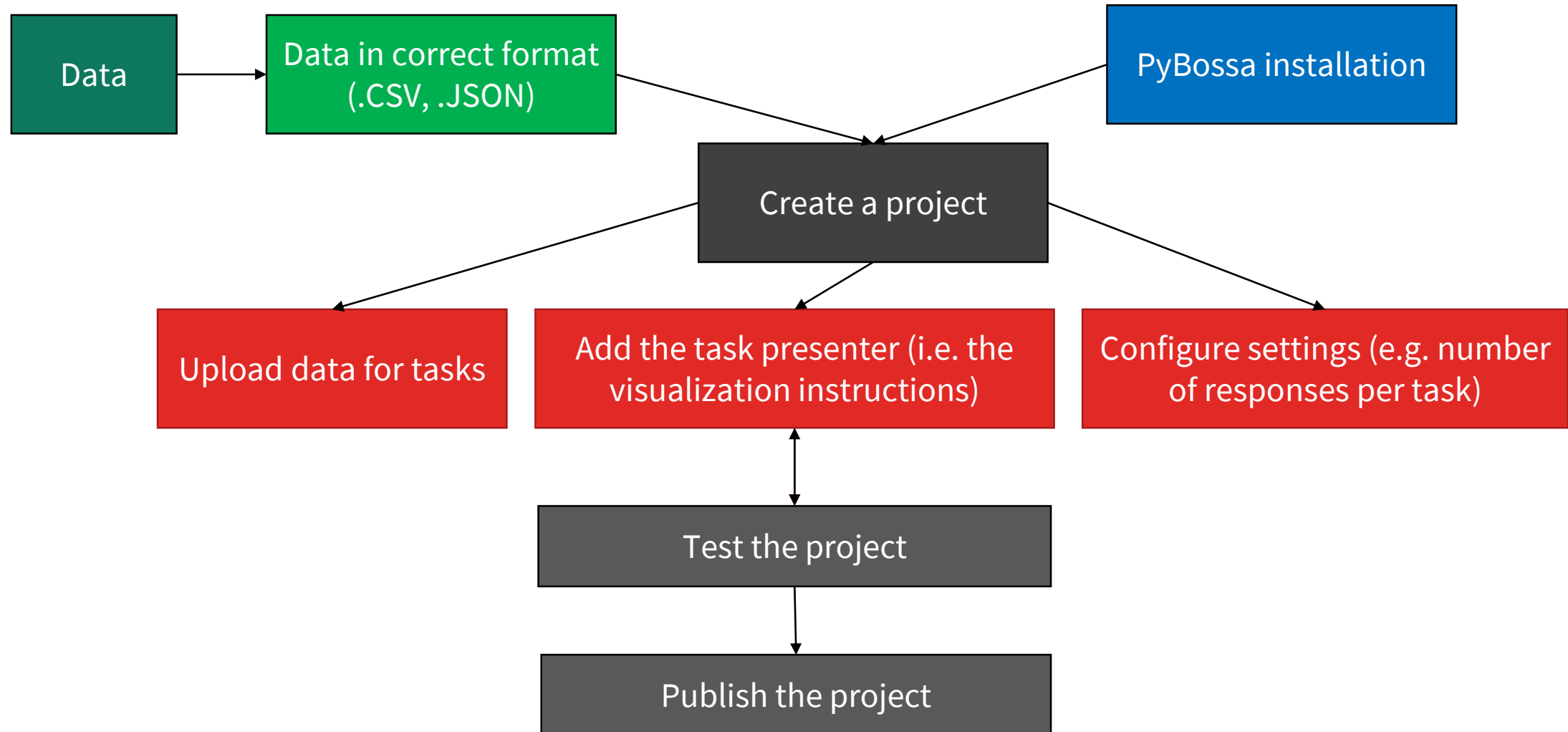
Part 5: Hands-on experiment

We will learn how to convert this:

