
oTree Manager Documentation

Christian Koenig-Kersting

Jan 29, 2019

Contents:

| | | |
|----------|----------------------------|-----------|
| 1 | Features | 1 |
| 2 | Demo | 3 |
| 3 | Code | 5 |
| 4 | Paper | 7 |
| 4.1 | User Guide | 7 |
| 4.2 | Super-User Guide | 12 |
| 4.3 | Installation | 16 |
| 5 | Indices and tables | 19 |

CHAPTER 1

Features

- create and manage fully independent, production-ready oTree instances
- intuitive web interface for lab managers and experimenters
- Heroko-ish deployment of experiments with Git
- easy to remember URLs, no manual port configuration
- automatic configuration of PostgreSQL, Redis, and environment variables for production
- written in Python, built on industry standards (Docker, Dokku, Django)
- less resource intensive than virtual machine setups
- Lobby feature for easy experiment administration

CHAPTER 2

Demo

<https://demo.otree-manager.com>

CHAPTER 3

Code

https://github.com/otree-manager/otree_manager

If oTree Manager made running experiments easier for you, please consider citing the accompanying paper when publishing your results:

König-Kersting, C. (2019) [oTree Manager: Multi-user oTree installations made easy](#). Working Paper.

4.1 User Guide

4.1.1 Setup

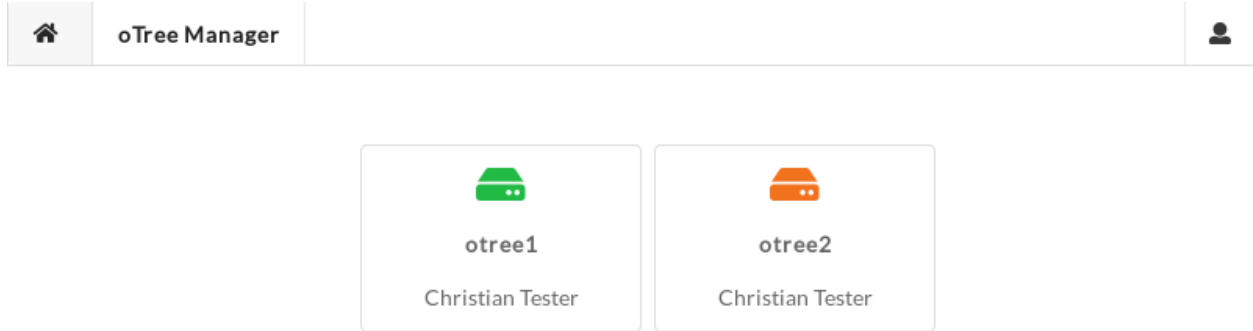
As a user, ask a super-user to set up an account for you. S/he will need your name and a valid email address. Once the account has been created, you will receive an email with your username. It will ask you to set your password. After you have completed this step, you can login to the web interface.

4.1.2 First Login

Upon first login, you will be asked to set your public Secure Shell (SSH) deployment key. Setting this key is necessary for deploying your experiments. It serves two purposes: The key allows oTree Manager to identify users when they are uploading an experiment. Thus, oTree Manager can make sure that you can only upload experiments to oTree instances that belong to you. Furthermore, setting the SSH key removes the hassle of handling user names and passwords.

The web interfaces contains instructions on how to find and uploading your public SSH key. It also tells you how to generate a key if it is not yet present on your system.

4.1.3 Dashboard



After logging in, you are presented with a dashboard which shows all oTree containers currently assigned to you. Green symbols indicate that an oTree instance has been created and that the container is up and running. Orange symbols indicate that oTree is not currently deployed on the respective container. Clicking on any container brings you to its detail screen.

The house icon in the top left corner always brings you back to the dashboard. The user icon in the top right corner opens a menu which allows you to navigate to password change and SSH key change pages. It also offers an option to logout.

4.1.4 Detail View

This is the main management screen for your oTree installation. It gives you all the details on its status and its configuration. The screen also provides access to all management features, including setting the oTree admin password, resetting the database, and configuring the Lobby feature.

