

Setting up a dev environment

Install Docker

We rely heavily on using docker containers to run the various services in a dev environment - make sure Docker is installed first before continuing.

Clone the repository & set up pre-commit

Next, let's grab the code and set up pre-commit, which will validate any commits for us before we attempt pull requests.

It's recommended to fork the `protohaven_api` repository rather than clone it directly (in the following command) so that you can make edits and follow the pull request process later on.

```
# Clone the repository
git clone git@github.com:protohaven/protohaven_api.git

# Set up pre-commit validation hooks
pip install pre-commit
pre-commit install
```

Load Data (NocoDB)

We use NocoDB as a local/dev database replacement for the production Airtable DB, and as a mock data source for other integrations such as Neon and Wyze. **Nocodb must be running in order to run other local dev services** like Flask and the svelte frontend, **and we must prefill it** with tables and dev data so that it behaves as expected locally.

You can load in data as follows:

```
./nocodb/init_or_update_nocodb.sh
```

Make sure to run this in the root of the repository (i.e. `protohaven_api/`) or else it will fail.

Follow the prompts in the script, and you should end up with a running instance of nocodb accessible at <http://localhost:9090>. You can login with

- Username: `admin@example.com`
- Password: `password`

Run all services

Starting all services is as simple as:

```
docker compose watch
```

Whenever you make a change, containers will build (if a rebuild is needed) and the relevant services will be reloaded.

Run the CLI

All services must be running in order to use `docker compose exec` in this way

```
# Test your setup - you should see a YAML formatted message reminding instructors to schedule classes
docker compose exec flask python3 -m protohaven_api.cli gen_instructor_schedule_reminder
```

Run Tests

To run all python tests: `python -m pytest -vv`, optionally with `-k test_substring` to run a specific test.

Logging in as different users

There's a helpful handler for setting your login session to one of the accounts in the "Fake Neon" base of your local Nocodb instance:

https://localhost:5173/admin/login_as?neon_id=1797

Replace `1797` with an account ID of your choice to populate your browser session with info from the matching row in [Fake Neon > accounts](#). 1791 in this case is set up as an instructor so can be useful for working on the class scheduling UI and automation.

Run Bookstack (<http://localhost:6875>)

First time setup

We run Bookstack in a Docker container with several volumes/files mounted - but Bookstack's first-run initialization process doesn't kick off if files are already present. To run Bookstack for the first time so this initialization happens:

```
cd protohaven_api/bookstack
docker compose --profile no_plugin up
```

Wait for initialization to settle (a couple of minutes), then proceed to adding custom head content below.

Adding custom head content

Manually copy the contents of `custom_head_content.html` (from `protohaven_api/bookstack/`) into the "Custom HTML Head Content" textarea in the [settings of the wiki](#).

Normal development

To start a server after the first run:

```
cd protohaven_api/bookstack
docker compose down
docker compose --profile protohaven up -d
```

To view debug logs:

```
tail -f protohaven_api/bookstack/bookstack_app_data/log/bookstack/laravel.log
```

"Invalid redirect_uri parameter value."

Because it's a local instance trying to do OAuth authentication with Neon CRM, you may see this login error when initially trying to view a page.

When the server is running locally, authentication can be shortcut using

http://localhost:6875/login_as?email=admin@admin.com. You may also create other users and login as them by changing the value of the `email` param.

Testing OAuth and role syncing

OAuth testing requires production credentials. This should be done on the Cron server in the `~/protohaven_api` instance of the repository.

With the a running instance of Bookstack on the Cron server:

1. Open a new tab, and open the browser console to the **Networking** tab so you can see network requests.
2. Browse to wiki.protohaven.org - logout if you're already logged in, using the triple-dot menu on the top right of the page. You should be immediately taken to a Neon login page.
3. Submit your login credentials and get redirected back to the prod wiki.
4. Search for `/login?code=<random string>`
5. Replace the base of the URL with the URL of the bookstack instance and browse to it to login to the local instance.

Run Wordpress

To start a wordpress dev container, run the following:

```
cd protohaven_api/wordpress
docker compose up
```

Note that the plugin directories are automatically installed via volume mounts in `docker-compose.yml`, however to get live updates you'll also need to run the nodejs script for the plugin you're currently editing:

```
cd protohaven_api/wordpress/protohaven-class-ticker
npm install
npm run start
```

Then you can browse to <http://localhost:8080> to interact with your wordpress server.

Revision #45

Created 18 January 2025 17:40:26 by Scott Martin

Updated 29 June 2025 14:33:53 by Bookstack Admin