

Documentation: oTreeZTS

Quick Demo	2
Setup & Installation	2
Python Installation	2
oTree Installation	2
Download Source Code	2
Starting Application on local testing server	3
User Guide	3
ZTS App	4
Instructions Page	4
Start Page	4
Trading Page	4
Results Page	6
Survey	6
Administrator Guide	6
oTree basics	6
Overview	6
Setup a Session	6
Setup a Room	7
Session Menu	7
ZTS Specific	8
Experiment Data	8
Session Configurations	9
Payoff	11
Adjust Page Content	12
Deployment	12
MTurk	12

Quick Demo

This section shows how to quickly run a demo session of the oTree ZTS application, which is intended only for demonstration purposes (For a customizable experiment we need to set up a session).

1. Open the following URL in a browser: <https://zts.otree.ethz.ch/>
2. Click on the “ZTS” button to open a demo session.
3. Click on the provided link under Single-use links and the session should start in a separate tab.

Setup & Installation

These instructions will get you a copy of the Zurich Trading Simulator project up and running on your local machine. Please note that you should only use this setup for testing and development! If you want to run real experiments please read through the [Deployment](#) section that will explain in more detail how to set up a stable version on a live system.

Python Installation

Make sure that Python is installed on your machine before continuing with the next steps by issuing the following command in the terminal and checking the response (version should ideally be at least 3.7 or higher):

```
$ python3 --version
Python 3.7.5
```

NOTE: If python is not yet installed follow the steps for your specific operating system listed on the following website: <https://realpython.com/installing-python/>

Setup Virtual Environment

It is recommended to run your setup inside a Python virtual environment to avoid any requirement conflicts with other Python projects you might be working on.

```
$ python3 -m venv env
```

Then to activate/deactivate the virtual environment type the following commands (please leave it activated for now). When restarting your machine you might have to reactivate the virtual environment before you can continue:

```
$ source env/bin/activate
$ deactivate
```

NOTE: Venv should be automatically installed together with python, if it is still missing for some reason please execute the following command to install it separately:

```
$ sudo apt install python3-venv
```

Download Source Code

Download the zip of the source code of the Zurich Trading Simulator (oTreeZTS) on github (<https://github.com/Zurich-Trading-Simulator/OtreeZTS>) by clicking on Code/Download ZIP. Extract the ZIP file into your working location (/Documents/ or wherever you prefer). People that are familiar with github should instead just clone the repository to their working directory.

Alternative: Download the .otreezip file of the oTreeZTS directly on the oTreeHub (<https://www.otreehub.com/projects/otree-zts/>) into your working directory and unzip it using the following command:

```
$ otree unzip /path/to/workingdirectory/otree-zts.otreezip
```

Starting Application on local testing server

To start the application locally for development and testing purposes follow the steps listed below:

1. Open your terminal and change to the oTreeZTS directory we set up previously.

```
$ cd /path/to/workingdirectory/otreeZTS/
```

2. Activate your virtual environment if not done yet.

```
$ source env/bin/activate
```

3. Install the python dependencies required for the application by running the following command. This will install the python libraries that contain functions the code is using. This command only needs to be run once in the beginning to install all the required libraries.

```
$ pip3 install -r requirements_base.txt
```

