

Running Online experiments with Pavlovia.org

In this session, we will explore how we run an experiment online using Pavlovia.org. This will include launching studies created in PsychoPy, and making surveys and questionnaires directly in Pavlovia!

 by Becca Hirst



Session content

- Introducing Pavlovia.org
- Daisy chaining with other platforms
- Making surveys with pavlovia
- Advanced online methods (multisession testing, counterbalancing etc.)
- Debugging tips for online experiments.
- Using git based features for version control and collaboration

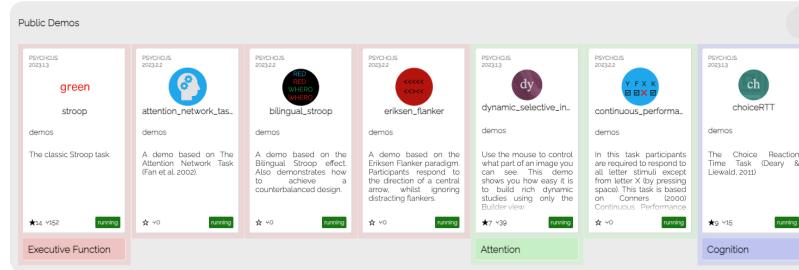
What is Pavlovia.org?

- A **secure** server for running experiments and storing data.
- A git based **version control** system.
- A huge **open access library of experiments** (that you can add to!)
- A place for creating and running Surveys (using **Pavlovia Surveys**)



Getting started with pavlovia.org

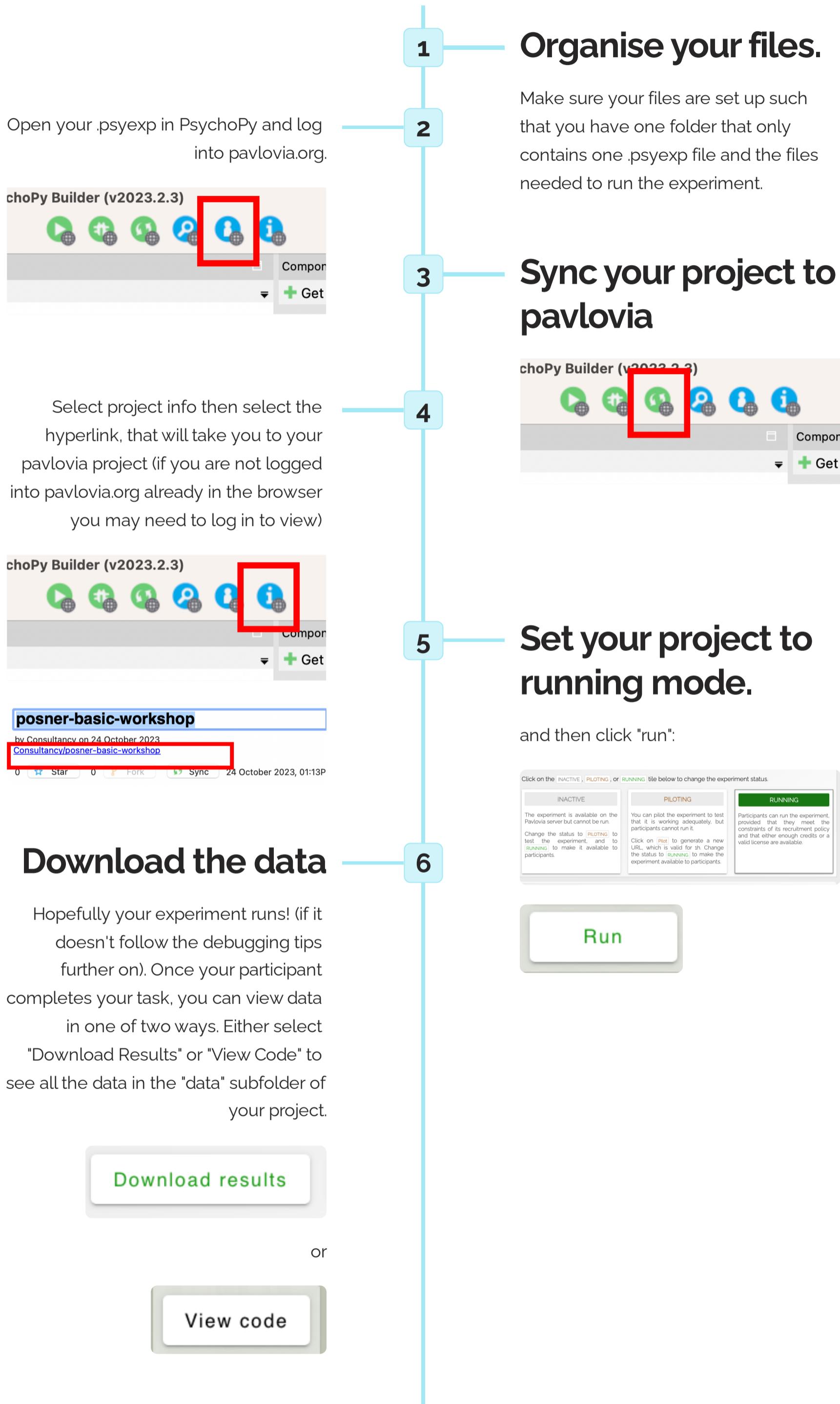
1. Make an account (use your institutional email).
2. **Explore** openly available experiments that you can use to get started.
3. **Dashboard:** This is your space, where your private experiments and surveys can be found.



