Biosecurity Game

Administration Instructions

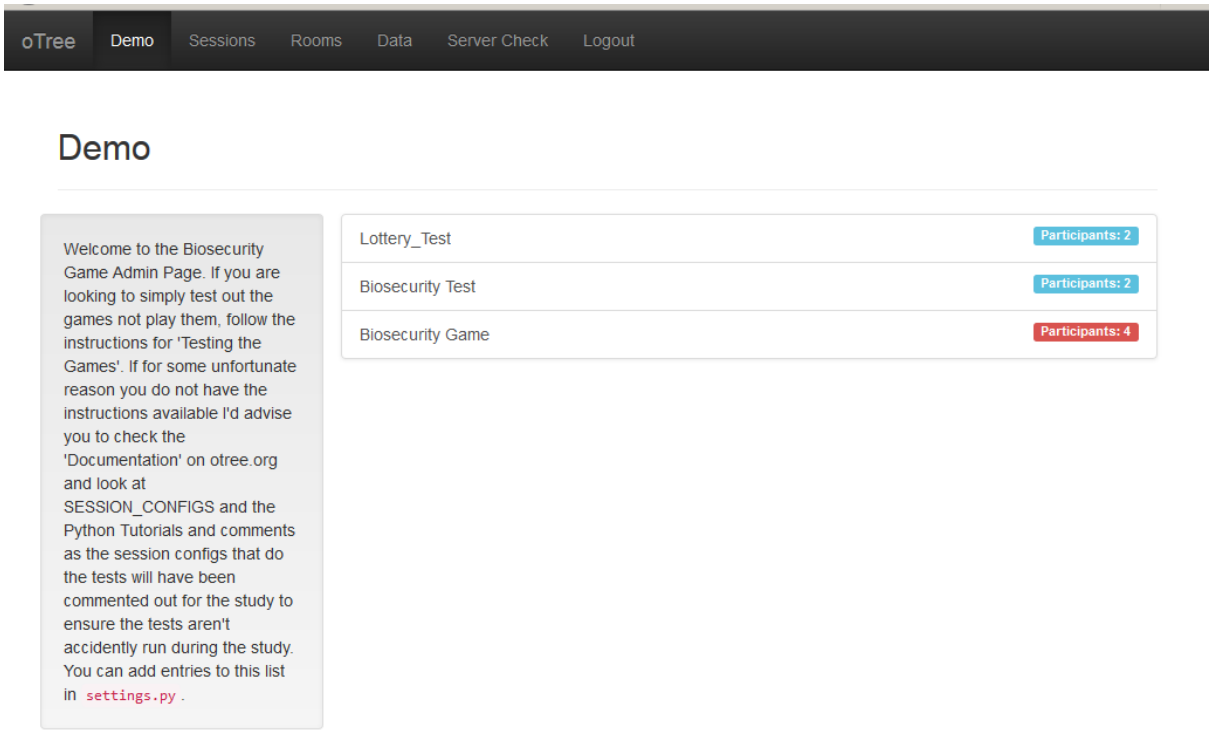
Authors: Joel Dunstan, Justin Chong

Biosecurity Game: Administration Instructions
Table of Contents

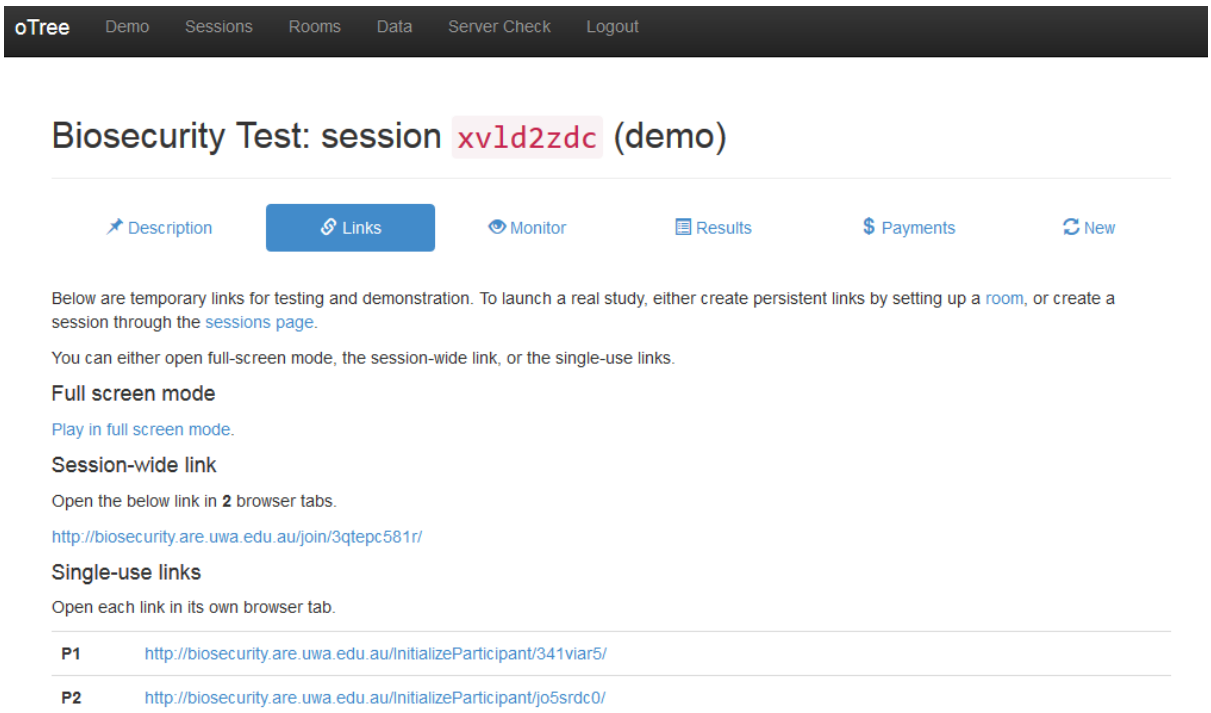
Demo

The “Demo” section of the site can be used for testing the games and for demonstration. It is not recommended for actual studies. To create a demo session, click on one of games in the list. “Lottery_Test” is the lottery game on its own, “Biosecurity Test” is the biosecurity game on its own. It is these Demos that can be primarily used for testing out the games without saving any results. The Biosecurity Test does not do the group logic as the group matching functions lie within the Lottery Game.
So when you doing the Biosecurity Test, the number of player per group will default to the number of participants in the session. In this case, the Biosecurity Test will have 2 players, it will also do the demo without any features enabled as all of these have been turned off by default, such a feature is the Chat Box which allow users to chat.

The Biosecurity Game will run the Lottery Game, the Biosecurity Game and the end Results page. While it will get to the end and show the results for the game, it will not save these results like the Test Sessions.

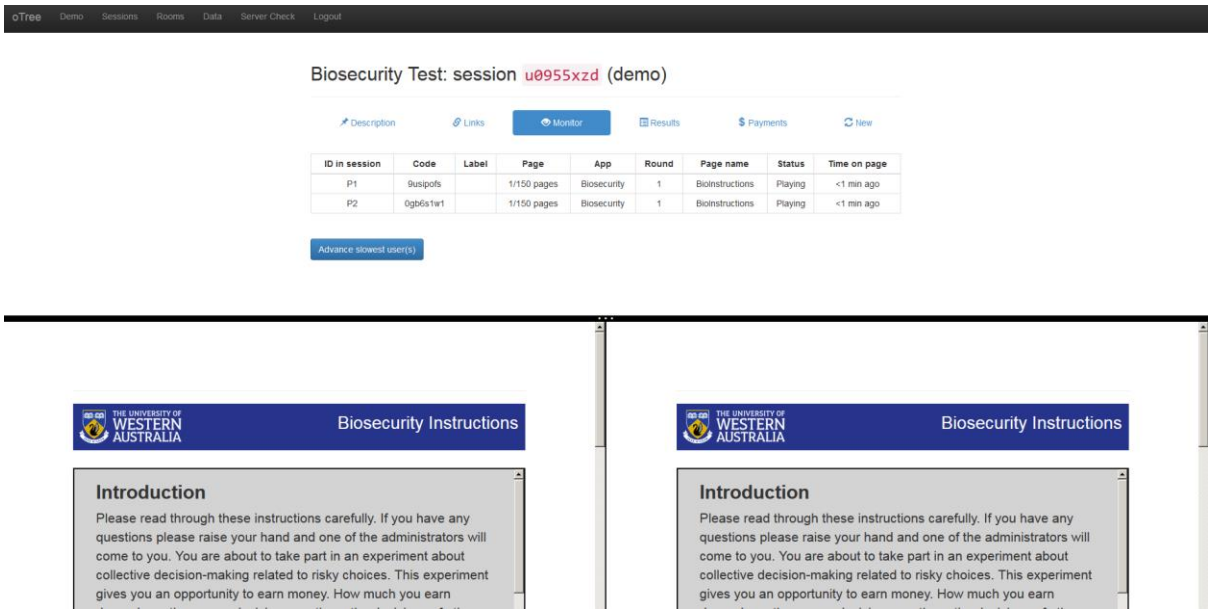


After clicking on a game link, you should see the following screen:



Full Screen Mode

Clicking “Play in fullscreen mode” allows you to view the game in fullscreen mode where you can view all windows of the game session in the same browser tab. This will only appear in the Demo page.

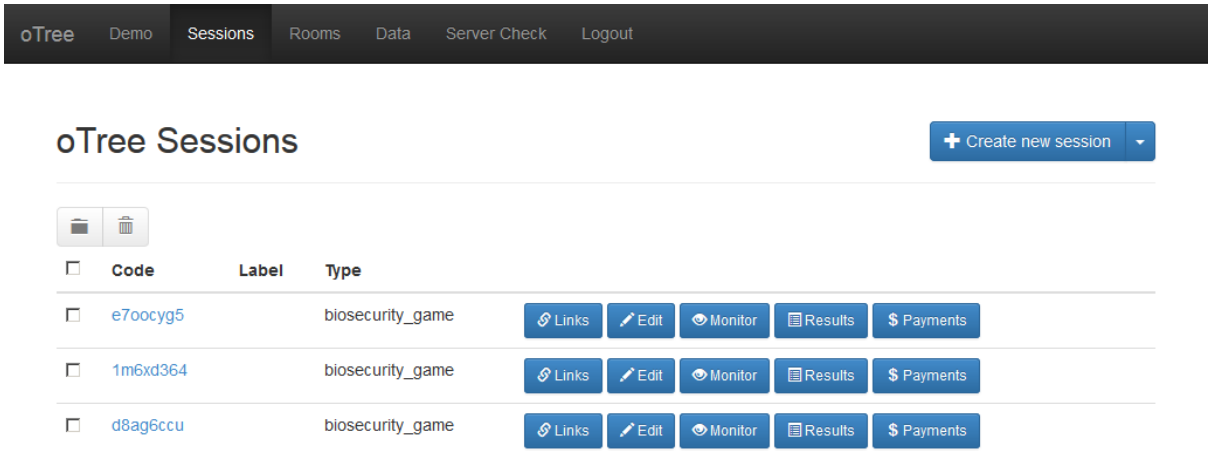


New

The New link exists only in the Demo; it will recreate the session if you want to try the demo again.

Sessions

The sessions tab displays a list of all the active sessions.



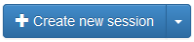
Archiving Sessions

To archive sessions, click on the checkboxes to select the desired sessions that you want to archive (a session is selected when it shows a tick inside the check box). Once you have selected the sessions you wish to archive, click the folder button next to the trash bin, it will highlight that it's the archive button. Once it's archived it will come up with a link called "Show Archived Sessions" below all the list of all active sessions, clicking on this link will show all the archived sessions, allowing you access to those sessions again.

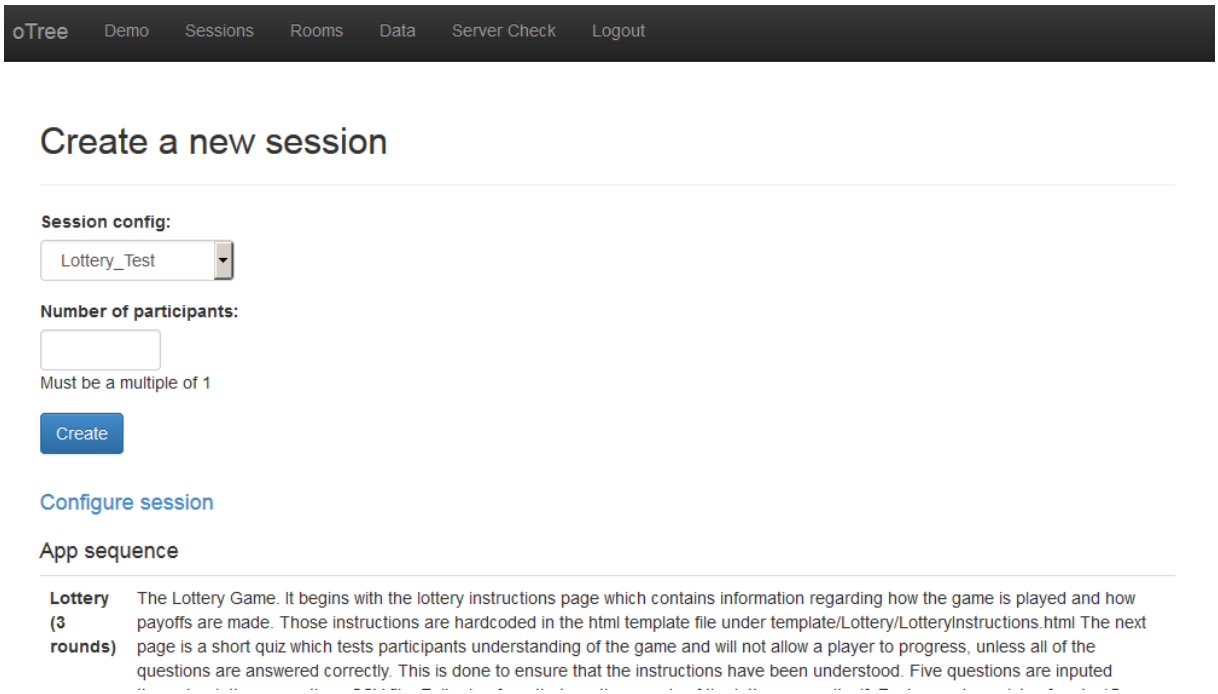
Deleting Sessions

To delete sessions, click on the checkboxes to select the desired sessions that you want to delete (a session is selected when it shows a tick inside the check box). Once you have selected the sessions click the trash bin button to delete the sessions, it will come up with a warning saying that the action is irreversible allowing you to cancel the request, if you did the delete button by accident. Click Delete to delete the selected sessions.

Creating sessions

To create a new session, click this button 

It should take you this screen



There are two forms on this page, one is the Session_config drop down menu, this allows you to choose between different session configurations, we have the Lottery Test, the Biosecurity Test and the Biosecurity Game. The Biosecurity Game will be the one you use to create new sessions for the experiments.

The Number of participants is exactly as it sounds, the number of players inside the session, please enter an integer here. When you have configured the session as per the variables below, click the “Create” button to create the session. If you aren’t concerned with configuring the session, skip the configure session, straight to Links

Configure Sessions

The link, “Configure Sessions” will bring up another form list, this will allow you to set custom settings for the session. I will explain the settings below:

dynamic_finances

The dynamic finances feature that reads from a csv file the revenue, upkeep and max protection for each round. This will allow and administrator to have custom and dynamic costs and revenue per round.

max_protection

This is the max protection for each round and will directly affect the slider on the Round pages. The max protection is one of those elements of the session that will affect the balance of the game and its difficulty, take great care with the value in here.

player_communication

This is a checkbox for the enabling of the Chat Box, when the player_communication check box is ticked it will run the Chat Box on rounds 1, 6 and 11.

players_per_group

This is the amount of players per group, the amount of groups is Number of participants / players_per_group. The group logic installed will accept of Number of participants that isn’t a multiple of players_per_group. In the case where Number of participants isn’t a multiple of players_per_group, it will pluck one player from every group starting from the first, until the last group is full or until there are no other groups to pluck from other than the last.

revenue

This is the revenue that each player gets per round in the event an incursion doesn’t occur. The revenue is one of those elements that can directly affect the game’s difficulty, choose a number wisely and with reason.

set_leader

This is the One Player Feature check box, when this is enabled, it will ensure that one player is chosen to be the leader, the leader will choose their amount of protection first before everyone else. After the leader, has chosen their amount of protection, everyone else in the round will get to see what the leader put in as their protection before choosing their own amount of protection.

starting_funds

This is the funds each player starts with, while this does directly affect the balance of the start of the game, it will not have as much affect as the revenue, upkeep or max protection.

