# ng e2e

Builds and serves an Angular app, then runs end-to-end tests using Protractor.

ng **e2e** <*project*> [*options*]

ng **e** <*project*> [*options*]

## [**mode\_edit**](https://github.com/angular/angular-cli/edit/master/packages/angular/cli/commands/e2e-long.md?message=docs%3A%20describe%20your%20change...)[**code**](https://github.com/angular/angular-cli/tree/v8.3.13/packages/angular/cli/commands/e2e-long.md)

## **Description**

Must be executed from within a workspace directory. When a project name is not supplied, it will execute for all projects.

## **Arguments**

| ARGUMENT | DESCRIPTION |
| --- | --- |
| <project> | The name of the project to build. Can be an application or a library. |

## **Options**

| OPTION | DESCRIPTION |
| --- | --- |
| --baseUrl=baseUrl | Base URL for protractor to connect to. |
| --configuration=configuration | A named build target, as specified in the "configurations" section of angular.json. Each named target is accompanied by a configuration of option defaults for that target. Setting this explicitly overrides the "--prod" flag  Aliases: -c |
| --devServerTarget=devServerTarget | Dev server target to run tests against. |
| --elementExplorer=true|false | Start Protractor's Element Explorer for debugging.  Default: false |
| --help=  true|false|json|JSON | Shows a help message for this command in the console.  Default: false |
| --host=host | Host to listen on. |
| --port | The port to use to serve the application. |
| --prod=true|false | Shorthand for "--configuration=production". When true, sets the build configuration to the production target. By default, the production target is set up in the workspace configuration such that all builds make use of bundling, limited tree-shaking, and also limited dead code elimination. |
| --protractorConfig=  protractorConfig | The name of the Protractor configuration file. |
| --specs | Override specs in the protractor config. |
| --suite=suite | Override suite in the protractor config. |
| --webdriverUpdate=true|false | Try to update webdriver.  Default: true |

element(by.id('firstName'))  
  
element(by.css('.signout'))  
  
element(by.model('address.city'))   
  
element(by.binding('address.city'));   
  
element(by.input('firstName'));  
  
element(by.input('firstName')).clear();  
  
element(by.buttonText('Close'));  
  
element(by.partialButtonText('Save'));  
  
element(by.linkText('Save'));  
  
element(by.partialLinkText('Save'));  
  
element(by.css('img[src\*='assets/img/profile.png']')); element(by.css('.pet .cat'));element(by.cssContainingText('.pet', 'Dog'));  
  
allColors = element.all(by.options('c c in colors'));

**Cheatsheet for typing (sendKeys):**

element(by.id('firstName').sendKeys("John");  
  
sendKeys(Key.ENTER);  
  
sendKeys(Key.TAB);sendKeys(Key.BACK\_SPACE)element(by.id('user\_name')).clear()

**Cheatsheet for collection:**

var list = element.all(by.css('.items));  
  
var list2 = element.all(by.repeater('personhome.results'));  
  
expect(list.count()).toBe(3);  
  
expect(list.get(0).getText()).toBe('First’)  
  
expect(list.get(1).getText()).toBe('Second’)  
  
expect(list.first().getText()).toBe('First’)  
  
expect(list.last().getText()).toBe('Last’)

import { browser, element, by, Key } from 'protractor';

export class PokemonPage {

navigateTo() {

return browser.get('/pokemon');

}

getPokemonCardElements() {

return element.all(by.css('.card--media'));

}

getFirstPokemonCardElement() {

return element(by.css('.card--media'));

}

getOpenModalElement() {

return element(by.tagName('app-pokemon-modal'));

}

getOpenModalHeadingElement() {

return element(by.css('app-pokemon-modal h1'));

}

selectNextKey() {

browser

.actions()

.sendKeys(Key.ARROW\_RIGHT)

.perform();

}

selectPrevKey() {

browser

.actions()

.sendKeys(Key.ARROW\_LEFT)

.perform();

}

selectEscapeKey() {

browser

.actions()

.sendKeys(Key.ESCAPE)

.perform();

}

}