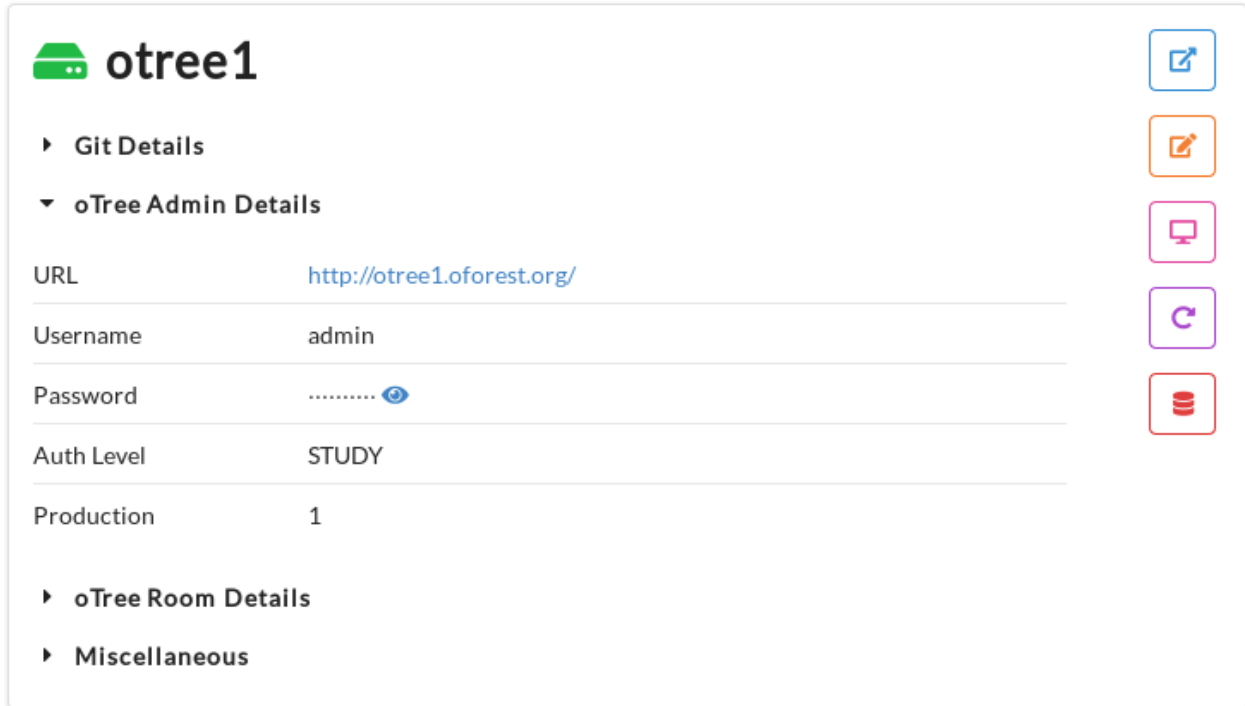
Multiple folding sections make up the biggest part of the detail box. Depending on the state of the instance, different sections will open by default.

Git Details




This section shows the repository URL you need to add as a Git remote in order to push your experiments to the instance. It also shows a ID of the last commit for reference.

oTree Admin Details



The screenshot shows the 'oTree Admin Details' section for an instance named 'otree1'. The page features a sidebar with navigation icons and a main content area with a table of configuration details.

| Field | Value |
|------------|---|
| URL | http://otree1.oforest.org/ |
| Username | admin |
| Password |  |
| Auth Level | STUDY |
| Production | 1 |

Below the table, there are three expandable sections: 'oTree Room Details', 'Miscellaneous', and 'oTree Admin Details' (which is currently expanded).