4. Start the application on a local testing server with the following command:

```
$ otree devserver
```

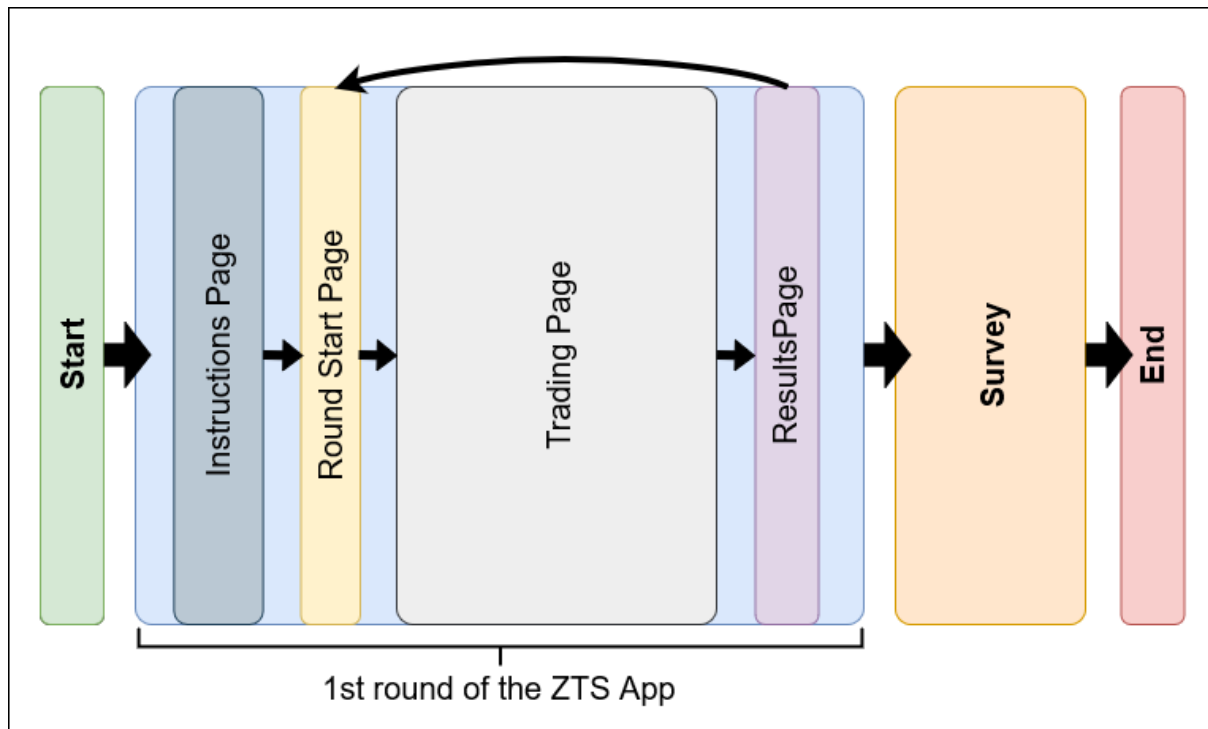
5. Open your browser to <http://localhost:8000/> and the application should appear.

When you make changes to the code, save the changes with `ctrl + s` and the running application in the browser will automatically adjust.

User Guide

In this section we will look at how a participant should play the apps of the oTreeZTS experiment and how the interface is set up. The oTreeZTS consists of two apps, the actual ZTS (trading) app and the Survey. In oTree an app is just a programmatically independent

module of the entire experiment. The ZTS app can consist of multiple rounds, you can set the number of rounds in the session configurations as shown [here](#). In each round the player's holdings are reset to the initial endowment.



ZTS App

Instructions Page

The session participation links will redirect you directly to this page. It is currently empty, but is intended to be filled with instructions for the participants to clarify the setup and workflow of the experiment. Text can be added to this page under `/ZTS/templates/ZTS/InstructionPage.html`

The Instructions Page is only displayed at the start of the first round, in later rounds it will not be shown again.