i Anything in the "Public Demos" section is created and maintained by the Open Science Tools team.

Launching an experiment on Pavlovia

When you make an experiment in PsychoPy Builder, your experiment can compile to python (for running offline) or JavaScript (which allows it to run in a browser). Pavlovia.org acts as the way of **hosting** the JavaScript file online (it handles creating a URL for you to share with your participant, saving the data etc.).



- ⓘ "Database" mode will store all participants to one file, you will now be able to view this data in the repository via the "View code" option. "csv" mode will store one csv and log file per participant (much like how data are stored when you run an experiment offline in PsychoPy).



Daisy chaining with other platforms

Sometimes you might want to connect your experiment with other platforms. For example, you might use a third party service to help with recruitment (such as [prolific](#) or [SONA](#)). In these cases you want your experiment to go something like the chain below, you want information like "participant ID" as well as other info to be passed forward through the chain.



How does it work?

Information is passed to a website using something known as a *query string*. This is essentially any information that follows a "?" in the URL in your browser. For example, your task could have the URL:

run.pavlovia.org/your_user_name/your_experiment_name/

This would likely start up your task with a dialogue window where you can type participant and session. You could however provide the value for "participant" already within the URL using a query string, like this:

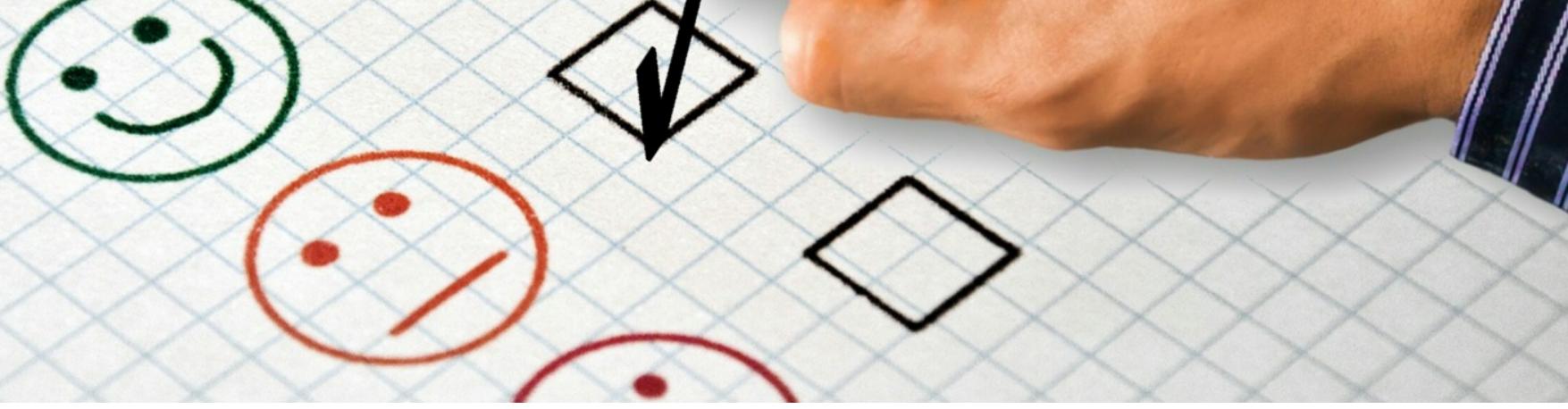
run.pavlovia.org/your_user_name/your_experiment_name/?participant=123&session=1

Here the last part of the URL is a query string, and you will notice that this autocompletes the fields "participant" and "session" in the startup dialogue of your experiment. This is what happens when you daisy chain with other platforms, platform 1 will send information via query string to platform 2, then platform 2 sends it to platform 3 and so on...

