oTree Setup

Corey Scheinfeld

December 22, 2021

Contents

1 Installation

To install the package on your own computer:

1. Download and install Python at `https://www.python.org/downloads/`. Note that oTree requires Python 3.7 and will not run properly with other versions.

   - If a higher version of Python is already installed, use the command

     ```
     python3.7 –m pip install
     ```

     instead of

     ```
     pip3 install
     ```

     for following instructions.

2. Open the Terminal app and type the following:

   ```
   pip3 install –U otree
   ```

3. Navigate to `https://github.com/corey-scheinfeld/gametheorysims_final.git` and download the games.otreezip file or, if you have Git installed, clone the project with

   ```
   git clone https://github.com/tenzorp/gametheorysims.git
   ```

   If you downloaded games.otreezip, unzip the file with

   ```
   otree unzip games.otreezip
   ```

4. Change directories to the 'games' folder. 'gametheorysims' is the parent of the 'games' project folder.

   ```
   cd \Path\to\folder
   ```

   If you are unsure of the exact path, type 'cd ', drag the folder from your Finder window to the Terminal window, and then hit enter.

5. Install any dependencies with

   ```
   pip3 install –r requirements.txt
   ```

6. Start the development server by typing the following:

   ```
   otree devserver
   ```

7. Navigate to the demo page at `http://localhost:8000`.

2 Creating/Editing Games

2.1 oTree Studio (Recommended)

oTree Studio is a point-and-click interface for creating and editing games provided by the makers of oTree. This method is recommended for users with no programming experience, as it requires no familiarity with Python. Some features of oTree are not supported by oTree Studio, so more complex games may need to be created manually. Examine the documentation to determine whether oTree Studio can meet your project's needs. This method does require you to make an account with oTree.

2.2 Python (For those with coding experience)

To edit existing apps: Follow the steps in the Installation section of this guide and use your preferred text editor to edit game files. Each game's code is contained in an appropriately named folder within 'games'. Global assets (images and CSS files) are placed in '_static'.

To create your own oTree project:

1. Follow the first two steps of the Installation section of this guide. Open your Terminal and create the project folder (replace 'newProject' with name of your choice):

   ```
   otree startproject newProject
   ```

2. Change directories to your new project folder:

   ```
   cd newProject
   ```

   and create your first game (replace 'myApp'):

   ```
   otree startapp myApp
   ```

   Follow steps 5 & 6 of the Installation section to run the development server. You can now edit files generated for you in the app folder to create your game. To see your game on the demo page, you must add an entry to SESSION_CONFIGS in 'settings.py', e.g.:

   ```
   SESSION_CONFIGS = [
   {
       'name': 'myApp',
       'display_name': 'My App,
       'num_demo_participants': 2,
       'app_sequence': ['myApp'],
   }]
   ```

When in doubt, consult the documentation at `https://otree.readthedocs.io/en/latest/index.html`.

3 Server Setup

1. Use the above instructions to install the Game Theory Simulations package from Github.

2. Create a Heroku account and install the Heroku toolbelt from the command line. With homebrew, this appears as follows:

```
brew tap heroku/brew && brew install heroku
```

3. Open terminal and cd to your project folder. Within this, cd to the "games" folder

4. Initialize git

```
git init
```

5. Log in using the email address and password you used when creating your Heroku account:

```
heroku login
```

6. Create the Heroku app. This will create your website my-app-name.herokuapp.com.

```
heroku create my–app–name
```

7. Install Redis add-on using the following:

```
heroku addons:create heroku–redis:premium–0
```

8. Upgrade Otree

```
pip3 install –U otree
```

9. Run:

```
otree ––version
```

10. Open requirements_base.txt within your project folder and replace whatever is in that file with this single line, based on your current oTree version:

```
otree>=X.X.X
```

11. Run:

```
Python ––version
```

12. Open 'runtime.txt' within your project folder and replace whatever is in that file with this single line, based on your current Python version:

```
python–X.X.X
```

13. Commit your changes:

```
git add .
git commit –am "your commit message"
```

14. Push your code to Heroku

```
git push heroku master
```

15. Reset the oTree database on Heroku

```
heroku run "otree resetdb —noinput"
```

16. Open the site in your browser

```
heroku open
```