Start Page

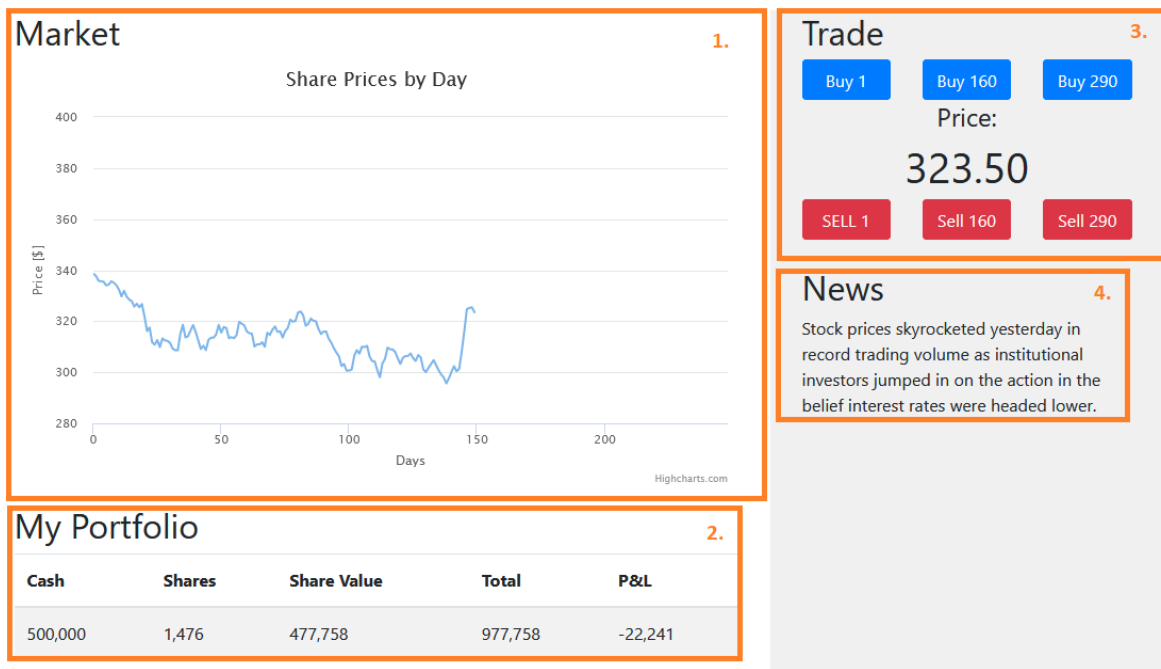
This page shows which round we are in and serves as a preparation page before this round's trading game starts. To alternate the text shown in this page edit `/ZTS/templates/ZTS/StartPage.html`

Trading Page

On the trading page the actual trading game takes place, each participant sees a period of fictional or historical prices of a single share or an index. At the beginning, the share prices of the first day are shown and after the refresh rate the share prices of the second day are

added and so on. Each day the player can buy or sell shares according to his holdings and the current share price.

His **holdings** are set up according to the cash and share endowments specified in the sessions config for this round. It is only possible to buy/sell full shares (minimum buy is 1 share), we do not provide the possibility to trade with partitions of assets.



- Prices Chart:** The chart shows the prices from day zero until the current day and after the time period of the refresh_rate it gets updated with a new price. It helps the player to understand how the asset has evolved in the past and makes it possible to make predictions on its future course.
- Portfolio:** In the portfolio we have metrics on the holdings and the current trading performance. It helps the player to assess how well his/her strategy is working up to this point in time.
- Trading Interface:** This panel is used for the buying and selling of shares. There are three buttons with different quantities for both the buying and the selling actions. The quantities are assigned before the start of the game according to the values specified in the session config for this particular round:
- News Panel:** If the time series file lists a column that provides news related to the market situation at some days, this will be shown in the news panel. The goal of it is to give the players further relevant information on the shares other than just the past prices. The news panel can hold up to 500 characters, so make sure the texts in the time series file are limited to this constraint.

Results Page

This page shows the final portfolio and performance metrics of the current round.

Survey

The survey part consists of a single page with a “Proceed to Survey” button that redirects the players to a survey under the link specified in `survey_link` from the session configurations. The survey can also be directly implemented in oTree, however we found it more useful to use an elaborate survey software (e.g. qualtrics).

Administrator Guide

In this guide we will mainly cover the specific administrative functionalities for the oTreeZTS application, but we will also look at some basic oTree functionalities. However, to find more in-depth information on the oTree features please visit <https://otree.readthedocs.io/>.

oTree basics

Overview

In oTree, a **session** is another word for the entire experiment using the same or at least similar application configurations. An example of a session in the case of the oTreeZTS would be:

“A number of participants will come to the lab to play several rounds of the ZTS trading game, followed by a questionnaire. Participants get paid some amount of money for showing up, plus their earnings from the games.”

