
link: null
title: 珠峰架构师成长计划
description: babel.config.js
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=137 sentences=415, words=2351

0.AntDesign

- 框架 react
- 测试 jest+enzyme
- 检查 eslint
- 打包 webpack+gulp
- 文档 bisheng
- 钩子 husky
- .husky git钩子
- _site 网站
- components 组件
- docs 文档
- dist 打包生成的文件
- es ES6
- lib ES5
- scripts 脚本
- site 组件预览项目
- tests 测试
- typings 类型定义
- webpack配置
- storybook文档和组件编写
- 单元测试+E2E快照测试+代码覆盖率
- eslint+prettier+editorconfig
- git hook
- 编译发布
- 持续集成

1.创建项目

```
mkdir ant  
cd ant  
npm init -y
```

```

{
  "name": "@zhangry/ant",
  "version": "1.0.0",
  "description": "React组件的企业级UI设计",
  "main": "lib/index.js",
  "scripts": {
    "build": "webpack"
  },
  "publishConfig": {
    "access": "public",
    "registry": "http://registry.npmjs.org"
  },
  "homepage": "https://zhangrenyang.github.io/ant",
  "repository": {
    "type": "git",
    "url": "git+https://github.com/zhangrenyang/ant.git"
  },
  "keywords": [
    "ant",
    "component",
    "components",
    "design",
    "framework",
    "frontend",
    "react",
    "react-component",
    "ui"
  ],
  "author": "zhangrenyang",
  "license": "MIT",
  "dependencies": {
    "react": "^17.0.2",
    "react-dom": "^17.0.2"
  },
  "peerDependencies": {
    "react": "^17.0.2",
    "react-dom": "^17.0.2"
  },
  "devDependencies": {
    "@typescript-eslint/parser": "^4.31.1",
    "and": "^0.0.3",
    "eslint": "^7.32.0",
    "eslint-config-airbnb": "^18.2.1",
    "eslint-plugin-import": "^2.24.2",
    "eslint-plugin-jsx-ally": "^6.4.1",
    "eslint-plugin-react": "^7.25.2",
    "eslint-plugin-react-hooks": "^4.2.0",
    "husky": "^7.0.2",
    "@babel/core": "^7.15.5",
    "@babel/plugin-transform-runtime": "^7.15.0",
    "@babel/plugin-transform-typescript": "^7.15.4",
    "@babel/preset-env": "^7.15.6",
    "@babel/preset-react": "^7.14.5",
    "@commitlint/cli": "^13.1.0",
    "@commitlint/config-conventional": "^13.1.0",
    "@storybook/addon-essentials": "^6.3.8",
    "@storybook/react": "^6.3.8",
    "@types/enzyme": "^3.10.9",
    "@types/jest": "^27.0.1",
    "@types/jest-environment-puppeteer": "^4.4.1",
    "@types/jest-image-snapshot": "^4.3.1",
    "@types/node": "^16.9.2",
    "@types/puppeteer": "^5.4.4",
    "@types/react": "^17.0.21",
    "@types/react-dom": "^17.0.9",
    "@wojtekmaj/enzyme-adapter-react-17": "^0.6.3",
    "autoprefixer": "^10.3.4",
    "babel-loader": "^8.2.2",
    "commitizen": "^4.2.4",
    "css-loader": "^6.2.0",
    "cz-customizable": "^6.3.0",
    "enzyme": "^3.11.0",
    "eslint-config-prettier": "^8.3.0",
    "eslint-plugin-prettier": "^4.0.0",
    "gulp": "^4.0.2",
    "gulp-babel": "^8.0.0",
    "gulp-typescript": "^6.0.0-alpha.1",
    "jest": "^27.2.0",
    "jest-environment-puppeteer": "^5.0.4",
    "jest-image-snapshot": "^4.5.1",
    "jest-puppeteer": "^5.0.4",
    "less": "^4.1.1",
    "less-loader": "^10.0.1",
    "merge2": "^1.4.1",
    "mini-css-extract-plugin": "^2.3.0",
    "postcss-loader": "^6.1.1",
    "prettier": "^2.4.1",
    "puppeteer": "^10.2.0",
    "rimraf": "^3.0.2",
    "typescript": "^4.4.3",
    "webpack": "^5.53.0",
    "webpack-cli": "^4.8.0"
  }
}

