# Contributing to Capacitor

This guide provides instructions for contributing to Capacitor through [issues & discussions](#issues--discussions) and [code](#developing-capacitor).

## Issues & Discussions

The Capacitor repo uses GitHub [issues](https://github.com/ionic-team/capacitor/issues) and [discussions](https://github.com/ionic-team/capacitor/discussions) to track bugs and feature requests, as well as to provide a place for community questions, ideas, and discussions.

\* \*\*When to use [issues](https://github.com/ionic-team/capacitor/issues)\*\*:

\* To report specific, reproducible bugs (see [Creating a Code Reproduction](#creating-a-code-reproduction)).

\* To propose detailed feature requests.

\* \*\*When to use [discussions](https://github.com/ionic-team/capacitor/discussions)\*\*:

\* To ask for help.

\* To ask general questions.

\* To show off cool stuff.

\* To propose ideas for improvement.

\* If you think you found a bug, but may need help to further uncover it.

\* Anything else! :rainbow:

### Creating a Code Reproduction

When reporting bugs, we ask you to provide a minimal sample application that demonstrates the issue. Without a reliable code reproduction, it is unlikely we will be able to resolve the issue, leading to it being closed.

To create a code reproduction:

\* Create a new application using `npx @capacitor/cli create` (or `ionic start --capacitor`).

\* Add the minimum amount of code necessary to recreate the issue you're experiencing.

\* Push the code reproduction to a public GitHub repository and include a link when you create a bug report.

\* Be sure to include steps to reproduce the issue.

## Developing Capacitor

### Repositories

\* [Capacitor](https://github.com/ionic-team/capacitor) (this repo): Core Capacitor platforms, CLI, and APIs

\* [Capacitor Plugins](https://github.com/ionic-team/capacitor-plugins): Official Capacitor plugins

\* [Capacitor Community](https://github.com/capacitor-community/): GitHub org for Capacitor Community plugins and platforms

\* [Capacitor Site](https://github.com/ionic-team/capacitor-site): Capacitor website and online documentation

\* [Capacitor TestApp](https://github.com/ionic-team/capacitor-testapp): Test app used by the core team for developing Capacitor

### Design Philosophy

Before working on Capacitor, it's important to understand the philosophy behind the project to avoid investing time in things that won't fit into the goals of the project.

Please read [@maxlynch](http://twitter.com/maxlynch)'s essay [How Capacitor Works](https://tinyletter.com/ionic-max/letters/how-capacitor-works) for a deep dive into the project and its goals.

### Consult with the team

For any large changes, make sure you've consulted with the team first. You can [open a discussion](https://github.com/ionic-team/capacitor/discussions) to bring up your idea.

### About Third Party Libraries

To achieve Capacitor's goal of being stable and easy to upgrade, we would like to avoid unnecessary third party libraries as much as possible. Before embarking on Capacitor contributions, make sure you aren't planning on introducing third party libraries without consulting with the team first.

On native, that means avoid adding any new Cocoapod or Gradle dependencies without explicit approval. If you just need a small bit of functionality from that library, consider adding an implementation to the codebase directly.

On web, this means do not add any third party libraries such as Firebase or Lodash. Strive for implementations that use pure Web APIs even if it means more work.

### Local Setup

1. Fork and clone the repo.

1. Install the monorepo dependencies.

```shell

npm install

```

1. Install SwiftLint if you're on macOS. Contributions to iOS code will be linted in CI if you don't have macOS.

```shell

brew install swiftlint

```

1. Install package dependencies. [Lerna](https://github.com/lerna/lerna) can automatically install each package's dependencies.

```shell

npx lerna bootstrap

```

### Branches

\* [`main`](https://github.com/ionic-team/capacitor/tree/main): Latest Capacitor development branch

\* [`2.x`](https://github.com/ionic-team/capacitor/tree/2.x): Capacitor 2 (bug and security fixes only)

\* [`1.x`](https://github.com/ionic-team/capacitor/tree/1.x): Capacitor 1 (not maintained)

### Directory Structure

This monorepo contains core Capacitor components. The current directory structure looks like this:

\* `app-template`: The default app template used during `create`

\* `cli`: Capacitor CLI/Build scripts

\* `core`: Capacitor Core JS library

\* `ios`: Capacitor iOS Runtime

\* `ios-template`: Default iOS App installed by the CLI

\* `android`: Capacitor Android Runtime

\* `android-template`: Default Android App installed by the CLI

\* `plugin-template`: The default plugin template when creating a new plugin

## Publishing Capacitor

Capacitor packages are published using [Lerna](https://github.com/lerna/lerna) with fixed versioning.

During Capacitor 3 development, the following workflow is used to create dev releases:

1. Create the next development version. The following command will:

\* Create a release commit with a generated changelog

\* Create a git tag

\* Push to the `main` branch

\* Create a GitHub release

<br>

```

npx lerna version prerelease

```

1. Wait for CI to publish the new tagged version.