A session is a series of **subsessions**; subsessions are the “sections” or “modules” that constitute a session. For example, in our case each trading round and also the post task questionnaire would be a separate subsession.

Each subsession can be further divided into **groups** of players; for example, you could have a subsession with 30 players, divided into 15 groups of 2 players each.

In oTree, the terms “**player**” and “**participant**” have distinct meanings. The relationship between participant and player is the same as the relationship between session and subsession: A player is an instance of a participant in one particular subsession. A player is like a temporary “role” played by a participant. A participant refers to the same physical person during the whole session, he can be player 2 in the first subsession, player 1 in the next subsession, etc.

Setup a Session

To set up an experiment session follow the below steps:

1. Click on “Sessions” on the navigation bar on the top of the startpage.
2. Click on “Create New Session” or click on an existing session to continue with an existing experiment.
3. Choose the “ZTS” session config and type in the desired number of participants. The session config specifies the course of the experiment, namely Trading Game ->

Survey, but we can also choose the Survey in the beginning by adding a new session config. You can find detailed instructions on how to do this [here](#).

4. On the same page click on “Configure Session” and change the default configurations as desired. You can find A full list of the configurations and their meaning [here](#).
5. Click on “Create”

Now you will be redirected to the Session Menu, where you will be able to see the session links that can be used to distribute to the participants to start the experiment. You can find more information on the Session Menu [here](#).

Setup a Room







OTree offers the possibility to open virtual experiment rooms, that offer some additional features than normal sessions:

- Links that you can assign to participants or lab computers, which stay constant across sessions
- A “waiting room” that lets you see which participants are currently waiting to start a session.
- Short links that are easy for participants to type, good for quick live demos.

More information on Rooms and how to set them up you can find on the oTree documentation: <https://otree.readthedocs.io/en/latest/rooms.html>.

Session Menu

ZTS: session **brdwgn1j**

 Description  Links  Edit  Monitor  Data  Payments

You can either use the session-wide link, persistent links, or single-use links.

Session-wide link

If it is impractical to distribute distinct URLs to each participant, you can give the following start URL to all 1 participants.

<https://zts.otree.ethz.ch/join/pga3whc7jk/>

Note: unlike the other link modes, this does not prevent the same person from playing twice.

Persistent links

If you want to give your participants permanent links that don't change, you should create your session in a [room](#).

Single-use links

Below are single-use links, which you can distribute to your participants. Each link has a unique code for the participant.

P1	https://zts.otree.ethz.ch/InitializeParticipant/qc98tmc5
----	---

- **Description:** provides a description of the sequence of apps (e.g. ZTS, Survey) and some additional information on each of these apps.
- **Links:** On this page the experiment start links for the participants are listed. There are 3 types of links that are shown:

- **Session-wide link:** A single link that can be used by all the participants to open their experiment. It will automatically redirect each participant to a new oTree Participant. This makes link distribution much simpler in some cases, but unlike the other link modes, this does not prevent the same person from playing twice
- **Persistent links:** permanent links that persist over time, this feature is only enabled when inside a room. You can find more information on Rooms and persistent links [here](#).
- **Single-use links:** A list of single-use links, each one of them has a unique code for the corresponding participant.
- **Edit:** This page lets you edit some of the configurations for this session also after its creation.
- **Monitor:** On this page an experiment administrator can monitor the experiment including the status and progress of each participant. It also allows you to enforce a page advancement for particularly slow participants.
- **Data:** Here you can track the live experiment data during the session, this is very useful for debugging.
- **Payments:** Here you have a report of the payments, each participant has earned during this experiment. You can find more information on how the payoff is calculated or how this calculation can be altered in [here](#).