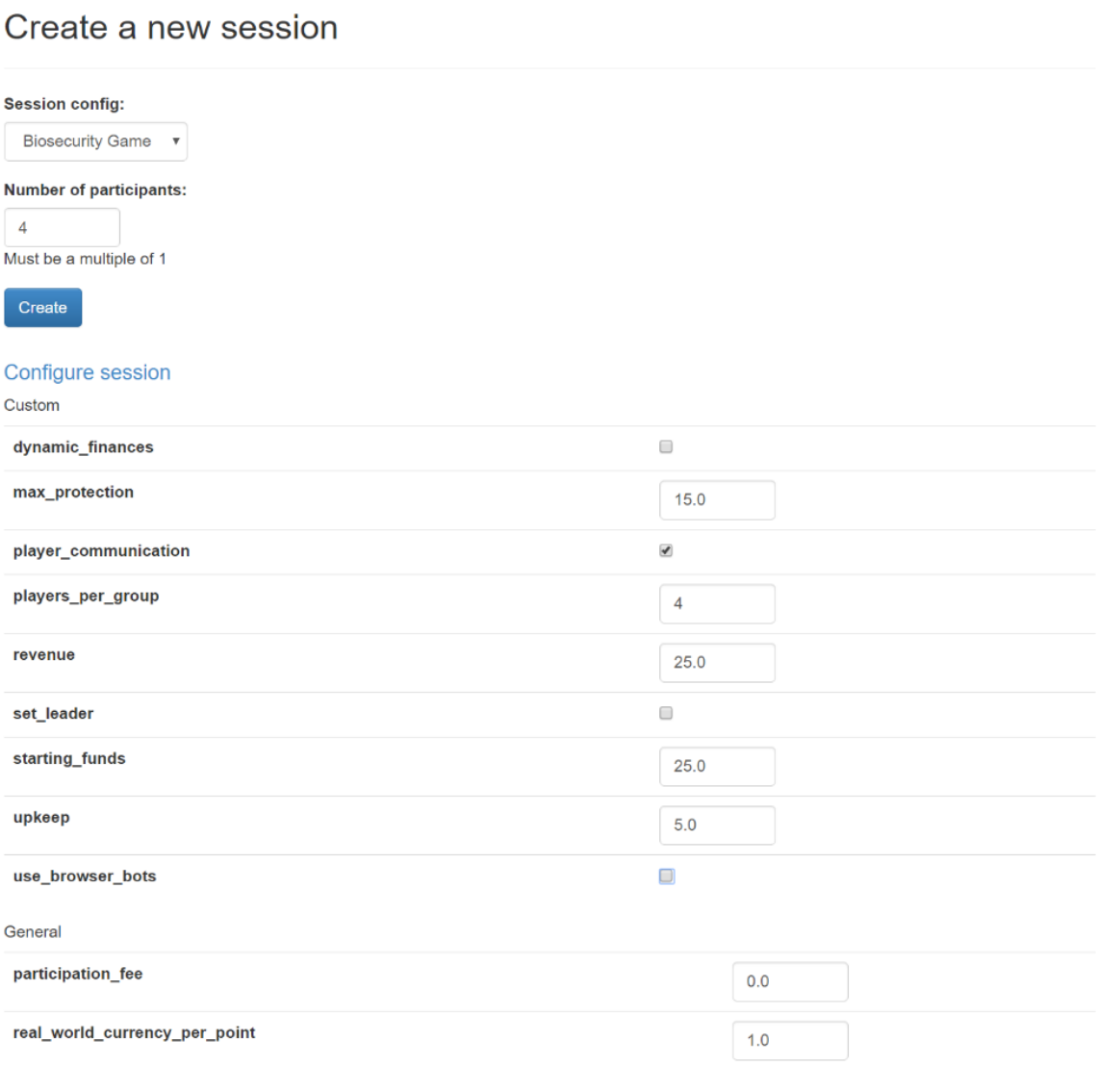
upkeep
This is the cost of produce per round, this is one of those elements that will directly affect the balance of the game, please have a reason for changing this value.

use_browser_bots
When this is enabled, it will run browser bots every time the game is opened, the bots will use the test.py in each app to run through the games. This is a way of testing the game inside the browser.

participation_fee
This is the amount you can set in real world currency ($AUD in this case) which denotes how much a user will be paid for simply playing the game.

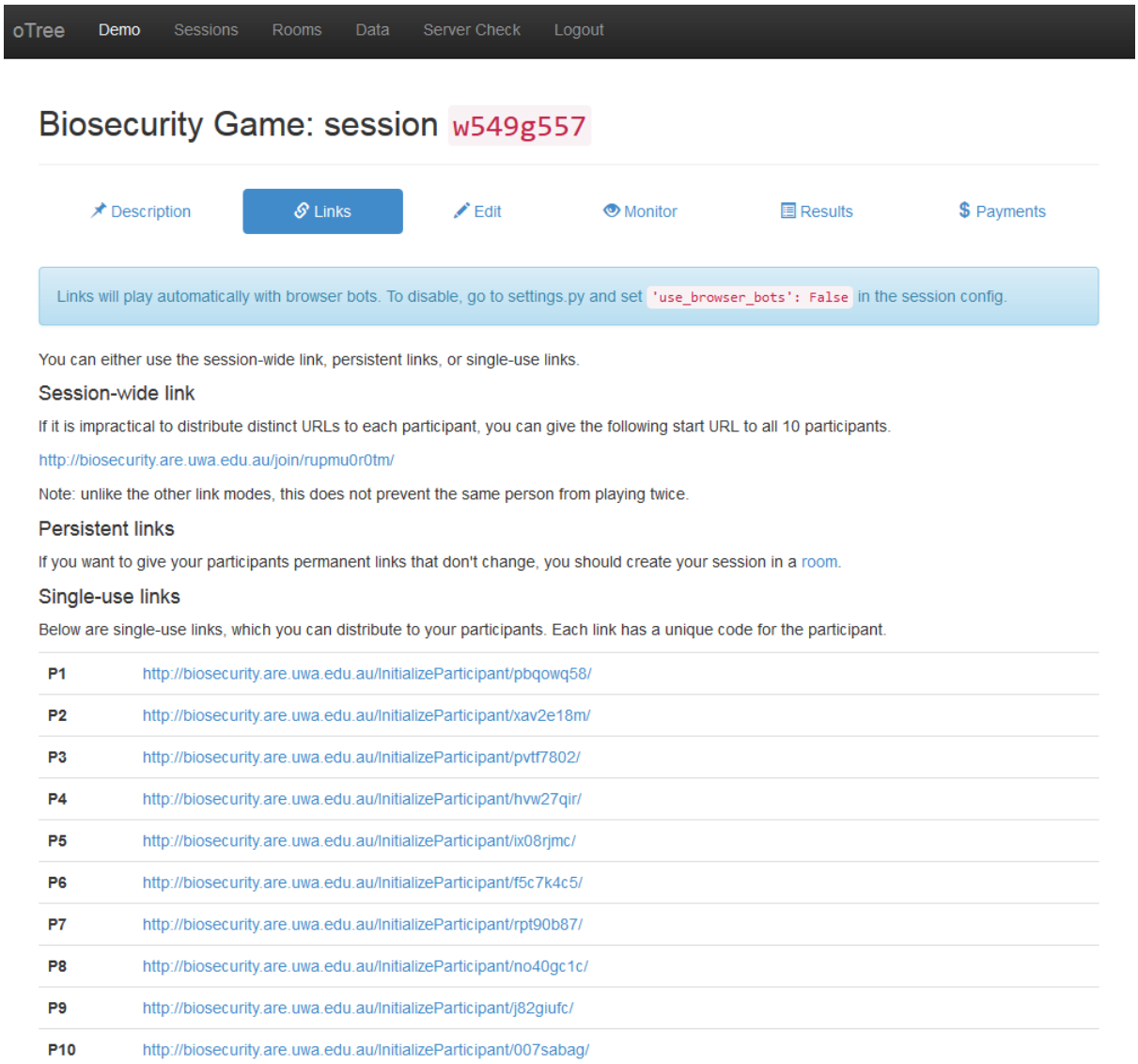
real_world_currency_per_point
This value is the exchange rate for points to real world currency.

