



[AutoGen](#)

S

[🏠 Contributing](#)

On this page

Contributing

This project welcomes and encourages all forms of contributions, including but not limited to:

- Pushing patches.
- Code review of pull requests.
- Documentation, examples and test cases.
- Readability improvement, e.g., improvement on docstr and comments.
- Community participation in [issues](#), [discussions](#), [discord](#), and [twitter](#).
- Tutorials, blog posts, talks that promote the project.
- Sharing application scenarios and/or related research.

Most contributions require you to agree to a Contributor License Agreement (CLA) declaring that you have the right to, and actually do, grant us the rights to use your contribution. For details, visit <https://cla.opensource.microsoft.com>.

If you are new to GitHub [here](#) is a detailed help source on getting involved with development on GitHub.

When you submit a pull request, a CLA bot will automatically determine whether you need to provide a CLA and decorate the PR appropriately (e.g., status check, comment). Simply follow the instructions provided by the bot. You will only need to do this once across all repos using our CLA.

This project has adopted the [Microsoft Open Source Code of Conduct](#). For more information see the [Code of Conduct FAQ](#) or contact opencode@microsoft.com with any additional questions or comments.

How to make a good bug report

When you submit an issue to [GitHub](#), please do your best to follow these guidelines! This will make it a lot easier to provide you with good feedback:

- The ideal bug report contains a short reproducible code snippet. This way anyone can try to reproduce the bug easily (see [this](#) for more details). If your snippet is longer than around 50 lines, please link to a [gist](#) or a GitHub repo.
- If an exception is raised, please **provide the full traceback**.
- Please include your **operating system type and version number**, as well as your **Python, autogen, scikit-learn versions**. The version of autogen can be found by running the following code snippet:

```
import autogen
print(autogen.__version__)
```

- Please ensure all **code snippets and error messages are formatted in appropriate code blocks**. See [Creating and highlighting code blocks](#) for more details.

Becoming a Reviewer

There is currently no formal reviewer solicitation process. Current reviewers identify reviewers from active contributors. If you are willing to become a reviewer, you are welcome to let us know on discord.

Guidance for Maintainers

General

- Be a member of the community and treat everyone as a member. Be inclusive.
- Help each other and encourage mutual help.
- Actively post and respond.
- Keep open communication.

Pull Requests

- For new PR, decide whether to close without review. If not, find the right reviewers. The default reviewer is microsoft/autogen. Ask users who can benefit from the PR to review it.
- For old PR, check the blocker: reviewer or PR creator. Try to unblock. Get additional help when needed.
- When requesting changes, make sure you can check back in time because it blocks merging.
- Make sure all the checks are passed.
- For changes that require running OpenAI tests, make sure the OpenAI tests pass too. Running these tests requires approval.
- In general, suggest small PRs instead of a giant PR.
- For documentation change, request snapshot of the compiled website, or compile by yourself to verify the format.
- For new contributors who have not signed the contributing agreement, remind them to sign before reviewing.
- For multiple PRs which may have conflict, coordinate them to figure out the right order.
- Pay special attention to:
 - Breaking changes. Don't make breaking changes unless necessary. Don't merge to main until enough heads up is provided and a new release is ready.
 - Test coverage decrease.
 - Changes that may cause performance degradation. Do regression test when test suites are available.
 - Discourage **change to the core library** when there is an alternative.

Issues and Discussions

- For new issues, write a reply, apply a label if relevant. Ask on discord when necessary. For roadmap issues, add to the roadmap project and encourage community discussion. Mention relevant experts when necessary.
- For old issues, provide an update or close. Ask on discord when necessary. Encourage PR creation when relevant.
- Use “good first issue” for easy fix suitable for first-time contributors.
- Use “task list” for issues that require multiple PRs.
- For discussions, create an issue when relevant. Discuss on discord when appropriate.

Docker for Development

For developers contributing to the AutoGen project, we offer a specialized Docker environment. This setup is designed to streamline the development process, ensuring that all contributors work within a consistent and well-equipped environment.

Autogen Developer Image (autogen_dev_img)

- **Purpose:** The `autogen_dev_img` is tailored for contributors to the AutoGen project. It includes a suite of tools and configurations that aid in the development and testing of new features or fixes.
- **Usage:** This image is recommended for developers who intend to contribute code or documentation to AutoGen.
- **Forking the Project:** It's advisable to fork the AutoGen GitHub project to your own repository. This allows you to make changes in a separate environment without affecting the main project.
- **Updating Dockerfile.dev:** Modify your copy of `Dockerfile.dev` as needed for your development work.
- **Submitting Pull Requests:** Once your changes are ready, submit a pull request from your branch to the upstream AutoGen GitHub project for review and integration. For more details on contributing, see the [AutoGen Contributing](#) page.