ZTS Specific

Experiment Data

To download the experiment data click on “Data” on the navigation bar on the top of the startpage. For each App (ZTS, Survey) you can click on “Excel” (.xlsx) or “Plain” (.csv) to get a data file in the respective format that contains the entire data for each round of the app. For the ZTS app there is an additional data report, that contains the trading actions (BUY, SELL, START, END) and related data to the actions such as timestamp, which participant, how much he/she bought/sold. To open the trading action report click on the preferred file format under ZTS (custom):

- **BUY:** A participant bought an amount of shares.
- **SELL:** A participant sold an amount of shares.
- **START:** Marks a start of a round in the Trading game (day 0).
- **END:** Marks an end of a round in the Trading game (last day).

Note: In oTree the data reports usually contain the data of all sessions, when trying to access the data of a single session one needs to filter according to the session!

The oTree standard data fields and their meaning can be looked up in the oTree documentation (<https://otree.readthedocs.io/en/latest/models.html>). In the following there is a list of the data fields from both reports that are ZTS specific:

session	The id of the session to which this action belongs
round_nr	which round of the ZTS this action belongs to
participant	Who initiated this action
cash	The amount of cash a player has
shares	The amount of shares a player has
share_value	The amount of cash the holded shares are worth
portfolio_value	The worth of the holded portfolio (= cash + share_value)
pandl	Profits & losses of the current round
action	Trading Action (BUY, SELL, START, END)
quantity	If current action is a BUY or SELL, this specifies the amount of bought/sold shares
owned_shares	The amount of shares a player has (= shares), the "owned" prefix is added for clarification in the custom report.
cur_day	The market day, the action took place in
asset	Asset name (configurable in session configurations)
roi	Return of investment, percentage of portfolio_value at end w.r.t portfolio_value at start

Session Configurations

Each session can be customized to specific experiment settings at creation. In this section you will find a list of the configurable parameters and their meaning, instructions on how to change the time series file that determines the daily asset prices and some other configurability options listed such as changing your app sequence.

List of Configuration Parameters

session_name	The name of the session
survey_link	Link to your possession survey (e.g. www.qualtrics.com/)
timeseries_filename	List of the filenames for each round containing the timeseries data (e.g. ["demo_1.csv", "demo_2.csv"]). Also the length of this list will specify how many rounds the trading game will consist of. Note: The name of the asset in the data report will correspond to the filename specified here.
timeseries_filepath	The path where the timeseries files are stored, usually this should be _static/ZTS/timeseries_files/.

refresh_rate	List of durations of one market day in ms for each round (e.g. [500, 500])
initial_cash	Initial cash endowment for each round (e.g. [5000, 5000])
initial_shares	Initial shares endowment for each round (e.g. [10, 10])
trading_button_values	The values of the three buy/sell buttons for each round in the trading game. Usually the first button corresponds to buying/selling 1 share and the others are chosen s.t. the player has a reasonable choice of buying/selling a medium and large amount of shares (e.g. [[1, 10, 20], [1, 10, 20]])
random_round_payoff	If set to true, payoff will not be accumulated from all rounds but only from a single random round.
training_round	If set to true, the first round will be in practice mode, meaning that the payoff of this round will not count towards the final payoff.
graph_buffer	The amount of buffer on the top and bottom of the y-axis from the prices chart, where it triggers a y-axis resize, when a price ends up in the buffer. Default is set to 0.05 which corresponds to 5 % of the graph window
real_world_currency_per_point	The conversion rate used to get the converted payoff by multiplying it to the normal payoff

Change Time Series File

The time series file specifies the daily market prices and the news that are displayed in the trading game. To change this time series files follow the below instructions:

- **File Location:** The time series files should be located under:
`/_static/ZTS/timeseries_files/`
- **Replace or Add:** You can either replace the default files “demo_1.csv, demo_2.csv, demo_3.csv” by your custom ones or add new files to the folder with a different name and change the value of “timeseries_filename” from [“demo_1.csv”, “demo_2.csv”] to your own in the session configurations. By adding files you can e.g. create two experiment sessions that use different time series files.
- **File Naming:** ZTS will take the name of the timeseries file as the asset name that will be stored on each trading action in the custom data report.
- **File Format:** Make sure the files follow the same format as the default files, specifically they must both contain a column “price” (asset day prices) and a column “news” (the news displayed in the trading app). The column “date” is not required but can provide some information for yourselves. The file format should be .csv with a “,” as delimiter.