An image of the Configure Sessions form is below



Links

After clicking create it will load up a page full of links.

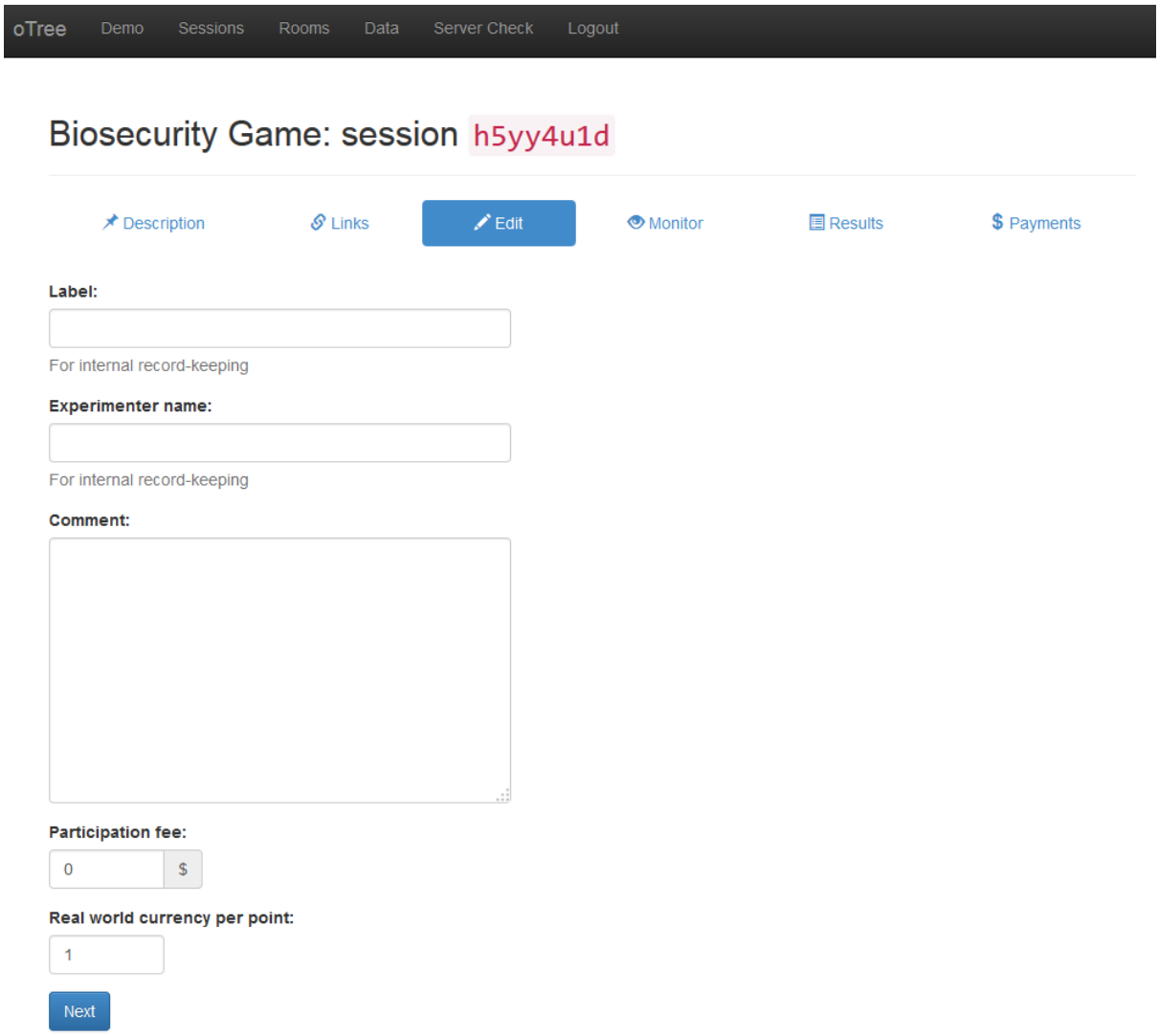


“Session-wide link” allows you to give the same link to multiple participants. This will allow users to open multiple windows and is not recommended. To prevent this you can append ?participant_label=<NAME>_<SURNAME> to the link. For example: http://biosecurity.are.uwa.edu.au/join/vdyqpgut8k/?participant_label=bill_gates. This will create a unique link that can only be used by one participant and set the name of the participant label to be the participant’s name. The participant label will also show up on the Payments page as discussed later which will make it easier to know how much to pay everybody. **This is the recommended way to distribute links to players.**

Persistent links aren’t used since we are not using Rooms.

“Single-Use Links” are unique links for each individual participant, ?participant_label= can also be appended to the link to set the participant labels. This will ensure that a participant can only access the game through one link, once, all the time.

Edit

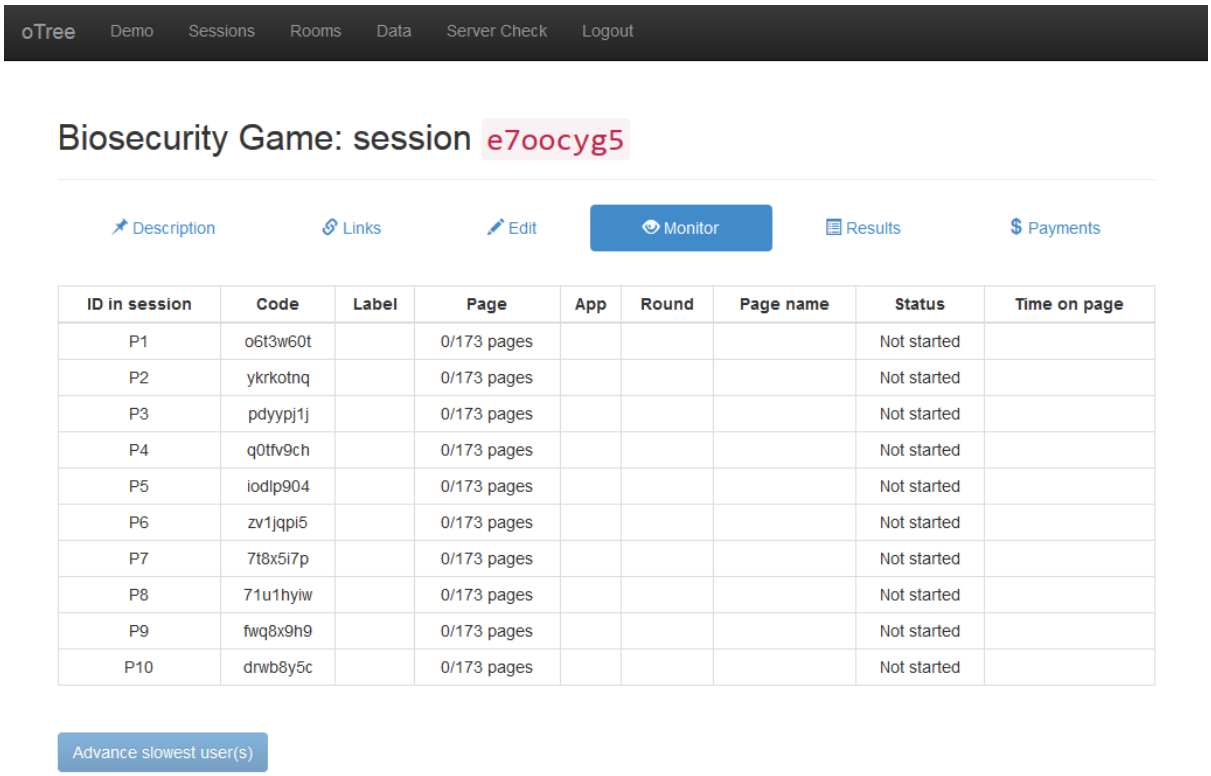


Allows you to edit the following parameters

- Label – It is recommended that you put in the date and time of the experiment, or the date and time the session was made, to make sessions easily distinguishable.
- Experimenter name – Purely optional
- Comment – Purely optional, if there’s something special to mention about the session, here is the best place to put it.
- Participation Fee – as per Configure Sessions
- Real world currency per point – as per Configure Sessions