collocation		collocation_example	task_id
inflict pain	1	I do it because at least you can inflict pain that way on somebody, in terms of legal fees and other things.	1
inflict damage	2	More serious are crushing bites on the limbs, which can be painful and can inflict damage when applied to the stifles or other joints.	2
inflict injury	3	Thus, wizards were largely free to despoil and defame each other, even to inflict injury or exile upon a rival, so long as they did not attract the attention of the government.	3
inflict more	4	Each day he demonstrates his incapacity to do anything but inflict more damage.	4
persecute Christians	5	The communist government continues to persecute Christians, even outlawing bibles by athletes at this summer's Olympic Games.	5
persecute religious	6	He has made open appeals to racism and he has tried to persecute religious minorities.	6
persecute people	7	Our goal here is not to persecute people. Our goal is to enforce laws that are important to protect the public.	7
commit suicide	8	I was kind of thinking, either I do something about this or I commit suicide.	8
commit crimes	9	When people commit crimes or harass others, they should always be held accountable.	9
commit murder	10	Walker is charged with two counts of assault with intent to commit murder and one count of first-degree home invasion.	10

A plan for PyBossa

Part 5: Hands-on experiment

… into this:

Part 5: Hands-on experiment – Creating a project

- Go to **PyBossa (mnozicenje.cjvt.si)**

- Log in by clicking **Sign in** (top right corner)

- Create a project by clicking **Create** (top menu)

A plan for PyBossa
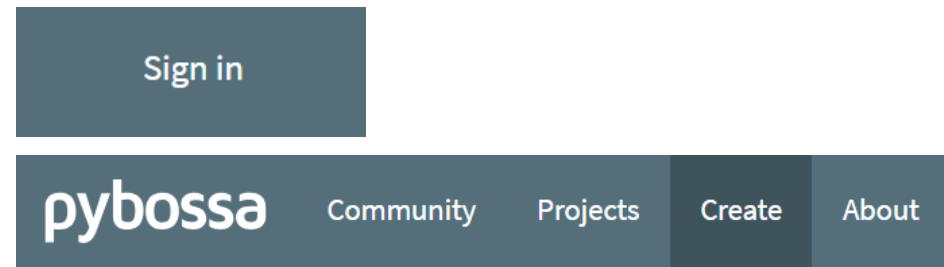
Part 5: Hands-on experiment – Creating a project

- Fill in the information:
  - Project name
    - e.g. Some Collocations
  - Project short name
    - e.g. some_collocations
    - **lower case letters!**
    - **no blank spaces!**
    - important later, **make a note of it**!

  - Long description
    - Describe what the project is about

- Once you're done, click **Create**.

Project: Create a new project

Name

The name of the project

Short Name

Short name or slug for the project

Long Description

Explain your project
· Why? Why you are creating this project, what you want to achieve with it
· How? Explain how your project works
· How you can help: Explain to the volunteers how they can help and contribute to the goal of your project
· Will you help make a difference? Explain to the volunteers why their contribution is essential and how the result of your project can help change things!

(you can use Markdown to give format to your description!)

Create   Cancel

Part 5: Hands-on experiment – Creating a project

- Configure initial settings
    - Choose the **Category** (the project will be shown under the chosen tab on the first page)
    - Example: by choosing the category **eNetCollect**, the project will be shown in the eNetCollect tab.

    - Tick **Allow Anonymous Contributors** if you want to allow unregistered users to contribute to the project.

Category

eNetCollect   ▾

☐ Allow Anonymous Contributors
☐ Protect with a password?

KSSJ   KOLOS   FRENK   Sopomenke   Arhiv   eNetCollect   Sloleks   Draft

eNetCollect Projects

eNetCollect crowdsourcing tasks

eNetCollect – WG1

Ranking English adverbial MWEs
(for native speakers of English)

Ranking English adverbial MWEs (for native speakers)

For native speakers of English only!

MORE INFO

eNetCollect – WG1

Ranking English verbal MWEs
(for native speakers of English)

Ranking English verbal MWEs (for native speakers)

For native speakers of English only!