Adjusting App Sequence

The default app sequence is set to ZTS -> Survey, however this can be adjusted to e.g. Survey -> ZTS. To adjust the app sequence add a new Session_Configs under `/settings.py` by adding the code signalled with the `###`.

```
SESSION_CONFIGS = [
    dict(
        name='ZTS',
        num_demo_participants=1,
        app_sequence=['ZTS', 'Survey']
    ),

    #####
    #### Add the following Lines to add a new session config ####
    dict(
        name='ZTS_new'
        num_demo_participants=1,
        app_sequence=['Survey', 'ZTS']
    ),
    #####
]
```

Payoff

The payoff is the amount of money a player will be paid in the end. While the ZTS application will take over the entire payoff calculation for you, the payments still have to be done by yourselves. Remember that if `training_round` is set to true in the session configurations, the first round will have no effect on the final payoff. There are two methods for the payoff of the ZTS, it can be switched between these two methods through “Configure Session” at Session creation:

- **Random Round Payoff:** the payoff will be chosen from a random round (not training round).
- **Accumulated Payoff:** the payoff will be an accumulation of the payoffs of all rounds. This mode is applied if `random_round_payoff` is not enabled in the session configurations.

How is each round's payoff calculated in detail?

Remember: Player is an oTree object for a person in a single round while Participant represents the same person across all rounds. Each player has a payoff field. `Participant.payoff` automatically stores the sum of payoffs from all subsessions (if `random_round_payoff` is not selected).

In the ZTS the players payoff (meaning the payoff of a participant in a single round) is calculated as the **portfolio_value** at the end of the round (cash + share_value).

At the end of the experiment, a participant's total profit is calculated as follows:

$$total\ profit = participant.payoff * real_world_currency_per_point + participation_fee$$

How to change the payoff calculation?

To manipulate the payoff calculation you have to adjust the python function that calculates it. The function is located in `/ZTS/models.py` and gets called at the end of each round.

```
def set_payoff(self):
    """
    Set the players payoff for the current round to the total portfolio
    value. If we want participants final payoff to be chosen randomly from
    all rounds instead of accumulated (oTree standard) subtract current
    payoff from participants.payoff if we are not in round_to_pay
    """
    self.payoff = self.portfolio_value
    random_payoff = self.session.config['random_round_payoff']
    if(random_payoff and self.round_number !=
        self.participant.vars['round_to_pay']):
        self.participant.payoff -= self.payoff
```

Adjust Page Content

All Pages that are displayed in the oTreeZTS application are formatted using HTML and are located in `/ZTS/templates/ZTS/` or `Survey/templates/Survey` (there is a template directory for each registered App). In Addition to standard html syntax oTree also allows template syntax on their pages, as stated in <https://otree.readthedocs.io/en/latest/templates.html>. This can make it a lot easier to format your pages.

As an example below you can see the lines from `/ZTS/templates/ZTS/InstructionPage.html` that declare the Content of the Instructions page. Here we can rewrite any paragraphs and add additional structure elements.

```
{% block content %}

    <p>TODO: Add specific Experiment instructions here, this will only be
    shown before round 1!</p>

    <p>To change this page's content adjust the html of the
    /ZTS/InstructionPage.html file.</p>

    {% next_button %}

{% endblock %}
```

Deployment

Running the application locally, is enough for testing purposes, but please be aware that in order to perform any experiments the application needs to be run in a production environment, to guarantee safety and correctness. To deploy the oTreeZTS for production follow the oTree documentation steps under

<https://otree.readthedocs.io/ja/latest/server/intro.html>.

MTurk

OTree provides integration with Amazon Mechanical Turk (MTurk):

1. From oTree's admin interface, you publish your session to MTurk.
2. Workers on Mechanical Turk participate in your session.
3. From oTree's admin interface, you send each participant their participation fee and bonus (payoff).

More information on how to setup this application with MTurk you can find on the otree documentation <https://otree.readthedocs.io/en/latest/mturk.html>.