```

2.配置webpack

```

yarn add webpack webpack-cli webpack-dev-server mini-css-extract-plugin babel-loader css-loader autoprefixer postcss-loader less-loader less @babel/core
@babel/preset-react @babel/preset-env @babel/runtime @babel/plugin-transform-typescript typescript @babel/plugin-transform-runtime @types/node --dev

yarn add react react-dom
yarn add @types/react @types/react-dom --dev

```

```

const path = require('path');
const MiniCssExtractPlugin = require('mini-css-extract-plugin');

const cwd = process.cwd();
module.exports = {
  mode: 'development',
  devtool: false,
  entry: {
    ant: './index.js',
  },
  output: {
    path: path.resolve('dist'),
    filename: '[name].js',
    library: 'ant',
    libraryTarget: 'umd',
  },
  externals: {
    react: {
      root: 'React',
      commonjs2: 'react',
      commonjs: 'react',
      amd: 'react',
    },
    'react-dom': {
      root: 'ReactDOM',
      commonjs2: 'react-dom',
      commonjs: 'react-dom',
      amd: 'react-dom',
    },
  },
  resolve: {
    extensions: ['.ts', '.tsx', '.js', '.jsx', '.json'],
    alias: {
      antdesign: cwd,
    },
  },
  module: {
    rules: [
      {
        test: /\.?(j|t)sx?$/,
        exclude: /node_modules/,
        loader: 'babel-loader',
      },
      {
        test: /\.css$/,
        use: [
          MiniCssExtractPlugin.loader,
          {
            loader: 'css-loader',
            options: {
              sourceMap: true,
            },
          },
          {
            loader: 'postcss-loader',
            options: {
              postcssOptions: {
                plugins: ['autoprefixer'],
              },
              sourceMap: true,
            },
          },
        ],
      },
      {
        test: /\.less$/,
        use: [
          MiniCssExtractPlugin.loader,
          {
            loader: 'css-loader',
            options: {
              sourceMap: true,
            },
          },
          {
            loader: 'postcss-loader',
            options: {
              postcssOptions: {
                plugins: ['autoprefixer'],
              },
              sourceMap: true,
            },
          },
          {
            loader: 'less-loader',
            options: {
              lessOptions: {
                javascriptEnabled: true,
              },
              sourceMap: true,
            },
          },
        ],
      },
      {
        test: /\.(png|jpg|jpeg|gif|svg) (\?v=\d+\.\d+\.\d+)?$/i,
        type: 'asset',
      },
    ],
  },
  plugins: [
    new MiniCssExtractPlugin({
      filename: '[name].css',
    })
  ]
}

```

```
    }},  
  ],  
};
```

babel.config.js

```
module.exports = {  
  presets: [  
    '@babel/preset-react',  
    [  
      '@babel/preset-env',  
      {  
        modules: 'auto',  
        targets: {  
          browsers: ['last 2 versions', 'Firefox ESR', '> 1%', 'ie >= 11'],  
        },  
      },  
    ],  
  ],  
  plugins: [  
    [  
      '@babel/plugin-transform-typescript',  
      {  
        isTSX: true,  
      },  
    ],  
    '@babel/plugin-transform-runtime',  
  ],  
};
```