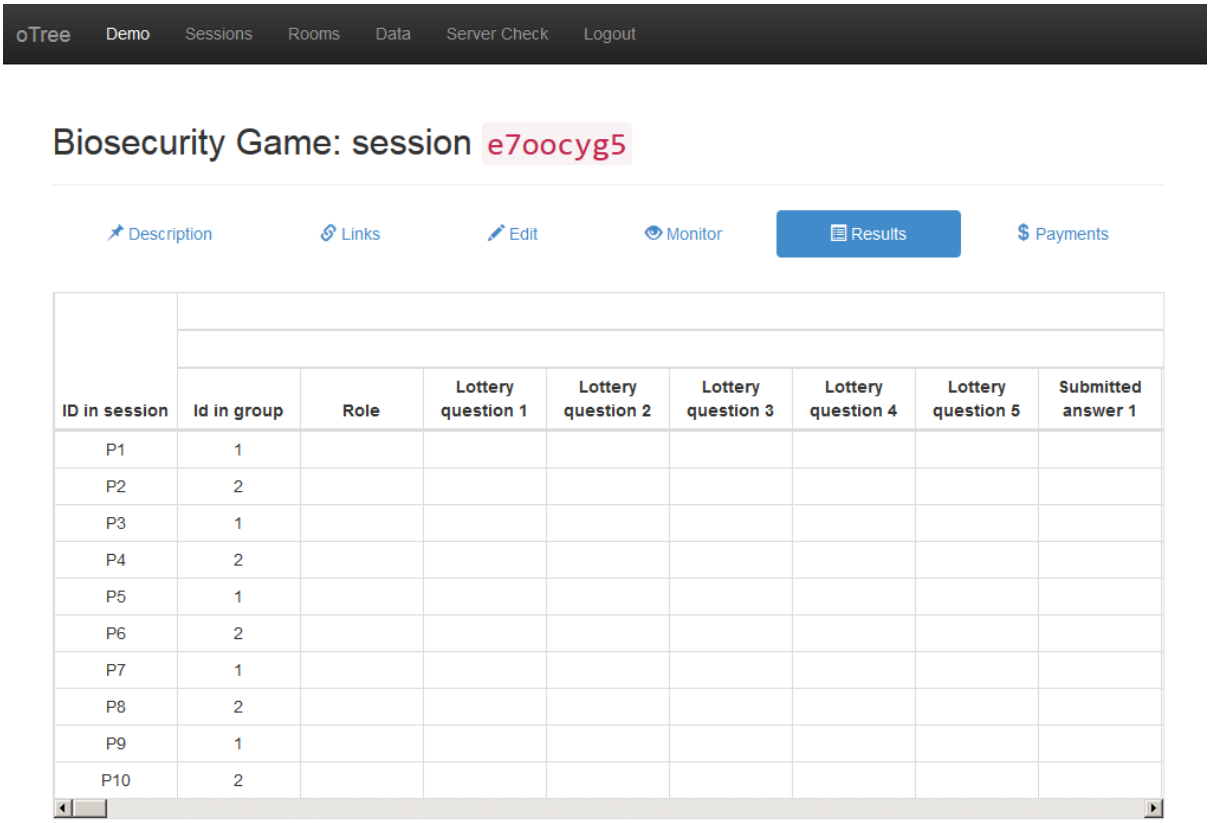
Monitor

Displays the status of all the players in the session and allows you to advance slowest users. Advance slowest users **DOES NOT** differentiate between groups; it will move everybody forward in the session if they are behind, even if they are in a different group.
It will also display the participant labels if you set them as per Links.



Results

Displays a table of results for the game that updates dynamically.
All player actions, form entries and incursion results will be displayed in this table.
Boolean values will show as 1 for true and 0 for false.

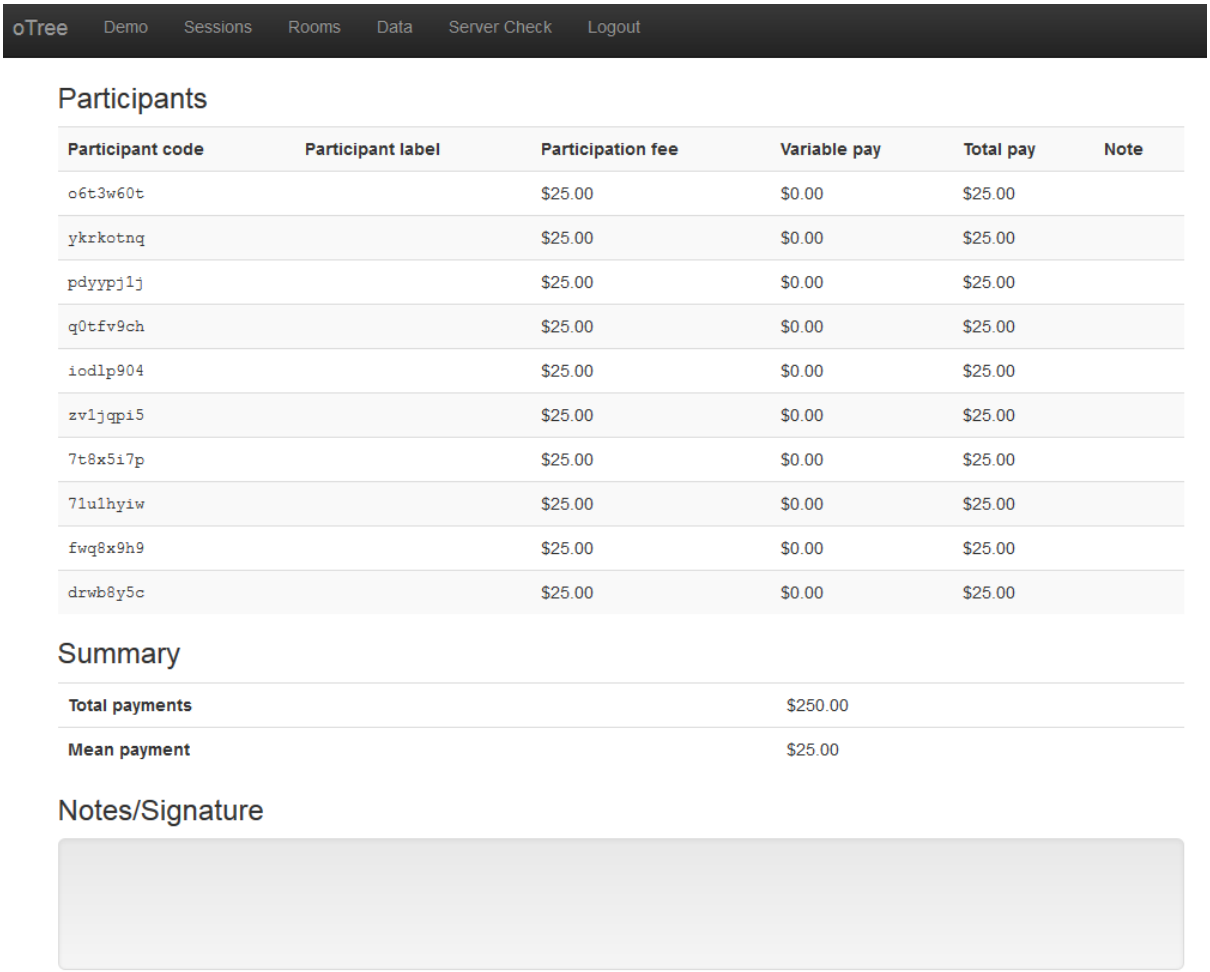


ID in session	Id in group	Role	Lottery question 1	Lottery question 2	Lottery question 3	Lottery question 4	Lottery question 5	Submitted answer 1
P1	1							
P2	2							
P3	1							
P4	2							
P5	1							
P6	2							
P7	1							
P8	2							
P9	1							
P10	2							

The results on this page can't be saved to local storage, these results are simply for viewing.
If you want to save the data to local storage, check Data.