How do I set up a daisy chain?

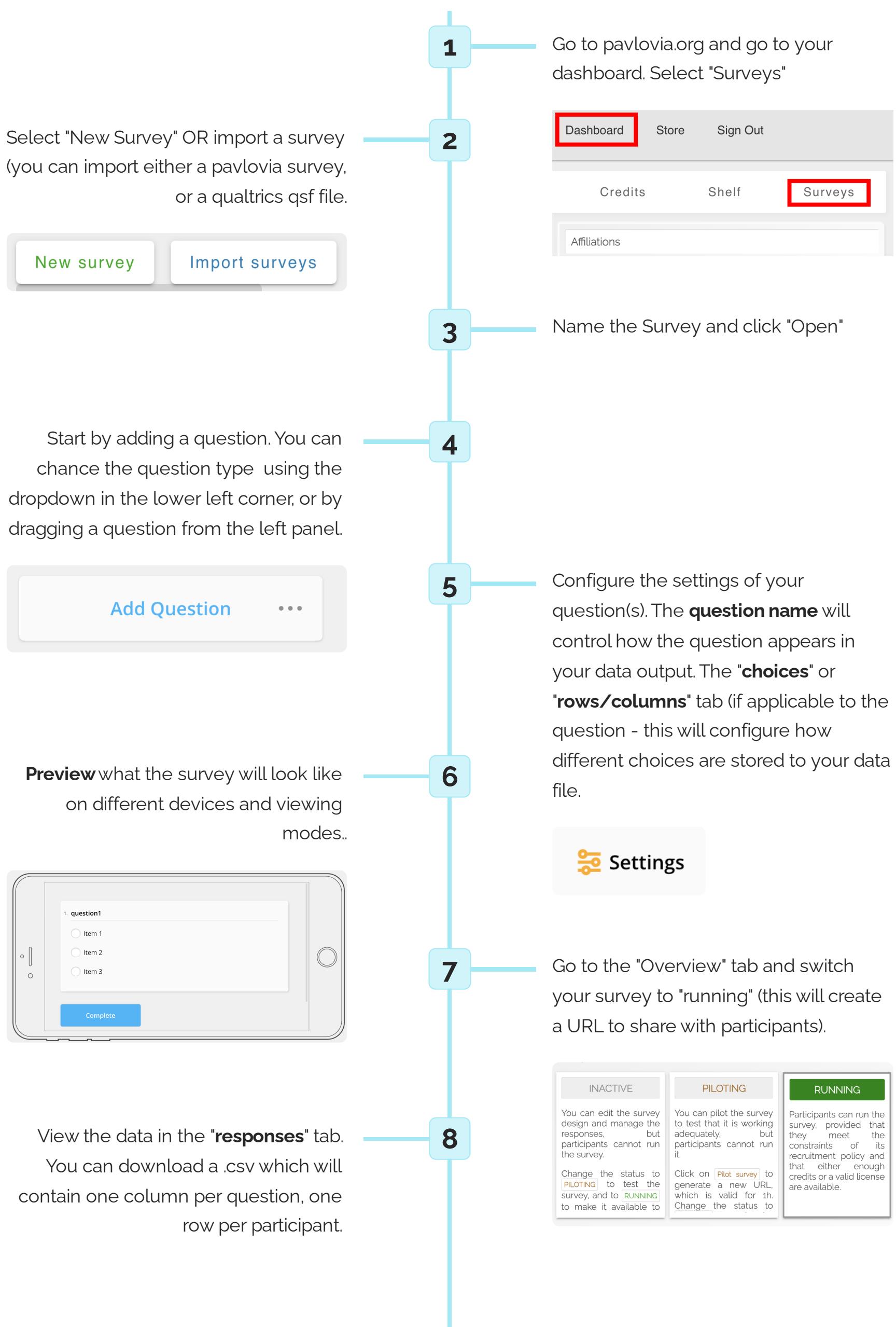
How you do this will vary depending on platform, but you can find individual guides below:

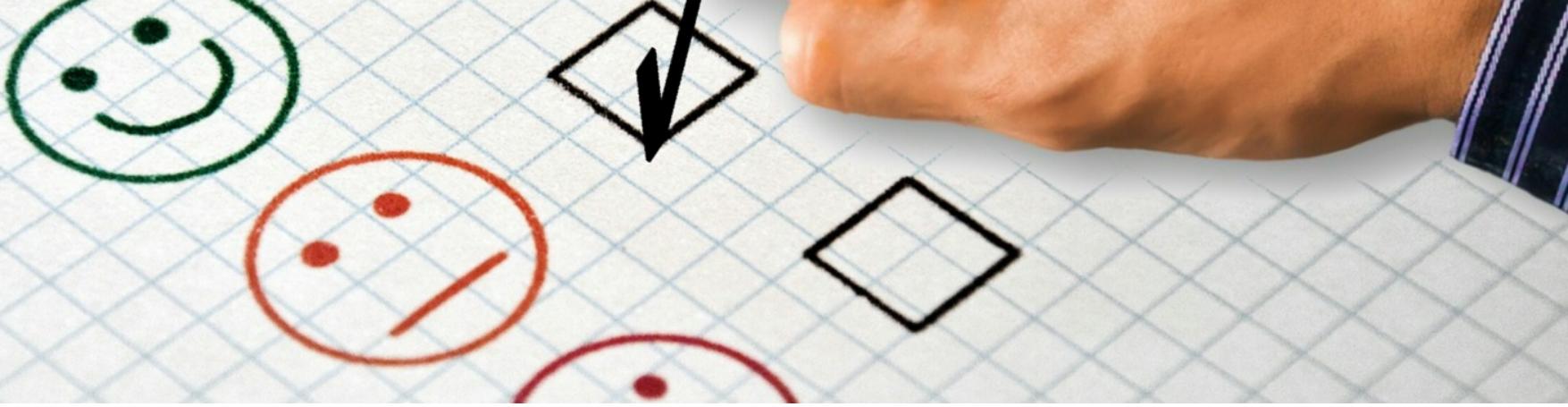
- [Prolific + Pavlovia](#)
- [SONA + pavlovia](#)
- [Qualtrics + pavlovia](#) (Note: you can now use pavlovia surveys in place of many qualtrics surveys, which can be added directly to your experiment).