This section gives you the URL of your oTree instance, its admin username and password. It also reports on the currently activated auth level and whether debug mode is off (Production == 1).

oTree Room Details

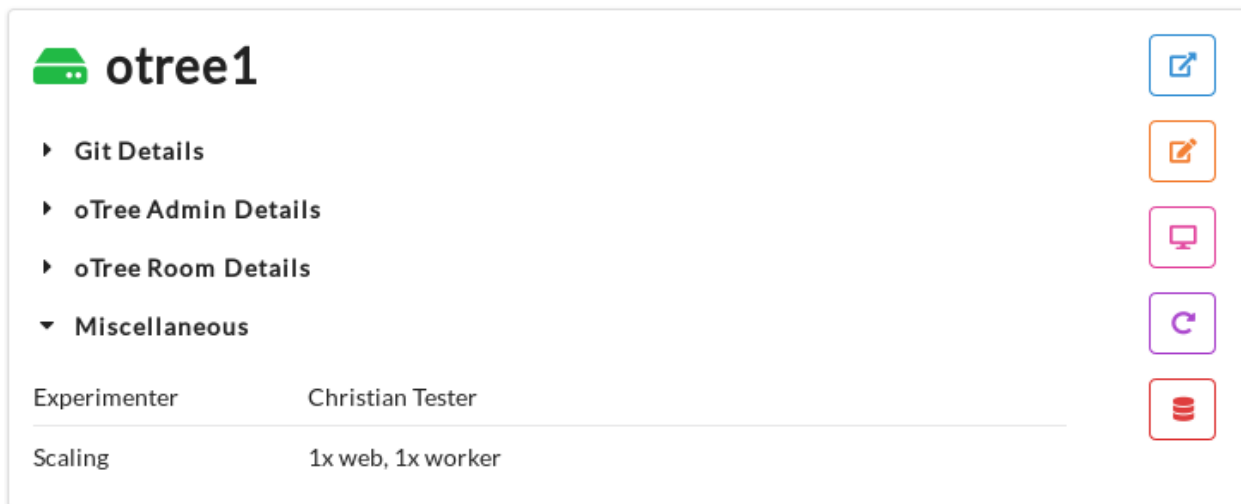


The screenshot shows the 'otree1' room details page. At the top left is a green house icon with two windows, followed by the text 'otree1'. To the right of this header are five vertically stacked icons: a blue square with a white arrow pointing up and right, an orange square with a white pencil, a pink square with a white monitor, a purple square with a white circular arrow, and a red square with a white database cylinder. Below the header is a list of expandable sections: 'Git Details', 'oTree Admin Details', 'oTree Room Details' (which is expanded), and 'Miscellaneous'. The expanded 'oTree Room Details' section contains a table with the following information:

| | |
|----------------------|---|
| Room Name | awi_lab |
| Participant Labels | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 |
| Lobby URL | http://oforest.org/lobby/otree1/ |
| Download Lobby Sh... | Chrome:    |

Here you can find the name of the room (from your settings.py) that is currently configured to be served by oTree Manager's Lobby feature. It also shows the corresponding participant labels. Finally it gives you the URL of the lobby, as well as allows you to download desktop shortcuts for each participant label. These shortcuts, which are available for all major platforms, open the Chrome (or Chromium) web browser in kiosk mode and direct it to the lobby page for the respective participant (label).

Miscellaneous



The screenshot shows the 'otree1' miscellaneous details page. It features the same header and icons as the previous screenshot. The 'Miscellaneous' section is expanded, showing a table with the following information:

| | |
|--------------|-------------------|
| Experimenter | Christian Tester |
| Scaling | 1x web, 1x worker |

This section shows who the instance belongs to as well as information on the scaling, that is, how many web and worker processes are currently available for the instance.

4.1.5 Actions

On the right hand side, there are buttons for these actions:



Open oTree Web Interface in a new tab.

This button is only visible if experiments have been uploaded.



Change the admin password of the oTree admin interface.



Setup a room configured in oTree to be used with oTree Manager's Lobby feature.



Restart the container.

This will prompt you confirm your action.



Reset the oTree database.

This will prompt you to confirm your action.

4.1.6 Lobby

When running experiments in a laboratory setting, experimenters often want to setup the room and workstations before participants enter the room. Typically, the experimenter boots the computers and opens the experimental software. In the case of oTree, this means starting a browser (in kiosk mode) and navigating to the correct session's URL. oTree's room feature caters to this use case as it provides persistent links to a waiting page. These links can be used in combination with participant labels to easily map workstations to data collected in the experiment for payment purposes.