.gitignore

```
*.iml
.idea/
.ipr
.iws
*~
~*
*.diff
*.patch
*.bak
.DS_Store
Thumbs.db
.project
.*proj
.svn/
*.swp
*.swo
*.log
*.log.*
*.json.gzip
node_modules/
.buildpath
.settings
npm-debug.log
nohup.out
_site
_data
diist
report.html
/lib
/es
elasticsearch-*
config/base.yaml
/.vscode/
/coverage
yarn.lock
package-lock.json
components/**/*.js
components/**/*.jsx
!components/**/*.__tests__/**/*.js
!components/**/*.__tests__/**/*.js.snap
/.history
*.tmp

# Docs templates
site/theme/template/Color/ColorPicker.jsx
site/theme/template/IconDisplay/*.js
site/theme/template/IconDisplay/*.jsx
site/theme/template/IconDisplay/fields.js
site/theme/template/Home/**/*.jsx
site/theme/template/utils.jsx
site/theme/template/Layout/Footer.jsx
site/theme/template/Layout/Header/**/*.jsx
site/theme/template/Layout/SiteContext.jsx
site/theme/template/Content/Article.jsx
site/theme/template/Content/EditButton.jsx
site/theme/template/Resources/*.jsx
site/theme/template/Resources/**/*.jsx
site/theme/template/NotFound.jsx
scripts/previewEditor/index.html
components/version/version.tsx

# Image snapshot diff
__diff_output__/_
__image_snapshots__/_
/jest-stare
/imageSnapshots
/imageDiffSnapshots
storybook-static

sh.exe.stackdump
/snapshots
/diffSnapshots
```

```
{
  "compilerOptions": {
    "strictNullChecks": true,
    "module": "esnext",
    "moduleResolution": "node",
    "esModuleInterop": true,
    "experimentalDecorators": true,
    "jsx": "react",
    "noUnusedParameters": true,
    "noUnusedLocals": true,
    "noImplicitAny": true,
    "target": "es6",
    "lib": ["dom", "es2017"],
    "skipLibCheck": true,
    "types": ["node"]
  },
  "exclude": ["node_modules", "lib", "es"]
}
```

```
module.exports = require('./components');
```

```
components/index.tsx
```

```
import Button from './button';

export type { ButtonProps } from './button';
export { Button };
```

```
components/button/index.tsx
```

```
import Button from './button';
export default Button;
export type { ButtonProps } from './button';
```

components/button/button.tsx

```
import React, { ButtonHTMLAttributes } from 'react';

interface ButtonProps extends ButtonHTMLAttributes {}
const Button: React.FC = (props) => {
  const { children } = props;
  return <button type="button">{children}</button>;
};

export default Button;
export type { ButtonProps };
```

3.storybook

```
yarn add @storybook/react @storybook/addon-essentials --dev
```

.storybook/main.js

```
module.exports = {
  stories: [
    '../components/Introduction.stories.mdx',
    '../components/Install.stories.mdx',
    '../components/Components.stories.mdx',
    '../components/**/*.stories.mdx',
    '../components/**/*.stories.@(js|jsx|ts|tsx)'
  ],
  addons: ['@storybook/addon-essentials'],
};
```

components/Introduction.stories.mdx

```
## Ant Design of React
antd 是基于 Ant Design 设计体系的 React UI 组件库，主要用于研发企业级中后台产品。
```

components/Install.stories.mdx

```
## 安装
使用 npm 或 yarn 安装

npm install ant --save

yarn add ant

## 浏览器引入
在浏览器中使用 script 和 link 标签直接引入文件，并使用全局变量 ant
我们在 npm 发布包内的 antdesign/dist 目录下提供了 ant.js

## 示例

import { Button } from 'antdesign';
ReactDOM.render(按钮, mountNode);
```

components/Components.stories.mdx

组件总览
antd 是基于 Ant Design 设计体系的 React UI 组件库，主要用于研发企业级中后台产品。

通用

- Button 按钮
- Icon 图标
- Typography 排版

布局

- Divider 分割线
- Grid 栅格
- Layout 布局
- Space 间距

导航

- Affix 固钉
- Breadcrumb 面包屑
- Dropdown 下拉菜单
- Menu 导航菜单
- Pagination 分页
- PageHeader 页头
- Steps 步骤条

数据录入

- AutoComplete 自动完成
- Checkbox 多选框
- Cascader 级联选择
- DatePicker 日期选择框
- Form 表单
- InputNumber 数字输入框
- Input 输入框
- Mentions 提及
- Rate 评分
- Radio 单选框
- Switch 开关
- Slider 滑动输入条
- Select 选择器
- TreeSelect 树选择
- Transfer 穿梭框
- TimePicker 时间选择框
- Upload 上传