MORE INFO

eNetCollect – WG1

Ranking English adverbial MWEs

Ranking English adverbial MWEs

Best-worst scaling of English adve…

MORE INFO

A plan for PyBossa

Part 5: Hands-on experiment – Uploading the tasks

- Time to upload the tasks!
  - Click on **Tasks**.
  - (Enter the username and password:
    - cjvt
    - cr0wd*z1cenje)

Info	>
Results	>
Tasks	>
Statistics	>
Settings	>
Audit Logs	>
Webhooks	>
Blog	>
Contribute	>

A plan for PyBossa
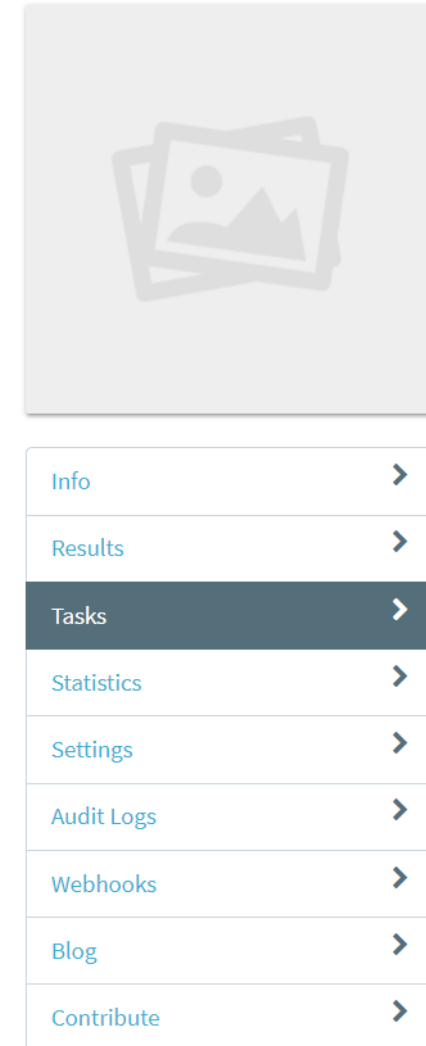
Part 5: Hands-on experiment – Uploading the tasks

- Time to upload the tasks!
    - Click on **Import** in the **Import Tasks** window.

- Click **Import Data** in the **Google Drive Spreadsheet** window.

A plan for PyBossa
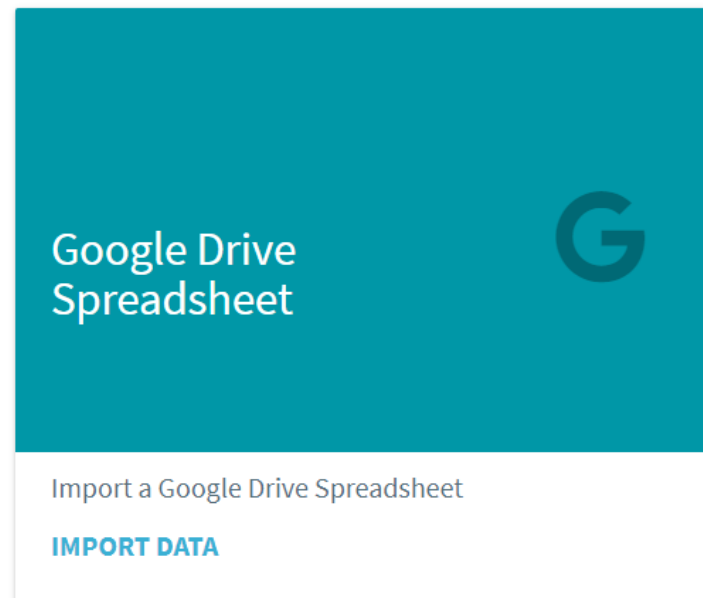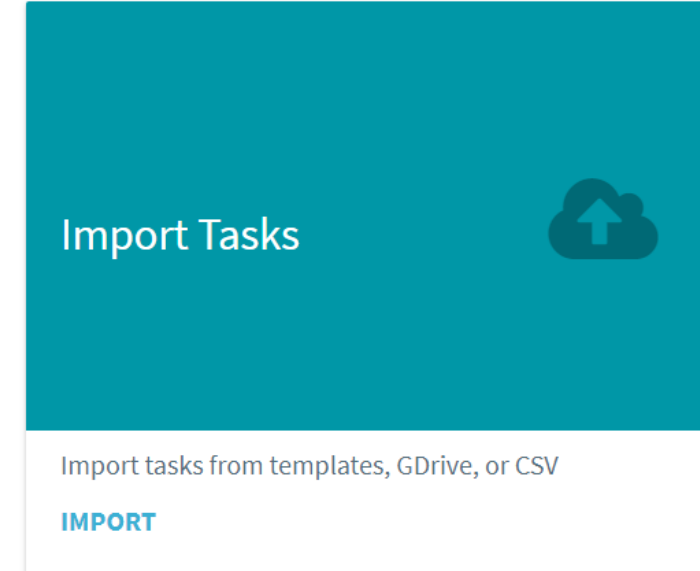
Part 5: Hands-on experiment – Uploading the tasks

