# Contributing

This project welcomes any type of contribution! ❤️ Opening an issue to document a problem you encountered or suggesting a new feature is always a good start.

Before you submit a pull request, please discuss potential changes with the maintainer either in an issue, using GitHub Discussions or via email.

# Development

## Get Started

To get started with development, follow these four steps:

1. Clone the repo and run `yarn` to install all dependencies.
2. Open `workspace.code-workspace` with Visual Studio Code.
3. Run the "extension: Build and Watch" task, which will continuously (re-)build the extension.
4. Run the "Testbench: NodeJS" launch configuration to open a new Visual Studio Code window, which:
    - loads the RxJS debugging extension in development mode, so you can use the debugger in the original Visual Studio Code window.
    - uses `packages/testbench-nodejs` as workspace, so you can test the RxJS debugging extension with a real example.

## Repository Structure

This repository is organized as monorepo. We use nx and lerna to streamline tasks.

Following packages can be found in the `packages` directory:

- `extension`: The main package containing the debugging extension for Visual Studio Code.
- `telemetry`: TypeScript types and helper functions used for communication between runtime and debugging extension.
- `runtime`: Contains rudimentary utilities to augment RxJS in an arbitrary runtime environment.
- `runtime-nodejs`: NodeJS specific augmentation functionalities.
- `runtime-webpack`: Webpack plugin, published as `@rxjs-debugging/runtime-webpack`, providing runtime augmentation for web applications built with Webpack.
- `extension-integrationtest`: An integration test suite verifying various aspects of

the extension.
- `testbench-*`: Test environments simulating various scenarios to test the debugger.

## Run Test Suites

Unit and integration tests are automatically executed once changes are pushed to Github. You can run them locally using the following commands:

- Unit tests:

```
yarn nx run-many --target=test --all --parallel
```

- Integration tests:

```
yarn nx run extension-integrationtest:integrationtest --configuration=test
```

## Architecture Concepts

The ARCHITECTURE.md file gives an overview on the most important architectural concepts.