数据展示

- Avatar 头像
- Badge 徽标数
- Comment 评论
- Collapse 折叠面板
- Carousel 走马灯
- Card 卡片
- Calendar 日历
- Descriptions 描述列表
- Empty 空状态
- Image 图片
- List 列表
- Popover 气泡卡片
- Statistic 统计数值
- Tree 树形控件
- Tooltip 文字提示
- Timeline 时间轴
- Tag 标签
- Tabs 标签页
- Table 表格

反馈

- Alert 警告提示
- Drawer 抽屉
- Modal 对话框
- Message 全局提示
- Notification 通知提醒框
- Progress 进度条
- Popconfirm 气泡确认框
- Result 结果
- Spin 加载中
- Skeleton 骨架屏

其他

- Anchor 锚点
- BackTop 回到顶部
- ConfigProvider 全局化配置

components/button/button.stories.tsx

```
import React from "react";
import { ComponentStory, ComponentMeta } from "@storybook/react";

import Button from ".";

export default {
  title: "通用/Button(按钮)",
  component: Button,
} as ComponentMeta<typeof Button>;

const Template: ComponentStory<typeof Button> = (args) => (
  <Button {...args} />
);

export const Basic = Template.bind({});
Basic.args = {
  children: "按钮",
};
```

```
"scripts": {
  "build": "webpack",
+  "storybook": "start-storybook -p 6006",
+  "build-storybook": "build-storybook"
},
```

4.测试

```
yarn add jest @types/jest @wojtekmaj/enzyme-adapter-react-17 puppeteer @types/puppeteer jest-environment-puppeteer @types/jest-environment-puppeteer jest-
puppeteer jest-image-snapshot @types/jest-image-snapshot --dev
yarn add enzyme @types/enzyme --dev
```

tests\setup.js

```
const React = require('react');
const Enzyme = require('enzyme');

const Adapter = require('@wojtekmaj/enzyme-adapter-react-17')
Enzyme.configure({ adapter: new Adapter() });
```

tests\index.html

```
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Amazing Antdtitle</title>
  <style>
    body {
      border: 5px solid #1890ff;
    }
  </style>
</head>
<body>
  <div id="root">div</div>
</body>
</html>
```

unit\jest.js

```
module.exports = {
  verbose: true,
  testEnvironment: 'jsdom',
  setupFiles: ['./tests/setup.js'],
  testMatch: ['**/unit/**/*.spec.test).(js|ts|jsx|tsx)'],
  collectCoverage: true,
  collectCoverageFrom: [
    'components/**/*.js|ts|jsx|tsx',
    '!components/**/*.stories.js|ts|jsx|tsx',
    '!components/**/*.spec.test).(js|ts|jsx|tsx)',
  ],
};
```

e2e\jest.js

```
module.exports = {
  verbose: true,
  testEnvironment: 'jest-environment-puppeteer',
  setupFiles: ['./tests/setup.js'],
  preset: 'jest-puppeteer',
  testMatch: ['**/e2e/**/*.spec.test).(j|t)sx'],
};
```

components\button\unit\index.test.tsx

```
import React from 'react';
import { mount } from 'enzyme';
import Button from '..';

describe('Button', () => {
  it('mount correctly', () => {
    expect(() => mount(<Button>FollowButton</Button>)).not.toThrow();
  });
});
```

components\button\le2e\snapshot.spec.tsx

```
import React from 'react';
import ReactDOMServer from 'react-dom/server';
import { configureToMatchImageSnapshot } from 'jest-image-snapshot';
import Button from '..';
import 'jest-environment-puppeteer';
const toMatchSnapshot = configureToMatchImageSnapshot({
  customSnapshotsDir: `${process.cwd()}/snapshots`,
  customDiffDir: `${process.cwd()}/diffSnapshots`,
});
expect.extend({ toMatchSnapshot });
describe('Button snapshot', () => {
  it('screenshot should correct', async () => {
    await jestPuppeteer.resetPage();
    await page.goto(`file://${process.cwd()}/tests/index.html`);
    const html = ReactDOMServer.renderToString(<Button>按钮Button</Button>);
    await page.evaluate((innerHTML:string) => {
      document.querySelector('#root')!.innerHTML = innerHTML;
    }, html);
    const screenshot = await page.screenshot();
    expect(screenshot).toMatchSnapshot();
  });
});
```