- Time to upload the tasks!
  - Copy the URL to the Google Sheet with the tasks into the **URL input field**
  - URL: https://docs.google.com/spreadsheets/d/1GLohlSZ20SIWcuow3FdiVw4OGAHVYxQPqH10HpEhphk/edit#gid=0

  - Click **Import**.

Import tasks

From a Google Docs Spreadsheet

Please provide a URL to a Google Drive Spreadsheet with data for all.
For more information, please look at the documentation.

URL

The URL of the Google Docs Spreadsheet

Import

A plan for PyBossa

Part 5: Hands-on experiment – Visualizing the tasks

- Visualize the tasks.
  - Again, click on **Tasks** in the menu.
  - Click on **Editor** in the **Task Presenter** window.
  - Click **Go** in the **Basic** window.

Task Presenter

Edit the task presenter

EDITOR

Info   >

Results   >

Tasks   >

Statistics   >

Settings   >

Audit Logs   >

Webhooks   >

Blog   >

Contribute   >

Basic

The most basic template

GO

A plan for PyBossa
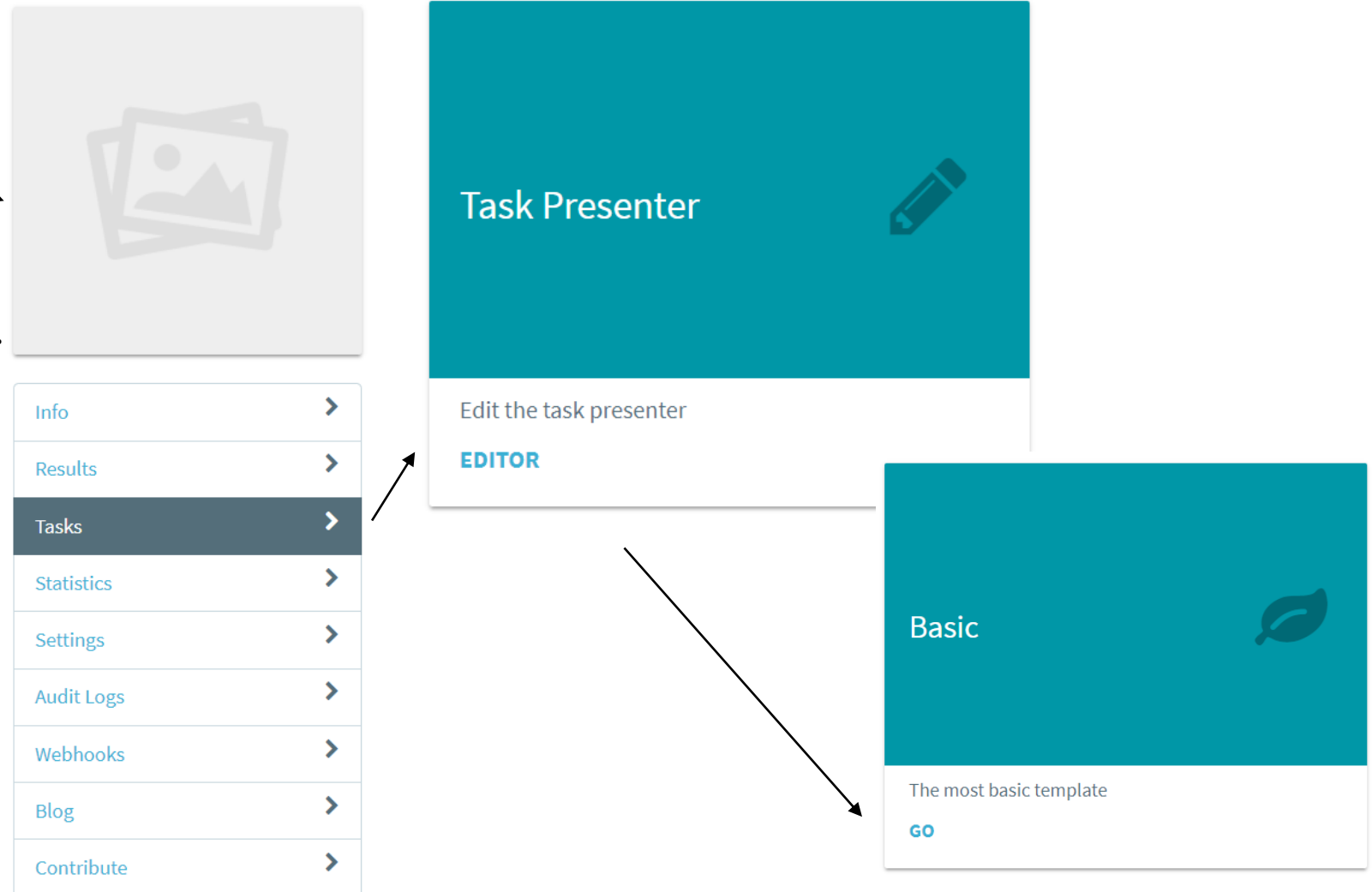
Part 5: Hands-on experiment – Visualizing the tasks

- Visualize the tasks.
  - Delete **all the code** in the window.
  - Copy the code you find at the following link into the window: https://docs.google.com/docume nt/d/1FkhqkX7gui5Dr2ojypkVeP8 H56v3Jj9EcoNQ8fXj80U/edit

  - In the code, the string 'PROJECT_SHORT_NAME' shows up two times towards the bottom – substitute it with your **project short name (and don't forget the quotes, e.g. 'some_collocations')**!

