Developing Discourse using a Dev Container

- Getting started
- Running tests
- Applying config/container updates
- References

Dev Containers is an open standard for configuring a development environment inside a container. This almost entirely eliminates the need to install/configure Discourse-specific tools/dependencies on your local machine, and makes it very easy to keep up-to-date as Discourse evolves over time.

Dev Containers can be used in a number of different IDEs, or directly using their reference CLI. This guide will describe the setup process for VSCode.

Getting started

1. Download and install VSCode
2. Install the Dev Containers extension in VSCode
3. Clone the Discourse repository onto your machine

   ```
   git clone https://github.com/discourse/discourse
   ```

4. In VSCode, use `File` -> `Open Folder`, then choose the Discourse directory
5. Open the folder in its Dev Container. This can be done via the popup prompt, or by opening the command palette (`Cmd/Ctrl + Shift + P`) and searching for "Open folder in container..."
6. If this is your first time launching a container, you will be prompted to install and start Docker Desktop. Once complete, go back to VSCode re-run "Open folder in container..."
7. Wait for the container to download and start. When it's done, the README will appear, and you'll see the Discourse filesystem in the sidebar.
8. Run the default build task using `Cmd/Ctrl + Shift + B`.

This will install dependencies, migrate the database, and start the server. It'll take a few minutes, especially on the lower-end machines. You'll see "Build successful" in the terminal when it's done.

9. Run the `dev/admin/create` task. Open the command palette and search for "Tasks: Run Tasks". It will present a menu of tasks; select `dev/admin/create` off of that list. You'll be prompted to enter an email address and a password for your admin user.
10. Visit `http://localhost:3000` in your browser to see your new Discourse instance
11. All done! You can now make changes to Discourse's source code and see them reflected in the preview.

Running tests

The first time you run tests, you'll need to install testing dependencies, including Playwright and the `discourse_test` DB. You can run the `deps/testing` task to install those. Open the command palette, select "Tasks: Run Tasks". It will present a menu of tasks; select `deps/testing` off of that list.

Once the test dependencies are installed, you can run `bin/lint`, `bin/qunit`, or the system specs

Applying config/container updates

Every so often, the devcontainer config and the associated container image will be updated. VSCode should prompt you to "rebuild" to apply the changes. Alternatively, you can run "Dev Containers: Rebuild Container" from the VSCode command palette. The working directory, and your Redis/Postgres data will be preserved across rebuilds.

If you'd like to start from scratch with fresh database, you'll need to delete the `discourse-pg` and `discourse-redis` docker volumes. This can be done from the "Remote Explorer" tab of the VSCode sidebar.

Discourse's sample vscode `.vscode/settings.json` and `.vscode/tasks.json` will be copied when you first boot the codespace. From that point forward, if you want to use the latest sample config, you'll need to manually copy `.vscode/settings.json.sample` to `.vscode/settings.json`.

References

- Development container specification
- VSCode devcontainer docs

□□□ #2
□ □□□ □□□ 2026-10-31 13:27:25 UTC
□ □□□ □□□ 2026-10-31 13:27:26 UTC