# Contributing to Visdom

We want to make contributing to this project as easy and transparent as

possible.

## Issues

Before you post an issue on our tracker, please check the following list of

issues to see if it resolves your issue. If this document does not resolve your

problem, please scroll all the way down for details on how to report an issue.

In all your interactions with us, please keep in mind that visdom is a side

project that we work on in our spare time. We are happy to help, but there are

no engineers dedicated to visdom so we cannot accommodate all your requests and

questions right away.

\*\*Issue: I cannot connect to the visdom server.\*\*

First, check that your visdom server is running. You can start the visdom server

via `python -m visdom.server`. Try restarting the server.

If your visdom server is running, but you don't see anything when trying to

access visdom in your browser, please check that your network settings don't

block traffic between the visdom server and your browser. Traffic may be blocked

by a firewall, or you may need to specify a proxy server when starting the

visdom server (via the `-proxy` option). In some cases, it may help to set up an

SSH tunnel to your server by adding the following line to your local

`~/.ssh/config`: `LocalForward 127.0.0.1:8097 127.0.0.1:8097`

It's also possible that the port is being blocked by your firewall, and some users have reported that the `sudo ufw allow 8097` command helps them.

\*\*Issue: I see a blue screen in my browser, but I do not see visualizations.\*\*

There may be an issue with downloading the JavaScript dependencies. This is,

unfortunately, a common issue for users in China. In Chrome, click `View ?

Developer ? JavaScript Console` to check for errors related to missing

JavaScript dependencies. If such errors appear, you can try to download and install

the dependencies manually:

\* Navigate to `/home/$USERNAME/$ANACONDA\_FOLDER/lib/python$PYTHON\_VERSION/site-packages/visdom-$VISDOM\_VERSION-py$PYTHON\_VERSION.egg`.

Note that the variables `$ANACONDA\_FOLDER`, `$PYTHON\_VERSION`, and

`$VISDOM\_VERSION` may not be set and depend on your configuration. Furthermore,

if you are installing from source or using another method of installing

dependancies, the folder to use may be different.

\* View the `download.sh` script and either execute it to automatically download the resources or manually download all the files that it requests.

\* Restart the visdom server, try again, and check the JavaScript Console to

confirm all dependencies are found.

\*\*Issue: I would like to make a plot that has feature X:\*\*

To produce visualizations, visdom uses [plot.ly](https://plot.ly/). Specifically,

the client code produces a JSON-structure that is passed on to plot.ly by the

server. This implies that, \_given the right input, visdom can display any

visualization that plot.ly supports\_. You can find an up-to-date guide to plot.ly

features [here](https://plot.ly/python/).

The visdom exposes easy access to the most common plot.ly features, but does not

expose all of them. You are more than welcome to hack the client code producing

the data structure (in `py/\_\_init\_\_.py`) to include the feature you want to use.

All available options for each plot type are described in [the plot.ly manual](https://plot.ly/python/).

You can even construct your own plot data structure from scratch, and [`\_send`](https://github.com/facebookresearch/visdom/blob/master/py/\_\_init\_\_.py#L247)

it to the visdom server directly.

If you believe a feature is so generally useful that it should be exposed

directly in the visdom client, please send us a pull request; we will happily

accept them!

\*\*Issue: I want to use a recently added visdom feature that is not in the pip version:\*\*

You can always install visdom from source. Clone the Github repo (and make your

own code changes, if any). In the visdom source folder, run:

```

pip uninstall visdom && pip install -e .

```

For some pip installs, this approach does not always properly link the visdom

module. In that case, try running `python setup.py install` instead.

## How to report an issue:

If you identified a bug, please include the following information in your bug report:

1. The error message produced by the visdom server (if any). Copy-paste this error message from your Terminal.

2. The error message produced by the JavaScript Console (if any). In Chrome, click View ? Developer ? JavaScript Console. Copy-paste any warnings or errors you see in this console.

3. The platform that you're running on (OS, browser, visdom version).

This information will help us to more rapidly identify the source of your issue.

## Pull Requests

We actively welcome your pull requests.

1. Fork the repo and create your branch from `master`.

2. If you've added code that should be tested, add tests.

3. If you've changed APIs, update the documentation.

4. Ensure the Lua and Python interfaces to Visdom are in sync.

5. If you change `js/`, commit the React-compiled version of `main.js`. For details, please see `Contributing to the UI` below.

6. Add demos for new features. Ensure the demos work.

7. Make sure your code lints.

8. If you haven't already, complete the Contributor License Agreement ("CLA").

## Contributing to the UI

The UI is built with [React](https://facebook.github.io/react/). This means that `js/` needs to be compiled. This can be done with `yarn` or `npm`.

To clarify an inconsistency, Panes in the UI are the containers for the

'windows' referred to by the Python and Lua APIs.

#### yarn

You can find instructions for install `yarn` [here](https://yarnpkg.com/lang/en/docs/install/).

```bash

cd /path/to/visdom

yarn # install node dependencies

yarn run build # build js

```

#### npm

You can find instructions for installing `npm` [here](https://github.com/npm/npm).

```bash

cd /path/to/visdom

npm install # install node dependencies

npm run build # build js

```

## Contributor License Agreement ("CLA")

In order to accept your pull request, we need you to submit a CLA. You only need

to do this once to work on any of Facebook's open source projects.

Complete your CLA here: <https://code.facebook.com/cla>

## Issues

We use GitHub issues to track public bugs. Please ensure your description is

clear and has sufficient instructions to be able to reproduce the issue.

Facebook has a [bounty program](https://www.facebook.com/whitehat/) for the safe

disclosure of security bugs. In those cases, please go through the process

outlined on that page and do not file a public issue.

## Coding Style

\* 3 spaces for indentation rather than tabs for Lua

\* Follow PEP 8 for Python

\* 80 character line length

## License

By contributing to Visdom, you agree that your contributions will be licensed

under the LICENSE file in the root directory of this source tree.