  - Click **Update**.

```
1  <div class="row">
2      <div class="col-md-12">
3          <h1>Write here your HTML Task Presenter</h1>
4      </div>
5  </div>
6  <script type="text/javascript">
7  (function() {
8      // Your JavaScript code
9      pybossa.taskLoaded(function(task, deferred){
10         // When the task is loaded, do....
11     });
12
13     pybossa.presentTask(function(task, deferred){
14         // Present the current task to the user
15         // Load the task data into the HTML DOM
16     });
17
18     pybossa.run('another_test');
19 })();
20 </script>
```
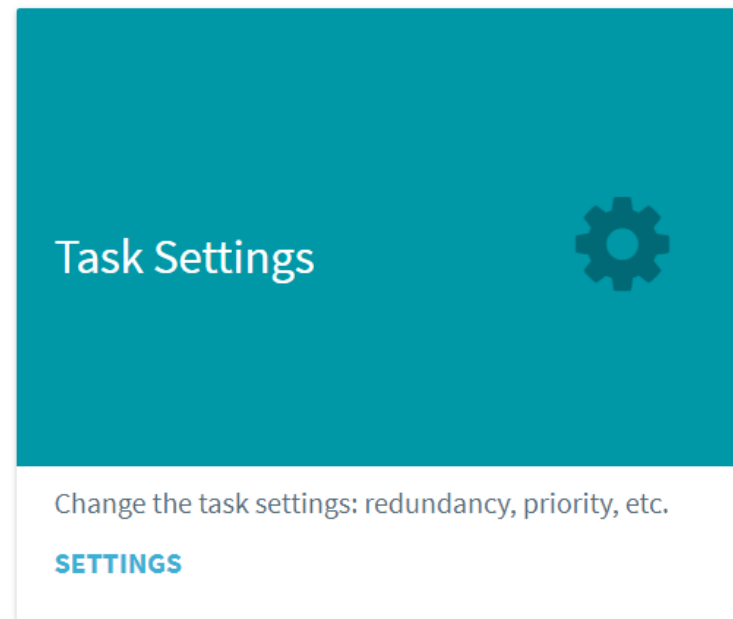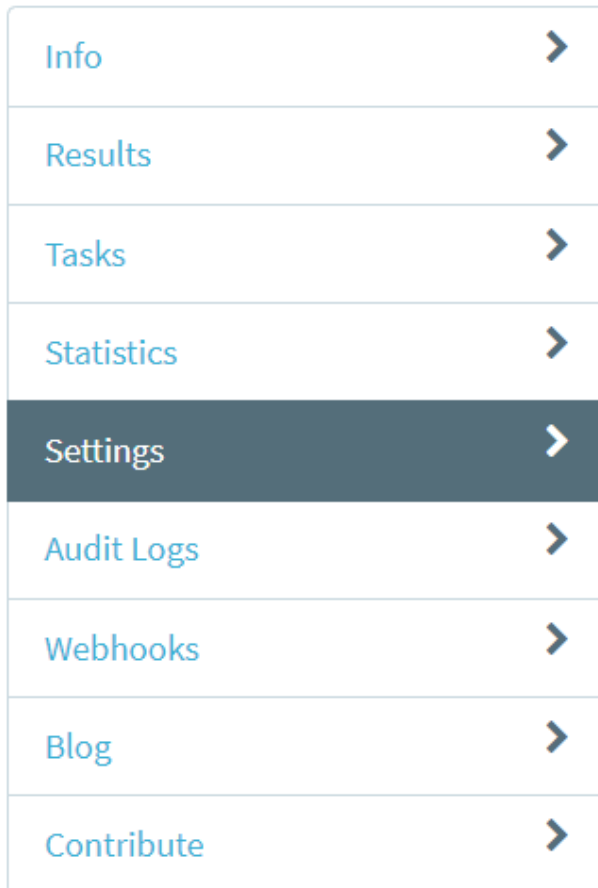
Update   Preview the task presenter

```
98         $("#finish").fadeIn(500);
99         }
100    });
101
102    function loadUserProgress() {
103        pybossa.userProgress('another_test').done(function(data){
104            var pct = Math.round((data.done*100)/data.total);
105            $("#progress").css("width", pct.toString() +"%");
106            $("#progress").attr("title", pct.toString() + "% completed!");
107            $("#progress").tooltip({'placement': 'left'});
108            $("#total").text(data.total);
109            $("#done").text(data.done);
110        });
111    }
112
113    pybossa.run('another_test');
114
115 </script>
```

Update   Preview the task presenter

A plan for PyBossa

Part 5: Hands-on experiment – Configure settings

- Set the number of responses per task.
  - Click **Settings**, then **Settings** (Task Settings), then **Redundancy** (Task Redundancy).
  - Enter the number and click **Set**.

Info	>
Results	>
Tasks	>
Statistics	>
Settings	>
Audit Logs	>
Webhooks	>
Blog	>
Contribute	>

Task Settings ⚙

Change the task settings: redundancy, priority, etc.

SETTINGS

Tasks Redundancy ↻

Change the redundancy level for the tasks

REDUNDANCY

Part 5: Hands-on experiment – Testing the project

- **Test** the project (top right corner).

- The project is still not public. You can test it and check if everything works fine before you publish it.

- Solve the tasks (there are only 10).

Part 5: Hands-on experiment – Exporting the responses

- **Export** the responses.

- Click Tasks, then Export (Export Tasks), then Export (Task Runs). Open the .CSV file in Excel.