Making Surveys on Pavlovia.org

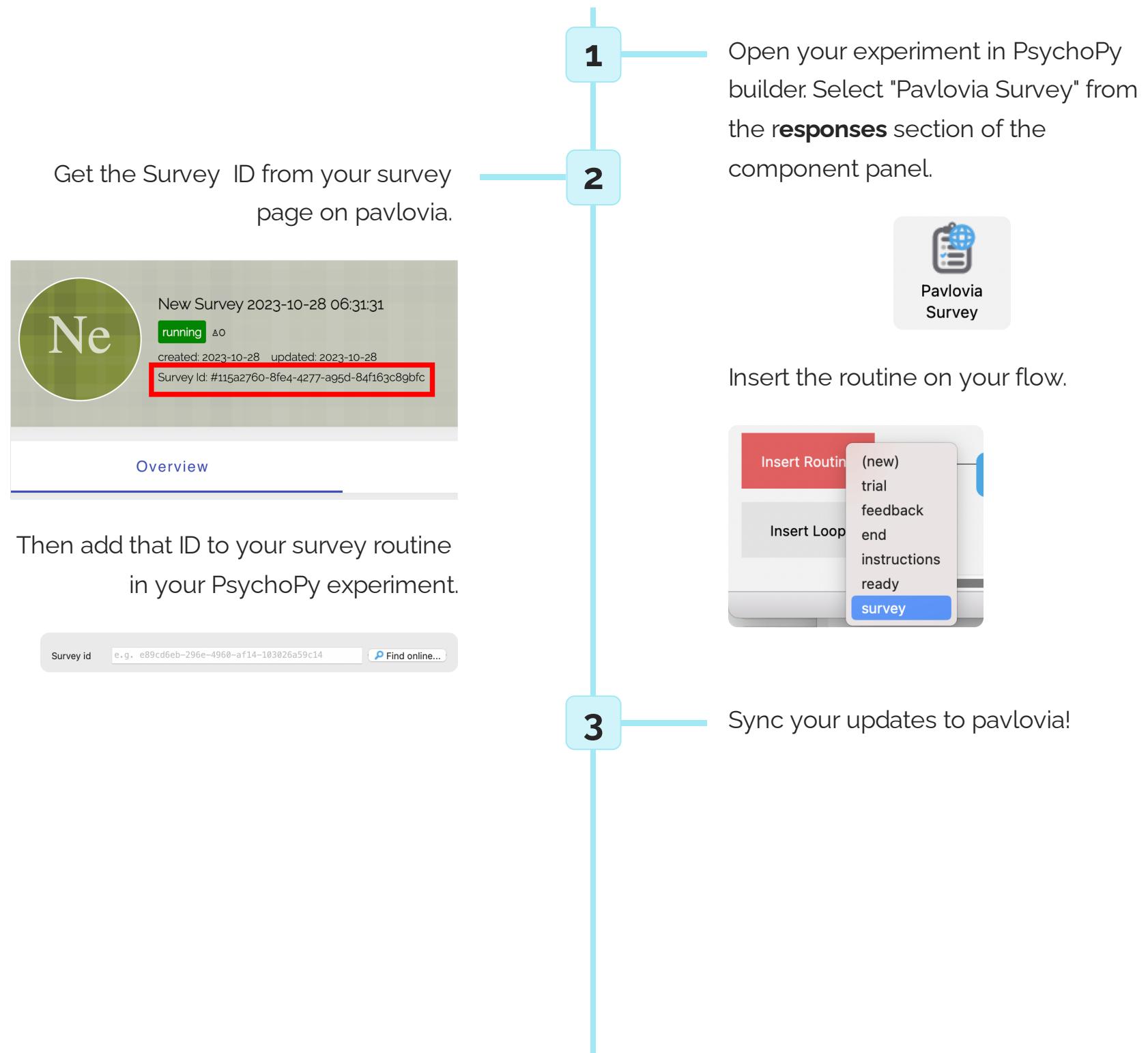
Pavlovia now has the capacity to make surveys, if you only want to run a survey (i.e. no experiment) then you technically do not need the PsychoPy app at all in order to build a survey!



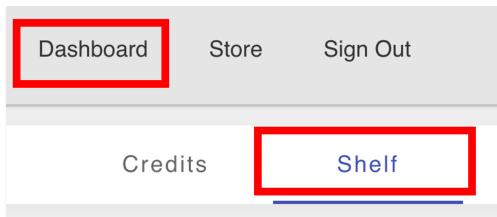


Integrating a survey with your experiment

If you are running your experiment online, you can embed your survey directly in your experiment. This means no daisy chaining is needed, everything will be presented as part of the same session.



Advanced online methods - the "Shelf"



The "Shelf" on pavlovia is used to house information to be stored and passed between experimental sessions or between different experiments. Examples include:

- **Counterbalancing:** Automatic counterbalancing online requires keeping track of how many slots are available per group, and updating available slots as the experiment progresses.
- **Multisession testing:** This requires tracking how many times each participant completes an experiment so information needs to be stored between each experiment load.

You can find a more complete guide on using Shelf [here](#).

⚠ The Shelf is still a relatively new feature, at the moment (PsychoPy 2023.2.3) you can only interact with it through code. For further support please use the [forum](#). We are still optimising the Shelf for other use cases e.g. multiplayer games.

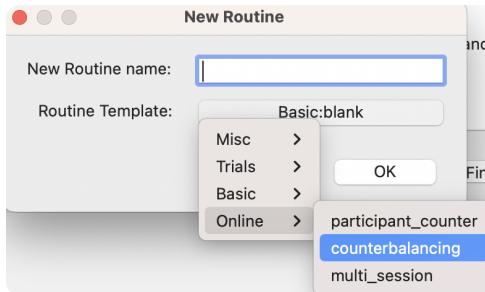
Counterbalancing online using the Shelf

Set up your Shelf record

Add a Shelf record:

- **Scope:** Experiment
 > Your experiment.
- **Type:** Dict
- **Value:** A dictionary listing group names and available slots.

```
{"groups":["A",  
"B"],  
"groupSizes":[10,  
10]}
```



Add a routine to your experiment

There is a template routine to get you started using the shelf for counterbalancing. From templates select Online > Counterbalancing.