Payments

The payments page will show useful information on the amount of money that is owed to each player. If you set the participant labels as we recommended you should be seeing the participants full name in the participant label, making it extremely easy to work out who should be paid what. It'll also show the Total payments and the average payment per user.

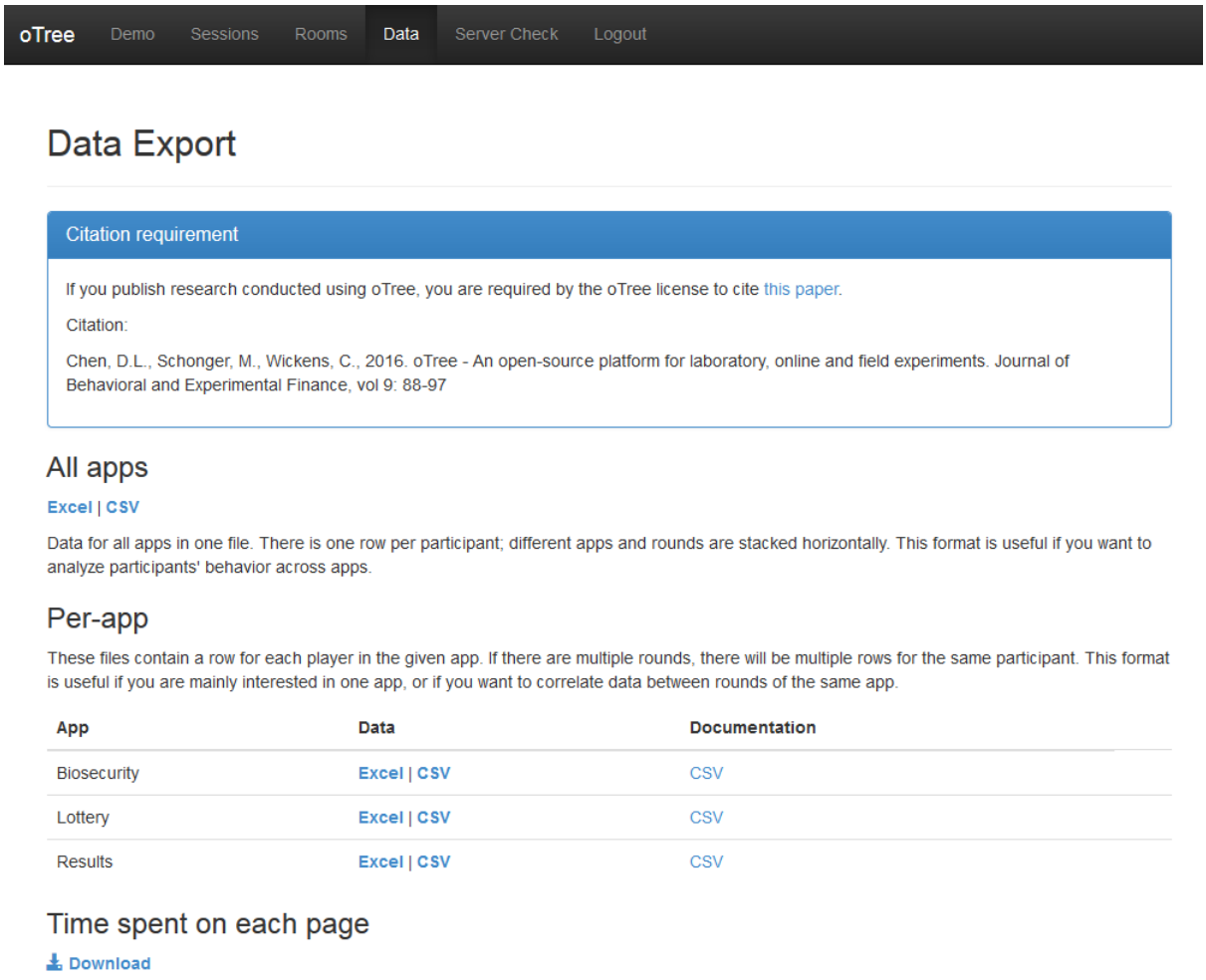


Rooms

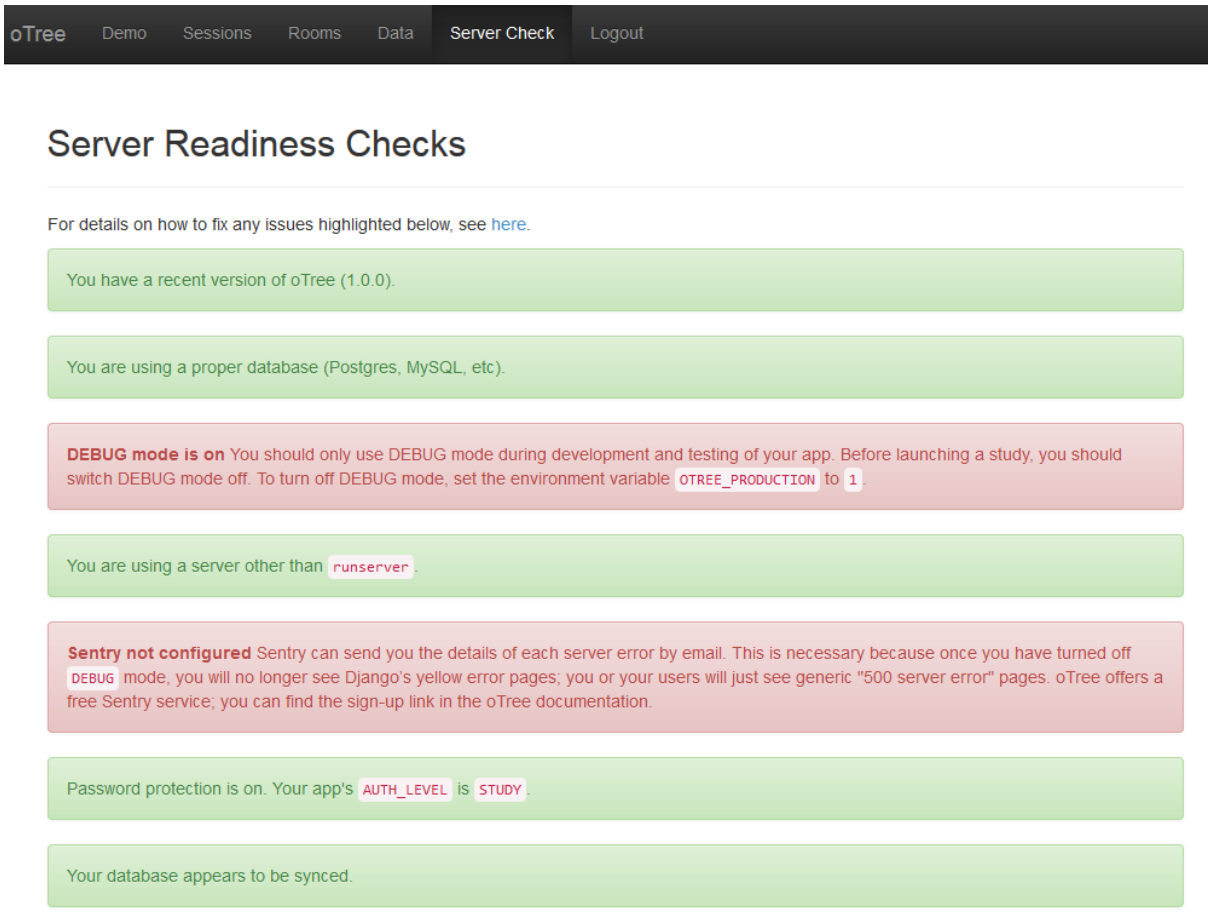
Rooms are not currently being implemented in the server as there was no need for such implementation. If you want to find out more about Rooms, have a look at this link. http://otree.readthedocs.io/en/latest/rooms.html