Building the Developer Docker Image

- To build the developer Docker image (`autogen_dev_img`), use the following commands:

```
docker build -f samples/docker/Dockerfile.dev -t autogen_dev_img https://github.com/microsoft/autogen.git
```

- For building the developer image built from a specific Dockerfile in a branch other than main/master

```
# clone the branch you want to work out of
git clone --branch {branch-name} https://github.com/microsoft/autogen.git

# cd to your new directory
cd autogen

# build your Docker image
docker build -f samples/docker/Dockerfile.dev -t autogen_dev-srv_img .
```

Using the Developer Docker Image

Once you have built the `autogen_dev_img`, you can run it using the standard Docker commands. This will place you inside the containerized development environment where you can run tests, develop code, and ensure everything is functioning as expected before submitting your contributions.

```
docker run -it -p 8081:3000 -v `pwd`/autogen-newcode:newstuff/ autogen_dev_img bash
```

- Note that the `pwd` is shorthand for present working directory. Thus, any path after the `pwd` is relative to that. If you want a more verbose method you could remove the "`pwd/autogen-newcode`" and replace it with the full path to your directory

```
docker run -it -p 8081:3000 -v /home/AutoGenDeveloper/autogen-newcode:newstuff/ autogen_dev_img bash
```

Develop in Remote Container

If you use vscode, you can open the autogen folder in a [Container](#). We have provided the configuration in [devcontainer](#). They can be used in GitHub codespace too. Developing AutoGen in dev containers is recommended.

Pre-commit

Run `pre-commit install` to install pre-commit into your git hooks. Before you commit, run `pre-commit run` to check if you meet the pre-commit requirements. If you use Windows (without WSL) and can't commit after installing pre-commit, you can run `pre-commit uninstall` to uninstall the hook. In WSL or Linux this is supposed to work.

Write tests

Tests are automatically run via GitHub actions. There are two workflows:

1. [build.yml](#)
2. [openai.yml](#)

The first workflow is required to pass for all PRs (and it doesn't do any OpenAI calls). The second workflow is required for changes that affect the OpenAI tests (and does actually call LLM). The second workflow requires approval to run. When writing tests that require OpenAI calls, please use `pytest.mark.skipif` to make them run in only when `openai` package is installed. If additional dependency for this test is required, install the dependency in the corresponding python version in [openai.yml](#).

Run non-OpenAI tests

To run the subset of the tests not depending on `openai` (and not calling LLMs)):

- Install pytest:

```
pip install pytest
```

- Run the tests from the `test` folder using the `--skip-openai` flag.

```
pytest test --skip-openai
```

- Make sure all tests pass, this is required for [build.yml](#) checks to pass

Coverage

Any code you commit should not decrease coverage. To run all unit tests, install the `[test]` option:

```
pip install -e."[test]"
coverage run -m pytest test
```

Then you can see the coverage report by `coverage report -m` or `coverage html`.

Documentation

To build and test documentation locally, install [Node.js](#). For example,

```
nvm install --lts
```

Then:

```
npm install --global yarn # skip if you use the dev container we provided
pip install pydoc-markdown # skip if you use the dev container we provided
cd website
yarn install --frozen-lockfile --ignore-engines
pydoc-markdown
yarn start
```

The last command starts a local development server and opens up a browser window. Most changes are reflected live without having to restart the server.

To build and test documentation within a docker container. Use the Dockerfile.dev as described above to build your image

```
docker build -f samples/docker/Dockerfile.dev -t autogen_dev_img https://github.com/microsoft/autogen.git
```

Then start the container like so, this will log you in and ensure that Docker port 3000 is mapped to port 8081 on your local machine

```
docker run -it -p 8081:3000 -v `pwd`/autogen-newcode:newstuff/ autogen_dev_img bash
```

Once at the CLI in Docker run the following commands:

```
cd website
yarn install --frozen-lockfile --ignore-engines
pydoc-markdown
yarn start --host 0.0.0.0 --port 3000
```

Once done you should be able to access the documentation at `http://127.0.0.1:8081/autogen`

Note: some tips in this guide are based off the contributor guide from [flaml](#).

 [Edit this page](#)

[Previous](#)

[« Enhanced Inference](#)

[Next](#)
[Research »](#)

Community

[Discord](#) 

[Twitter](#) 

Copyright © 2024 AutoGen Authors | [Privacy and Cookies](#)