Update the code so that it interacts with your shelf record.

Replace `my_groups` with the name of your shelf record.

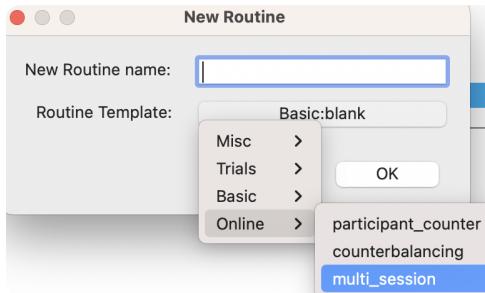
⚠ Remember to handle what happens if counterbalancing is complete. You can use `counterbal.finished` to skip the experiment if counterbalancing is complete (see branching in session 2 slides).

Multisession testing using the Shelf

Set up your Shelf record

Add a Shelf record:

- **Scope:** Experiment
 - > Your experiment.
- **Type:** List
- **Value:** An empty list []



Add a routine to your experiment

There is a template routine to get you started using the shelf for multisession testing. From templates select Online > multi_session.

Update the code so that it interacts with your shelf record.

Replace all instances of `session_tracker` with the name of your shelf record.

⚠️ It can take a moment for your experiment to fetch values from the Shelf, so you might want to make the text in "multi_session" last longer (or infinite with a response to end the routine)

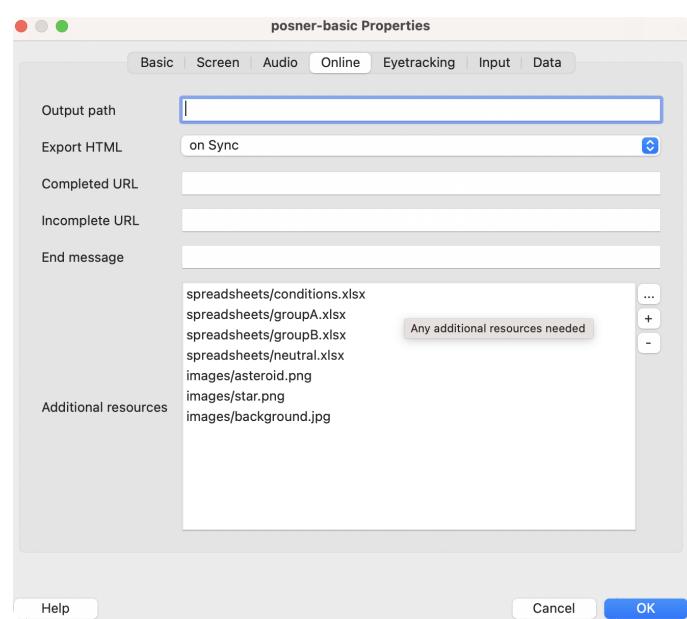
Debugging online: Common errors

When you run your experiment offline in the PsychoPy app, if your experiment doesn't work error messages will appear in the StdOut section of the "Runner" window. Online this can be trickier, but here are some tips to help, remember to use the [forum](#) if you are not sure!

Unknown Resource"

Explanation: This means your experiment cannot find a resource it needs in your experiment (e.g. a picture or spreadsheet). When you run an experiment online, all resources required first need to be loaded to the browser window.

Solution: Add resources to the "Online" tab of your experiment settings.



"Initializing experiment"

Explanation: There is a syntax error in the compiled js file of your experiment, meaning it cannot initialise properly.

Solution: Right click anywhere on the "initializing" window and select "inspect". On the developer window that opens select "console". Refresh your window. The console will provide more information about where the error is arising from. Common reasons for this error include:

- A component with invalid syntax e.g. the position argument of a text component missing a closing bracket (0, 0.5 rather than (0, 0.5)).
- A JS code component with a syntax error.
- An Auto > JS component where you use python that doesn't translate well to JS e.g. `import random`

ⓘ For a complete list of tips to help with potential issues in code component translations take a look at Wakefields [PsychoPy to Javascript crib sheet](#).