The room feature also allows to open browsers on all workstations and have the clients wait for the session to start. However, many times the exact number of participants in a session is ex-ante uncertain. Participants may fail to show up without prior notice. In these cases, if fewer participants are present than anticipated, the experimenter has to manually close the browsers on unused workstations so they do not take part in the session. This is especially important in experiments which rely on group matching, as participants might end up being matched with an unused client. Importantly, the 'present' counter on oTree's room admin page is also not helpful in checking for the correct number of participants if the links have been opened on all client computers beforehand.

oTree Manager's Lobby feature avoids the manual intervention by the experimenter. Building on the room feature of oTree, it provides individual links for each participant label. If opened, the screen greets the participant and asks him/her to simply click a button once they have settled in at their cubicle. Clicking the button forwards the participants to oTree's actual room waiting page.

This setup has multiple advantages: First, the 'present' counter on oTree's room admin page is meaningful, as only those clients show up, which have participants who clicked the 'ready' button. Second, because of the additional step necessary to get to the room's waiting page, there is no need to manually close the browser on unused workstations. They will not be connected to your oTree session.

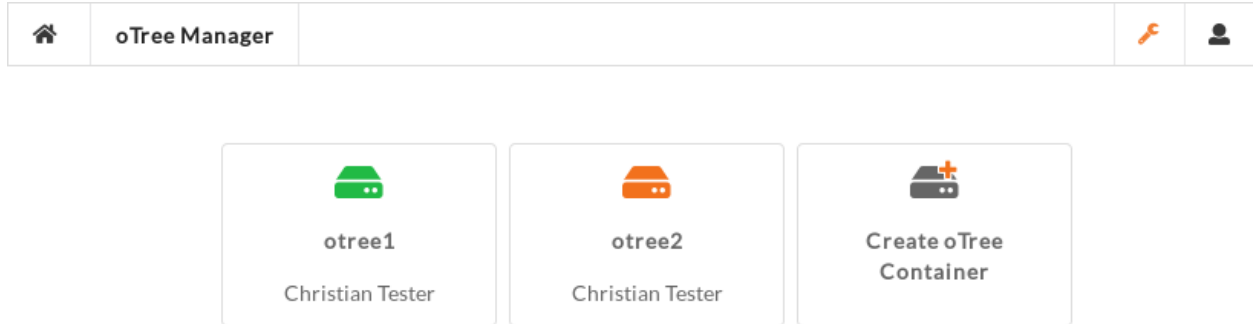
If a room has been configured for use with the Lobby feature, the detail view provides a link to the Lobby Overview, which shows all lobby links in a convenient grid. It also provides download links for desktop shortcuts for each client. The shortcuts expect the Chrome or Chromium browsers to be installed. The browser is both modern and comes with an integrated kiosk mode. The shortcuts are available for all major operating systems.

4.2 Super-User Guide

4.2.1 Overview

Super-Users, also called administrators or lab managers, have full administration and management permissions. They can do everything users can, and more.

4.2.2 Dashboard



For super-users, the dashboard shows all active oTree containers, independent of the assigned experimenter. In addition, it shows an icon to create a new oTree container. It also features an orange wrench icon in the top right corner, which opens a menu to quickly navigate to different management pages.

4.2.3 Actions

On the detail view of an oTree container, super-users have two additional actions compared to normal users.



Set the number of web and worker processes.



Delete the oTree container.

This will prompt you to confirm your action.

4.2.4 User Account Management

Add new user account

Add User

First name:

Last name:

Username:

Email address:

Superuser status:

Create

Create new experimenter accounts by clicking the wrench icon and choosing “add user”. Then, fill in the form with the user’s first and last names (used for identification throughout the web interface). Additionally, provide a username. This has to start with a letter and may only contain letters and numbers. Finally, provide an email address and choose whether the new user should also have super-user privileges.

On creation of the user account, oTree Manager will automatically send an email to the new user and ask him/her to set their password.