- Not very user-friendly and very tricky to analyze (technical help).

Info	>
Results	>
Tasks	>
Statistics	>
Settings	>
Audit Logs	>
Webhooks	>
Blog	>
Contribute	>

Export Tasks

Export tasks to JSON, CSV or a CKAN server

EXPORT

Task Runs

Export task runs in CSV format

EXPORT

cjvt

A plan for PyBossa
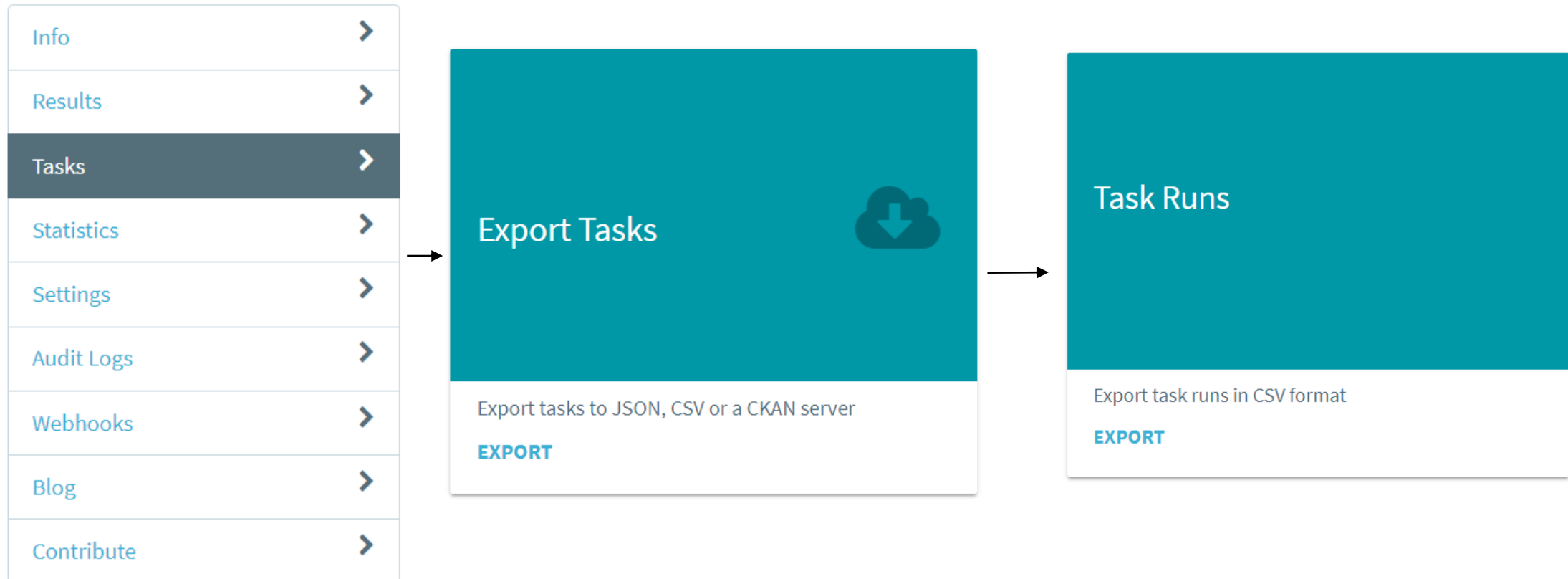
Last part: Summarizing the workflow

Data

Data in correct format (.CSV, .JSON)

PyBossa installation

Programmer to convert the data

Create a project

Upload data for tasks

Add the task presenter (i.e. the visualization instructions)

Configure settings (e.g. number of responses per task)

Test the project

Programmer to write the visualization instructions (JavaScript, HTML)

Programmer to analyze the results

Publish the project

Results

30

Thank you.

Jaka Čibej
jaka.cibej@cjvt.si

**Center za
jezikovne vire
in tehnologije**

Večna pot 113   www.cjvt.si
1000 Ljubljana   00386 14798299
Slovenija   info@cjvt.si