Data

Allows you to download the data collected from experiments as a CSV or Excel spreadsheet.
To Download all the data from all apps, click on one of the links below “All apps”.
To Download documentation of each app, click on the Excel or CSV link for the respective app.
The Documentation can be downloaded from here to as a CSV file, this will show all the variables in models.py for each app, and it will show the comment we used with doc= inside models.py. This comment is also the app description. The data for a session is only available from the website when the session can still be found within the Sessions List. To keep the Sessions list clean, you can use Archive Sessions



Server Check



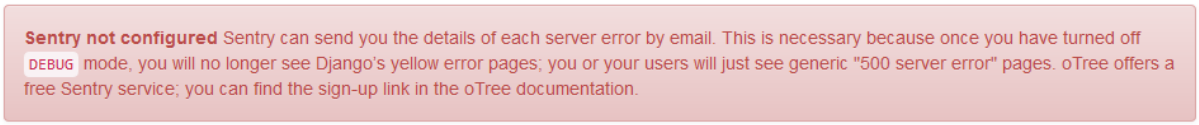
Indicates that the server has updated to the latest version of oTree

Indicates that server is setup with a database. The database currently being used is Postgres

Indicates whether DEBUG mode is on. DEBUG will be set to off when the product is fully launched

Indicates that the server is being run with otree runprodserver --port=80 which is recommended for when the product enters production.

Indicates that sentry is not being used. Sentry is a free service provided by oTree that sends server error messages via email. To sign up to sentry, click on this link: http://otree.readthedocs.io/en/latest/server/heroku.html#sentry



Indicates that the server is password protected

Indicates database is synced

Kiosk Mode

When working inside the labs all the time, it would be useful to set up a kiosk mode on a browser with prevents users from accessing the address bar, going back on the pages or accessing the console and elements of the web page via F12. We did not get a chance to install it, if you want to give it a go. Here's the link.

http://otree.readthedocs.io/en/latest/admin.html

Server Set Up

First thing you'll need to start your own server is a machine to work with, own server is running with Dual Cores, 4GB ECC RAM, 60GB Storage with room for expandability. The operating system can be anything, oTree server can be run on any Operating system that can handle Python and Django.

The instructions I provide will closely follow what is set in the oTree documentation mainly focusing on Windows Server Installation, the link for that is here:
http://otree.readthedocs.io/en/latest/server/intro.html

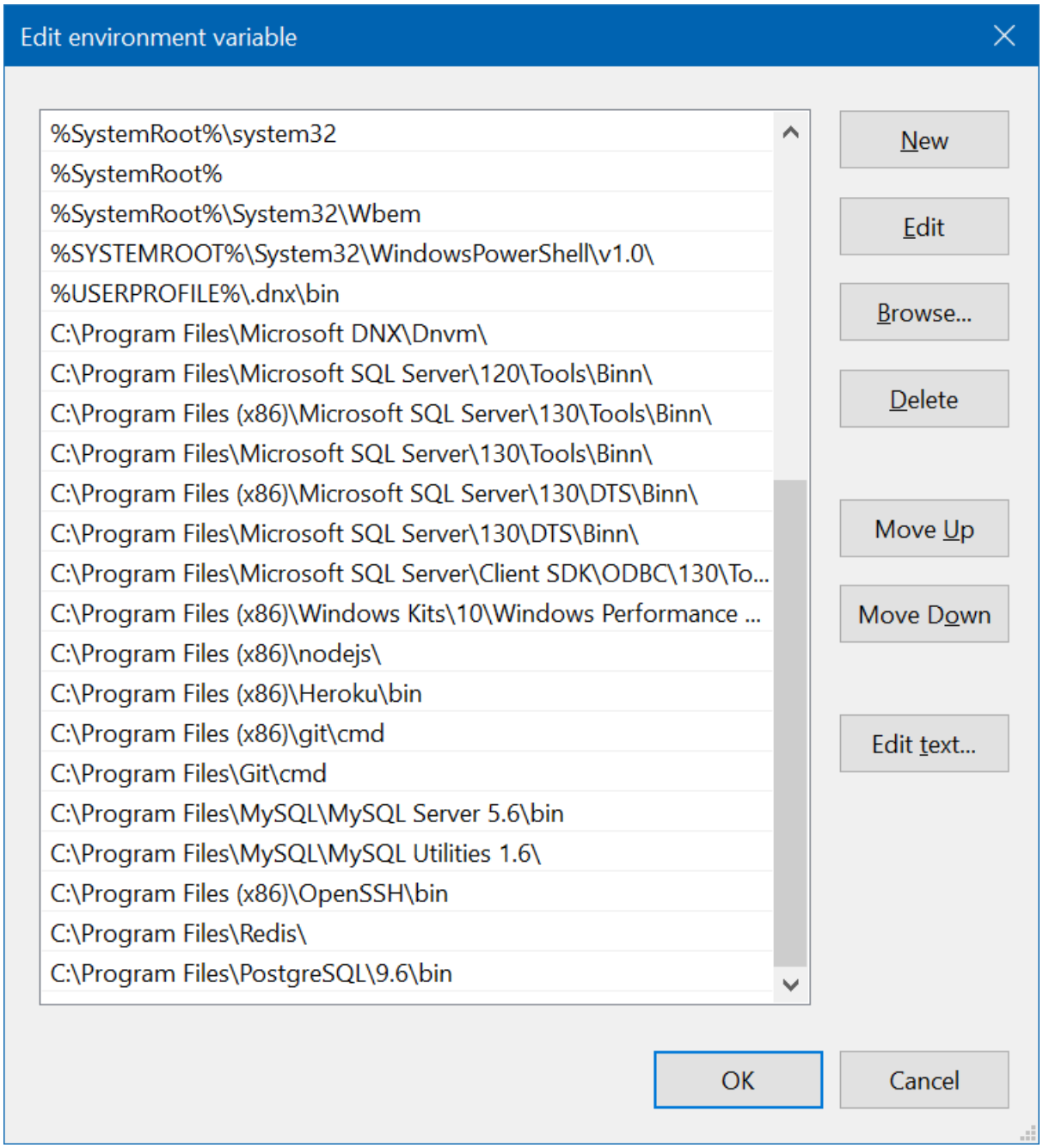
1) Install Python 3.5.2, Postgres 9.6, Redis

The links for their sites are below:
https://www.python.org/downloads/
https://www.postgresql.org/download/
Take note of your root Postgres password
The link for Redis, is for windows:
https://github.com/MSOpenTech/redis/releases

When installing Python, choose the option **ADD TO PATH**.

2) Install oTree and psycopg2

First access your environment variables and make sure that Python and Postgres are added to the PATH. Postgres is most likely not in there, to do this on windows search for environment variables and open the **System Environment Variables**. Look for PATH and then make sure Postgres is added to the PATH variable.