jest-puppeteer.config.js


```
module.exports = {
  launch: {
    dumpio: true,
    headless: process.env.HEADLESS !== 'false',
  },
  browserContext: 'default',
};
```

package.json

```
{
  "scripts": {
    "build": "webpack",
    "storybook": "start-storybook -p 6006",
    "build-storybook": "build-storybook",
+   "test:unit": "jest --config unit.jest.js",
+   "test:e2e": "jest --config e2e.jest.js",
+   "test": "npm run test:unit && npm run test:e2e"
  },
}
```

5.eslint

```
yarn add @typescript-eslint/parser eslint eslint-plugin-import eslint-plugin-react eslint-plugin-react-hooks and eslint-plugin-jsx-ally eslint-config-airbnb --dev
```

.eslintrc.js

```
module.exports = {
  parser: '@typescript-eslint/parser',
  extends: ['airbnb'],
  env: {
    browser: true,
    node: true,
    jasmine: true,
    jest: true,
    es6: true,
  },
  rules: {
    'import/extensions': 0,
    'import/no-unresolved': 0,
    'react/jsx-filename-extension': 0,

    'no-use-before-define': 0,
    'import/prefer-default-export': 0,
    'import/no-named-default': 0,
    'no-console': 0,
    'no-param-reassign': 0,
    'func-names': 0,
  }
};
```

.eslintignore

```
componentsunit*.stories.*
lib
es
umd
dist
.storybook
gulpfile.js
```

package.json

```
{
  "scripts": {
    "build": "webpack",
    "storybook": "start-storybook -p 6006",
    "build-storybook": "build-storybook",
    "test:unit": "jest --config unit.jest.js",
    "test:e2e": "jest --config e2e.jest.js",
    "test": "npm run test:unit && npm run test:e2e",
+   "lint": "eslint --ext .js,.jsx,.ts,.tsx components",
+   "lint:fix": "eslint --fix --ext .js,.jsx,.ts,.tsx components"
  }
}
```

6.prettier

```
yarn add prettier eslint-config-prettier eslint-plugin-prettier --dev
```

.eslintrc.js

```

module.exports = {
  parser: '@typescript-eslint/parser',
+ extends: ['airbnb','prettier'],
  env: {
    browser: true,
    node: true,
    jasmine: true,
    jest: true,
    es6: true,
  },
+ plugins: ['prettier'],
  rules: {
+   'prettier/prettier': ['error', { endOfLine: 'auto' }],
    'import/extensions': 0,
    'import/no-unresolved': 0,
    'react/jsx-filename-extension': 0,
    // https://github.com/typescript-eslint/typescript-eslint/issues/2540#issuecomment-692866111
    'no-use-before-define': 0,
    'import/prefer-default-export': 0,
    'import/no-named-default': 0,
    'no-console': 0,
    'no-param-reassign': 0,
    'func-names': 0,
  }
};

```

.prettierrc

```

{
  "singleQuote": true
}

```

components/button/index.tsx

```

+   const title = "hello";

```

.vscode/settings.json

```

{
  "editor.codeActionsOnSave": {
    "source.fixAll.eslint": true
  },
  "files.autoSave": "afterDelay"
}

```

7.editorconfig

```

# top-most EditorConfig file
root = true

# Unix-style newlines with a newline ending every file
[*.{js,css}]
end_of_line = lf
insert_final_newline = true
indent_style = space
indent_size = 2

```

8. git hook

```

yarn add husky --dev
npm set-script prepare "husky install"
npm run prepare

```

- **pre-commit**在git add提交之后，然后执行git commit时执行，脚本执行没报错就继续提交，反之就驳回提交的操作
- 可以在 git commit 之前检查代码，保证所有提交到版本库中的代码都是符合规范的

```

npx husky add .husky/pre-commit "npx lint-staged"