Edit user account

| | | | |
|---|-------|---|---|
|  | Users |  |  |
|---|-------|---|---|

| Last Name, First Name | Admin |
|-----------------------|-------|
| King, Christian | ✘ |
| König, Christian | ✔ |
| Tester, Christian | ✘ |

All existing users show up in the user list available from the wrench icon menu. It provides privilege information at a glance. Click on a user's name to edit their details. User accounts can be deleted. To do so, first remove all assigned oTree containers. Then, choose to edit the respective user account. A 'delete' button will be visible.

4.2.5 Container Management

Create new container

Add Container

Name:

Experimenter:

Experimenters ▼

Create

To create a new oTree container, click the button on the dashboard or find it in the wrench icon menu. oTree containers need a URL-safe name which you should choose to be as short as possible, as it will be part of the instance's URL. Select the experimenter the container belongs to and click 'create'.

Upon creation of a new container, several processes start running in the background. You will get periodic updates about the status of these processes on the web interface. First, the oTree container is initialized and write permission to the associated Git repository are granted. oTree Manager then starts two docker containers and links them to the instance. One container provides a PostgreSQL database, the other a Redis database. Once these are up and running, the oTree instance is pre-configured for production use. That is, debug mode is turned off, the authentication level is raised to STUDY, and a random password is generated and set for the admin account.

4.2.6 Site Management

Imprint information

You should customize the imprint page to display the information required by your jurisdiction. This typically includes the name of the person responsible for running the website as well as their contact details.

Super-Users can reach the dedicated imprint page editing form from the wrench-menu. Text formatting is done through standard HTML. Special characters are NOT escaped automatically.

Privacy statement

Again, you should customize the privacy statement page to display the information required by your jurisdiction. This typically includes information on which and how data is collected, how it is processed, and who can be contacted to request removal of any stored data.

Super-Users can reach the dedicated privacy page editing form from the wrench-menu. As with the imprint page, text formatting is done through standard HTML. Special characters are NOT escaped automatically.

4.3 Installation

4.3.1 Overview

oTree Manager uses Docker to containerize both databases and the actual oTree installations. Container management is handled by Dokku. The web interfaces is written in Python (3) and builds upon Django in combination with Django-Channels for running background tasks. The web interface requires PostgreSQL and Redis databases to store its data and handle user sessions. Supervisor keeps the web interface as well as the background task worker running. Nginx serves as a reverse proxy for the oTree installations and handles routing of requests between the oTree instance containers and the web interface.

Currently, I only recommend installations on Debian-based operating systems and write the installation instructions for a fresh installation of Debian 9.

Note that Dokku, a core dependency, does not support macOS or Windows operating systems. Thus, oTree Manager cannot be installed on these systems either.

4.3.2 Intended Architecture

oTree Manager should not be installed on the typical experimenter's computer in the laboratory. It should be installed on its own dedicated workstation or server-grade hardware. This is because of multiple reasons. First, oTree Manager is intended to serve its web interface and oTree containers continuously '24/7'. Most consumer-grade hardware is not designed with this use case in mind. It is also typically not designed for virtualization-heavy workloads which require many CPU cores and large amounts of system memory for optimal performance. Second, setting up oTree Manager on its own machine provides a natural separation from existing legacy setups. That is, it can run in parallel to hardware required to support z-Tree and other experimental software tools. It can also be phased-in in parallel to current oTree setups. This reduces down-times and allows for a step-wise migration to the new system. A separate machine also has the advantage that security measures can be implemented more easily. With oTree Manager serving requests from the 'open' Internet, it is advisable to implement more stringent security (firewall) policies compared to typical lab computers, which often only serve locally. If in doubt, ask your university's IT department for assistance.

4.3.3 Installation

Note: The installation instructions assume a clean Debian 9 installation to start from.

First, make sure to update all system packages to their latest versions and install git:

```
sudo apt update
sudo apt upgrade
sudo apt install git
```

Then, we install oTree Manager using its installation script:

```
git clone https://github.com/otree-manager/otree_manager_installation.git
cd otree_manager_installation
sudo bash ./otree_manager_setup.sh
```

Head over to GitHub to take a look at the installation script. https://github.com/otree-manager/otree_manager_installation

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`