Once the variable has been set up, open your command line (it is best to do this as administrator):
pip3 install -U otree-core
pip3 install psycopg2

If psycopg2 doesn't work try this link: http://www.stickpeople.com/projects/python/win-psycopg/
Now while you have the command prompt open you should test Redis using the command:
redis-cli ping

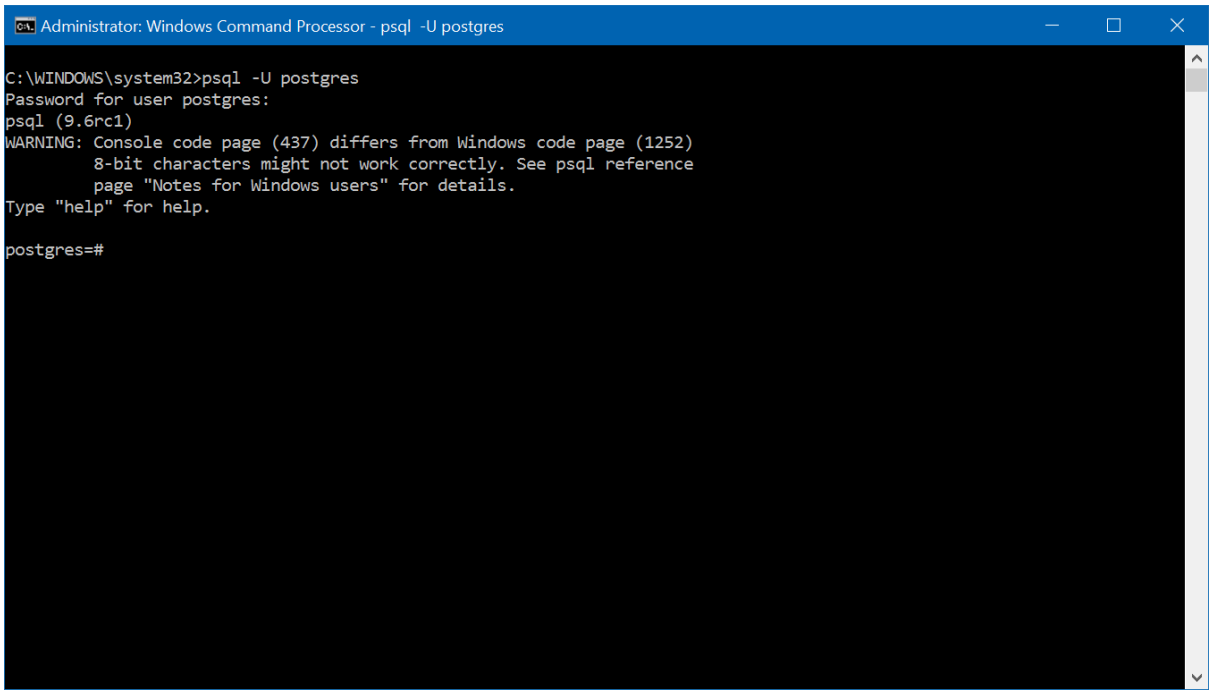
It should return **PONG**

3) Configuring Postgres

In your command line do the command:
psql -U postgres

Type in the password you used during the setup of Postgres.

It should come up with the Postgres bash.



Use these commands:
CREATE DATABASE django_db;
CREATE USER otree_user WITH PASSWORD '<your password here>';

GRANT ALL PRIVILEGES ON DATABASE django_db TO otree_user;

Do these commands but change the password to something you'll remember.
Now another thing to do here is to make otree_user the owner of the django_db database and grant otree_user permissions to create a Database. You'll need to do this if you intend of running tests using bots within the command line.

ALTER DATABASE django_db OWNER to otree_user;
ALTER ROLE otree_user WITH CREATEDB;

Now exit postgres with:

\q

Now go back to your System Environment Variables, if you can't remember how to do that, refer to 2) Install oTree and psycopg2. You'll need to create a new System Environment Variable called DATABASE_URL, you must give it the value:

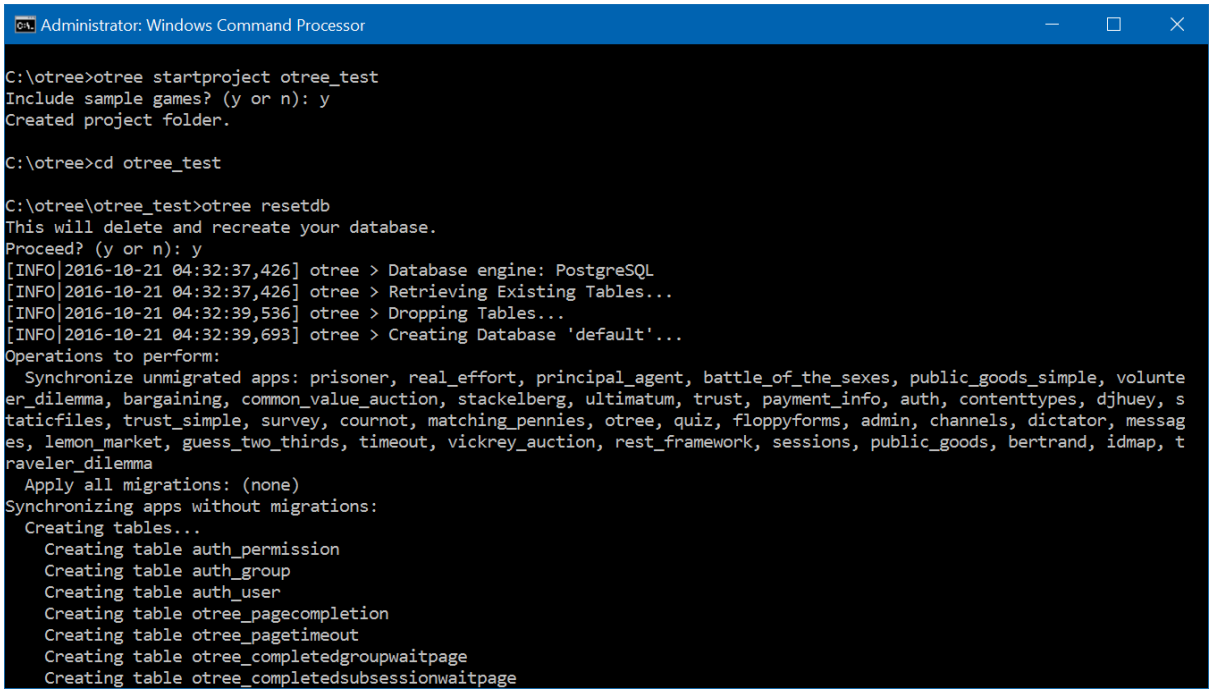
postgres://otree_user:<your password here>@localhost/django_db

This ensures that when oTree looks for a database to use, it searches for django_db a postgres database. **Upon testing it will create another database test_django_gb**.

4) Preliminary Checks

If you have done the above settings, you should be able to navigate to some directory inside your command line and use:

otree startproject otree_test
(It'll prompt for the sample games, type 'y' and hit enter)
cd otree_test
otree resetdb
(It'll prompt for the reset, type 'y' and hit enter)



If it gets through these tests, then oTree is ready on your system.

5) Run The Server

Now let's say you've followed my instructions to the letter and have made your own code, and have a website URL that you can use that's pointed towards your server's I.P Address via DNS (If you didn't understand that you may need to get someone to help you). If you meet those requirements, then try:

otree runprodserver –port=80

This will run the server for production, it will automatically run otree collectstatic for you, and since it runs on port 80 (which is HTTP) it will automatically route to your server when you type in the domain in a web browser.

Now close that for a moment, we need to make sure your oTree server is truly ready for its production stage.

You need to make System Environment Variables:
OTREE_AUTH_LEVEL with value **STUDY** or **DEMO**
(STUDY, restricts admin page to login only, **DEMO** allows any user who comes across the page to access the demos on the Demo page)
OTREE_PRODUCTION with value **1**
OTREE_ADMIN_PASSWORD with value **<another memorable password>**

Now, restart the command line and navigate to your oTree Server directory
And run:

otree runprodserver –port=80

add **–botworker** if you're server has browser bots enabled.

Make sure port 80 is allowed through your firewall and port forwarded in your router if necessary.

Congratulations, you have yourself an oTree server running on Windows!

6) Server Maintenance

Some advice for Maintenance, try to configure a .bat file for automatic maintenance of the server and run it weekly or fortnightly with Task Scheduler so you don't have to keep an eye on it. Try not to upgrade oTree on the server too much, updates could break your code!