Using Git based features

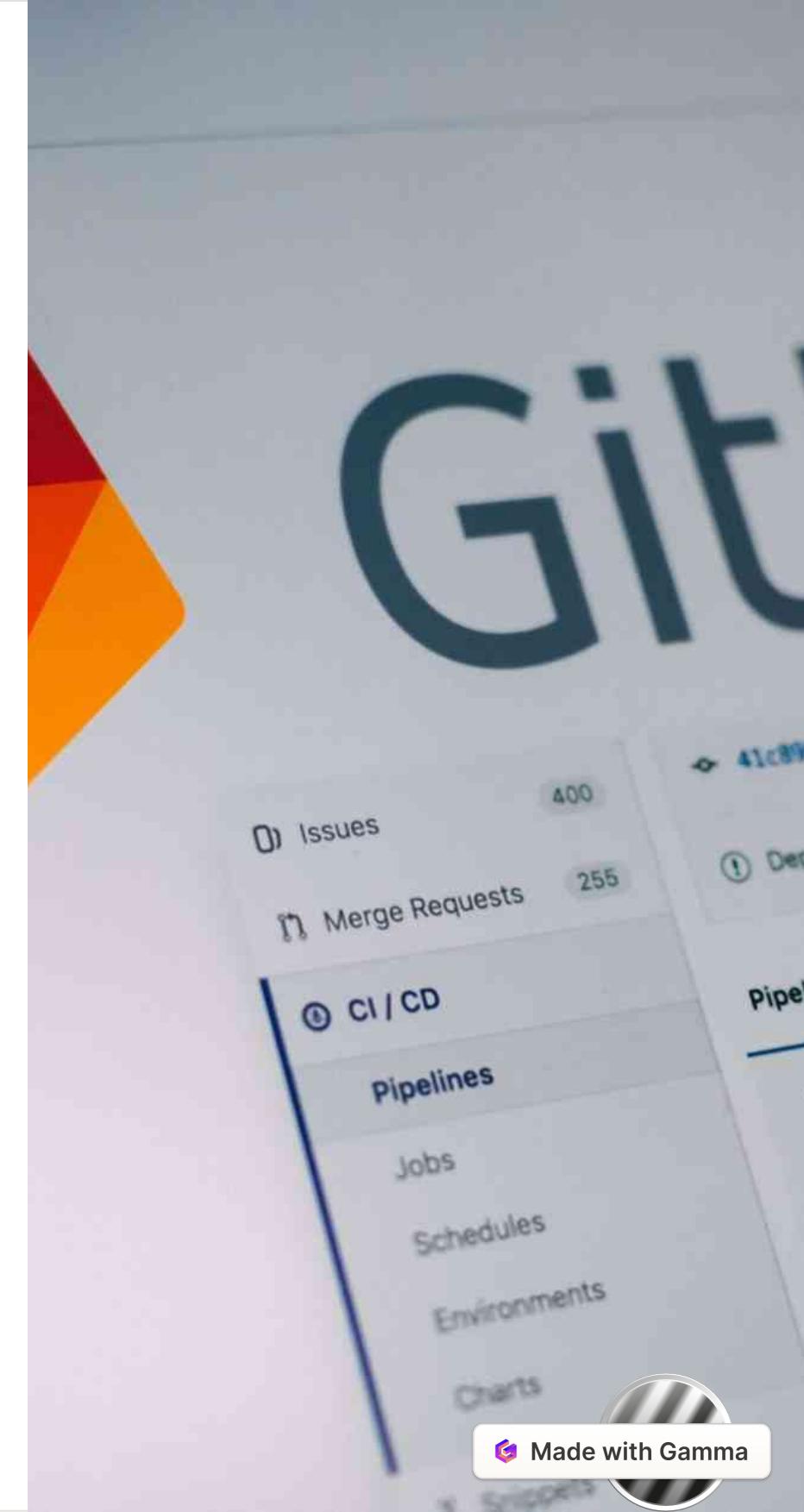
Pavlovia exists on a system known as [gitlab](#). Gitlab is pretty feature rich, but it allows many neat features and control over your projects:

- **Version control:** view the version history of your experiment, and restore old versions.
- **Collaboration:** Add team members or fork projects to groups.
- **Privacy settings:** Make your project public or private.

You can access the gitlab repository of your project by selecting "View code" on your project dashboard.

View code

ⓘ Your pavlovia projects are private by default. They do not become public for sharing until you change this setting in gitlab.





That's all for this session!