```

```

{
  "**.{js,ts,jsx,tsx}": "eslint"
}

```

```

yarn add commitizen cz-customizable @commitlint/cli @commitlint/config-conventional --dev

```

```

npx husky add .husky/commit-msg "npx --no-install commitlint --edit $1"

```

```

module.exports = {
  types: [
    { value: "feat", name: "feat:一个新特性" },
    { value: "fix", name: "fix:修复BUG" },
  ],
  scopes: [{ name: "admin" }, { name: "user" }],
};

```

commitlint.config.js

```

module.exports = {
  extends: ['@commitlint/config-conventional'],
};

```

- 可以在 git push 之前执行单元测试,保证所有的提交的代码经过的单元测试

```

npx husky add .husky/pre-push "npm run test"

```

npm publish

9. 编译发布

```

yarn add rimraf gulp gulp-typescript gulp-babel merge2 --dev

npm version patch
npm publish
cat ~/.npmrc

```

gulpfile.js

```

const gulp = require('gulp');
const path = require('path');
const rimraf = require('rimraf');
const ts = require('gulp-typescript');
const babel = require('gulp-babel');
const merge2 = require('merge2');
const { compilerOptions } = require('./tsconfig.json');

const tsConfig = {
  noUnusedParameters: true,
  noUnusedLocals: true,
  strictNullChecks: true,
  target: 'es6',
  jsx: 'preserve',
  moduleResolution: 'node',
  declaration: true,
  allowSyntheticDefaultImports: true,
  ...compilerOptions,
};

const babelConfig = require('./babel.config');

const source = [
  'components/**/*.{js,ts,jsx,tsx}',
  '!components/**/*.stories.{js,ts,jsx,tsx}',
  '!components/**/*.e2e/**',
  '!components/**/*.unit/**',
];

const base = path.join(process.cwd(), 'components');
function getProjectPath(filePath) {
  return path.join(process.cwd(), filePath);
}

const libDir = getProjectPath('lib');
const esDir = getProjectPath('es');

gulp.task('compile-with-es', (done) => {
  console.log('Compile to es...');
  compile(false).on('finish', done);
});

gulp.task('compile-with-lib', (done) => {
  console.log('Compile to js...');
  compile().on('finish', done);
});

gulp.task('compile', gulp.parallel('compile-with-es', 'compile-with-lib'));

function compile(modules) {
  const targetDir = modules === false ? esDir : libDir;
  rimraf.sync(targetDir);
  const { js, dts } = gulp.src(source, { base }).pipe(ts(tsConfig));
  const dtsFileStream = dts.pipe(gulp.dest(targetDir));
  let jsFileStream = js;
  if (modules) {
    jsFileStream = js.pipe(babel(babelConfig));
  }
  jsFileStream = jsFileStream.pipe(gulp.dest(targetDir));
  return merge2([jsFileStream, dtsFileStream]);
}

```

package.json

```

{
+ "main": "lib/index.js",
+ "module": "es/index.js",
+ "unpkg": "dist/antd.js",
+ "typings": "lib/index.d.ts",
+ "files": [
+   "dist",
+   "es",
+   "lib"
+ ],
}

```

10. 持续集成

- [Travis CI \(https://www.travis-ci.com/\)](https://www.travis-ci.com/)提供的是持续集成服务（Continuous Integration，简称 CI）。它绑定 Github 上面的项目，只要有新的代码，就会自动抓取。然后，提供一个运行环境，执行测试，完成构建，还能部署到服务器

```

language: node_js
node_js:
  - "stable"
cache:
  directories:
    - node_modules
env:
  - CI=true
install:
  - yarn config set registry https://registry.npm.taobao.org
  - yarn install
script:
  - npm run build-storybook
  - npm version patch
deploy:
  - provider: pages
    skip_cleanup: true
    github_token: $GITHUB_TOKEN
    local_dir: storybook-static
    on:
      branch: master
  - provider: npm
    email: zhang_renyang@126.com
    api_key: "$NPM_TOKEN"
    skip_cleanup: true
    on:
      branch: master

```

