

这篇指南会指导你如何为 Ant Design 贡献一份自己的力量，请在你要提 issue 或者 pull request 之前花几分钟来阅读一遍这篇指南。

行为准则

我们有一份 [行为准则](#)，希望所有的贡献者都能遵守，请花时间阅读一遍全文以确保你能明白哪些是可以做的，哪些是不可以做的。

透明的开发

我们所有的工作都会放在 [GitHub](#) 上。不管是核心团队的成员还是外部贡献者的 pull request 都需要经过同样流程的 review。

分支管理

基于我们的 [发布周期](#)，我们长期维护两个分支 `master` 和 `feature`。如果你要修一个 bug，那么请发 pull request 到 `master`，我们会每周从 master 发布一个 patch 版本；如果你要提一个增加新功能的 pull request，那么请基于 `feature` 分支来做，每月末我们会合并 feature 到 master，并发布一个包含新特性的 minor 版本。

Bugs

我们使用 [GitHub Issues](#) 来做 bug 追踪。如果你想要你发现的 bug 被快速解决，最好的办法就是通过我们提供的 [issue 小助手](#) 来提 issue。并且能使用这个 [模板](#) 来提供重现。

在你报告一个 bug 之前，请先确保已经搜索过已有的 issue 和阅读了我们的 [常见问题](#)。

新增功能

如果你有改进我们的 API 或者新增功能的想法，我们同样推荐你使用我们提供的 [issue 小助手](#) 来新建一个添加新功能的 issue。

如果你希望协助开发新的 API，请参考 [API 规范](#) 进行命名。

第一次贡献

如果你还不清楚怎么在 GitHub 上提 Pull Request，可以阅读下面这篇文章来学习：

[如何优雅地在 GitHub 上贡献代码](#)

为了能帮助你开始你的第一次尝试，我们用 [good first issues](#) 标记了一些比较容易修复的 bug 和小功能。这些 issue 可以很好地作为你的首次尝试。

如果你打算开始处理一个 issue，请先检查一下 issue 下面的留言以确保没有别人正在处理这个 issue。如果当前没有人在处理的话你可以留言告知其他人你将会处理这个 issue，以免别人重复劳动。

如果之前有人留言说会处理这个 issue 但是一两个星期都没有动静，那么你也可以接手处理这个 issue，当然还是需要留言告知其他人。

Pull Request

Ant Design 团队会关注所有的 pull request，我们会 review 以及合并你的代码，也有可能要求你做一些修改或者告诉你我们为什么不能接受这样的修改。

在你发送 Pull Request 之前，请确认你是按照下面的步骤来做的：

1. 基于 [正确的分支](#) 做修改。
2. 在项目根目录下运行了 `npm install`。

在 Windows 10 开发环境下，如果出现 `gyp err! find vs msvs_version not set from command line or npm config` 错误，请在运行 `npm install` 前安装 [最新版 Python v3](#)，并通过 [Visual Studio Installer](#) 安装 **Desktop development with C++**。

3. 如果你修复了一个 bug 或者新增了一个功能，请确保写了相应的测试，这很重要。
4. 确认所有的测试都是通过的 `npm run test`。小贴士：开发过程中可以用 `npm test -- --watch TestName` 来运行指定的测试。
5. 运行 `npm test -- -u` 来更新 [jest snapshot](#) 并且把这些更新也提交上来（如果有的话）。
6. 确认所有的 UI 改动通过 `npm run test-image`，可以运行 `npm run test-image -- -u` 更新 UI 快照并且把这些更新也提交上来（如果有的话），**UI 测试基于 Docker**，根据平台下载对应的安装程序。
7. 确保你的代码通过了 lint 检查 `npm run lint`。小贴士：Lint 会在你 `git commit` 的时候自动运行（通过 [Git Hooks](#)）。

给 [react-component](#) 发送 pull request：

由于 antd 的大部分组件都是基于 react-component 的，所以有时候你可能需要给相应的 react-component 仓库发送 pull request。如果你是修复了某个 bug，那么我们在合并你的修改后会尽快发布一个 patch 版本，然后你只要重新安装你的依赖就可以使用新发布的版本了。如果你的 pull request 是新增了某个功能，那么在你的修改合并并且发布版本后，你还需要发送一个 pull request 到 [Ant Design](#) 来升级相应的依赖、文档以及 TypeScript 的类型定义。

开发流程

在你 clone 了 antd 的代码并且使用 `npm install` 安装完依赖后，你还可以运行下面几个常用的命令：

1. `npm start` 在本地运行 Ant Design 的网站。
2. `npm run lint` 检查代码风格。
3. `npm test` 运行测试。（在运行测试前请确保 `NODE_ENV` 环境变量没有被设定，否则可能会引发一些问题）
4. `npm run compile` 编译 TypeScript 代码到 lib 和 es 目录。
5. `npm run dist` 构建 antd 的 UMD 版本到 dist 目录。

切换主题

在启动时，设置需要执行的主题：

```
DEV_THEME=dark npm start
```

访问 <http://127.0.0.1:8001/components/button-cn/?theme=dark>。

加入社区

如果你贡献度足够活跃，希望和 Ant Design 团队一起参与维护工作，你可以[申请成为社区协作者](#)。

你还可以参考下面两篇社区成员写的贡献指南，一步一步成为 antd 的贡献者吧：

- [记录向：如何快速的成为 Ant Design 的 contributor @Rustin-Liu](#)
- [从 0 开始，成为 Ant-Design Contributor @fireairforce](#)