

Angular 2 CLI Reference

ng new

```
$ ng new <project-name> [options]
```

创建一个新的Angular项目。默认情况下，项目在当前目录下创建。

选项:

`--dry-run` 别名: `'d'`

只输出创建的文件和操作执行，实际上不创建项目。

`--verbose` 别名: `'v'`

输出更多信息。

`--skip-npm`

创建项目后不要运行任何npm命令。

`--skip-git`

不要为项目创建一个git存储库。

`--directory`

父目录创建新项目。

ng init

```
$ ng init <project-name> [options]
```

在当前文件夹中创建一个新的Angular项目。

选项:

```
--dry-run 别名: 'd'
```

只输出创建的文件和操作执行，实际上不创建项目。

```
--verbose 别名: 'v'
```

输出更多信息。

```
--skip-npm
```

创建项目后不要运行任何npm命令。

```
--name
```

要创建的项目的名称。

ng completion

```
$ ng completion
```

将ng命令的自动完成功能添加到shell中。

ng doc

```
$ ng doc <keyword>
```

打开一个浏览器窗口, 在Angular文档中搜索关键字。

ng e2e

```
$ ng e2e
```

使用量角器(protractor)运行应用程序中定义的所有端到端测试。

ng format

```
$ ng format
```

ng format使用clang-format格式化此项目的代码。

ng generate

```
$ ng generate <type> [options]
```

```
# 别名: 'g'
```

在项目中生成新代码。

有效类型:

```
component <path/to/component-name>
生成一个组件。

directive <path/to/directive-name>
生成一个指令。

route <route/to/route-component>
生成一个路由。名称应为RouteConfig中使用的路由。

pipe <path/to/pipe-name>
生成一个管道。

service <path/to/service-name>
生成一个服务。
```

生成的组件有自己的目录，除非指定了`-flat`选项。

选项:

```
--flat
不要在自己的目录中创建代码。

--route=<route>
指定父路由。仅用于生成组件和路由。默认为指定的路径。

--skip-router-generation
跳过生成父路由配置。仅适用于路由。

--default
指定路由应为默认路由。

--lazy
指定路由是延迟的。默认为true。
```

ng get

```
$ ng get <path1, path2, ...pathN> [options]
```

从Angular CLI配置获取值。

`pathN`参数是一个有效的JavaScript路径，如“`users[1].userName`”。

如果未设置该值，将显示“undefined”。

此命令默认情况下仅在项目目录中工作。

选项:

```
--global
```

返回全局配置值，而不是本地配置值（如果都设置）。

此选项还使命令在项目目录外工作。

ng set

```
$ ng set <path1>=value1, path2=value2, ...pathN=valueN>  
[options]
```

在Angular CLI配置中设置值。

默认情况下，如果在项目内部运行，则设置项目配置中的值，如果不在项目内部，则失败。

pathN参数是一个有效的JavaScript路径，如“users[1].userName”。

该值将被强制转换为正确的类型，或者如果类型无法强制，则会抛出错误。

选项:

```
--global
```

设置全局配置值，而不是本地配置值。

这也使得ng set在项目之外工作。

ng github-pages:deploy

```
$ ng github-pages:deploy [options]
```

构建生产应用程序，设置GitHub存储库，然后发布应用程序。

选项:

```
--message=<message>
```

提交消息将与构建一起包括。默认为“new gh-pages version”。

```
--environment=<env>
```

The Angular environment to build. Defaults to "production".

构建的Angular环境。默认为“生产 (production)”。

```
--branch=<branch-name>
```

推送页面至的git分支。默认为“gh-branch”。

```
--skip-build
```

在发布之前跳过构建项目。

```
--gh-token=<token>
```

API令牌用于部署。必要。

```
--gh-username=<username>
```

要使用的Github用户名。必c。

ng lint

```
$ ng lint
```

运行codelyzer linter在项目上。

ng test

```
$ ng test [options]
```

运行单元测试，使用karma。

Options:

```
--watch
```

继续运行测试。默认为true。

```
--browsers, --colors, --reporters, --port, --log-level
```

些参数直接传递给karma。

ng version

```
$ ng version  
# 别名:  
$ ng v  
$ ng -v
```

输出 `angular-cli`, `node` 和操作系统的版本。