

# Building and Serving

---



**John Papa**

PRINCIPAL DEVELOPER ADVOCATE

@john\_papa [www.johnpapa.net](http://www.johnpapa.net)



# Building and Serving



**Building with the Angular CLI**

**Build targets**

**Exploring the builds**

**Serving**

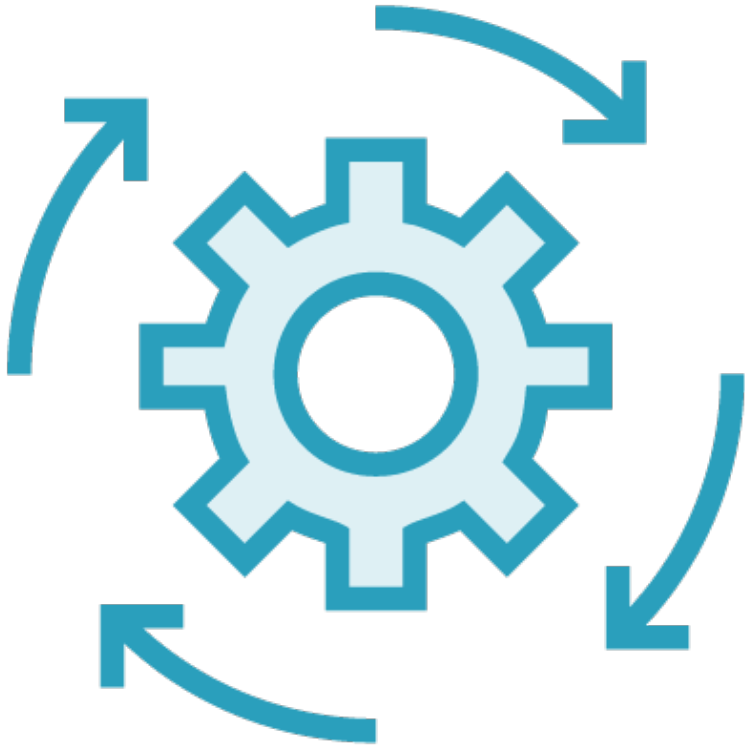
**Development cycle**



# Building Angular

---





# Builds

Compile the app to an output directory

Build targets determine the output

All builds use bundling

Prod builds add uglification and tree-shaking



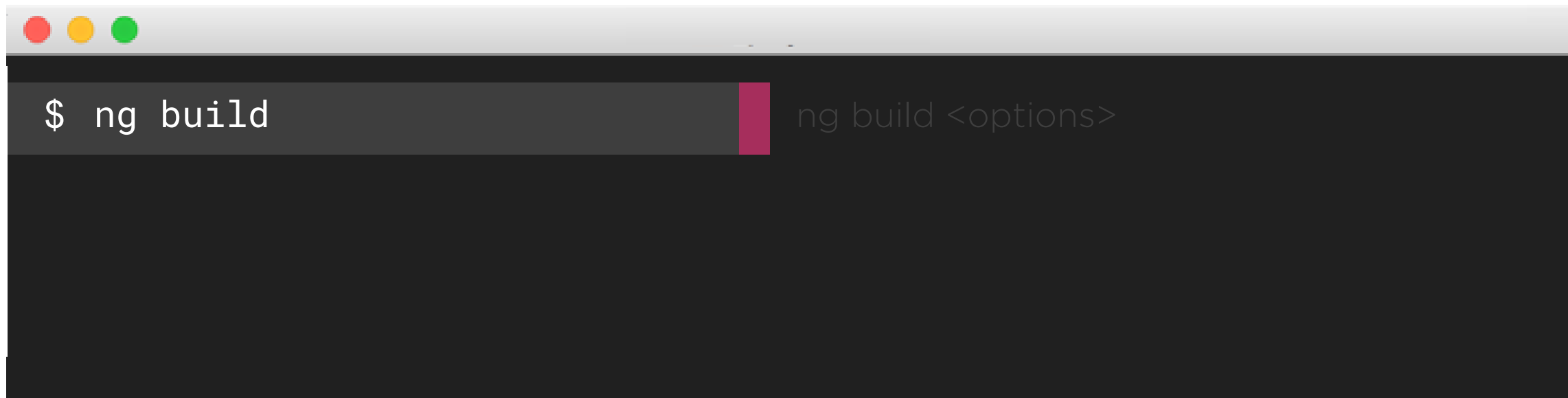
# Getting Help

Add the **--help** flag



```
$ ng build --help
```

A terminal window with a dark background and a light gray title bar. The title bar has three colored window control buttons (red, yellow, green) on the left. The terminal displays the command `$ ng build --help` in a white monospaced font.



```
$ ng build
```

ng build <options>

## Development Builds

**Ideal for development**

**Output to `/dist` folder, by default (see `.angular-cli.json`)**



# Exploring the Build Bundles

## File

## Description

inline.bundle.js

Webpack runtime

main.bundle.js

App code

polyfills.bundle.js

Polyfills

styles.bundle.js

Styles

vendor.bundle.js

Angular and other vendor files



# Exploring the Source

Analyze the sourcemap to draw a dependency map

Reveals exactly which modules and classes are in the bundle

<https://github.com/danvk/source-map-explorer/>



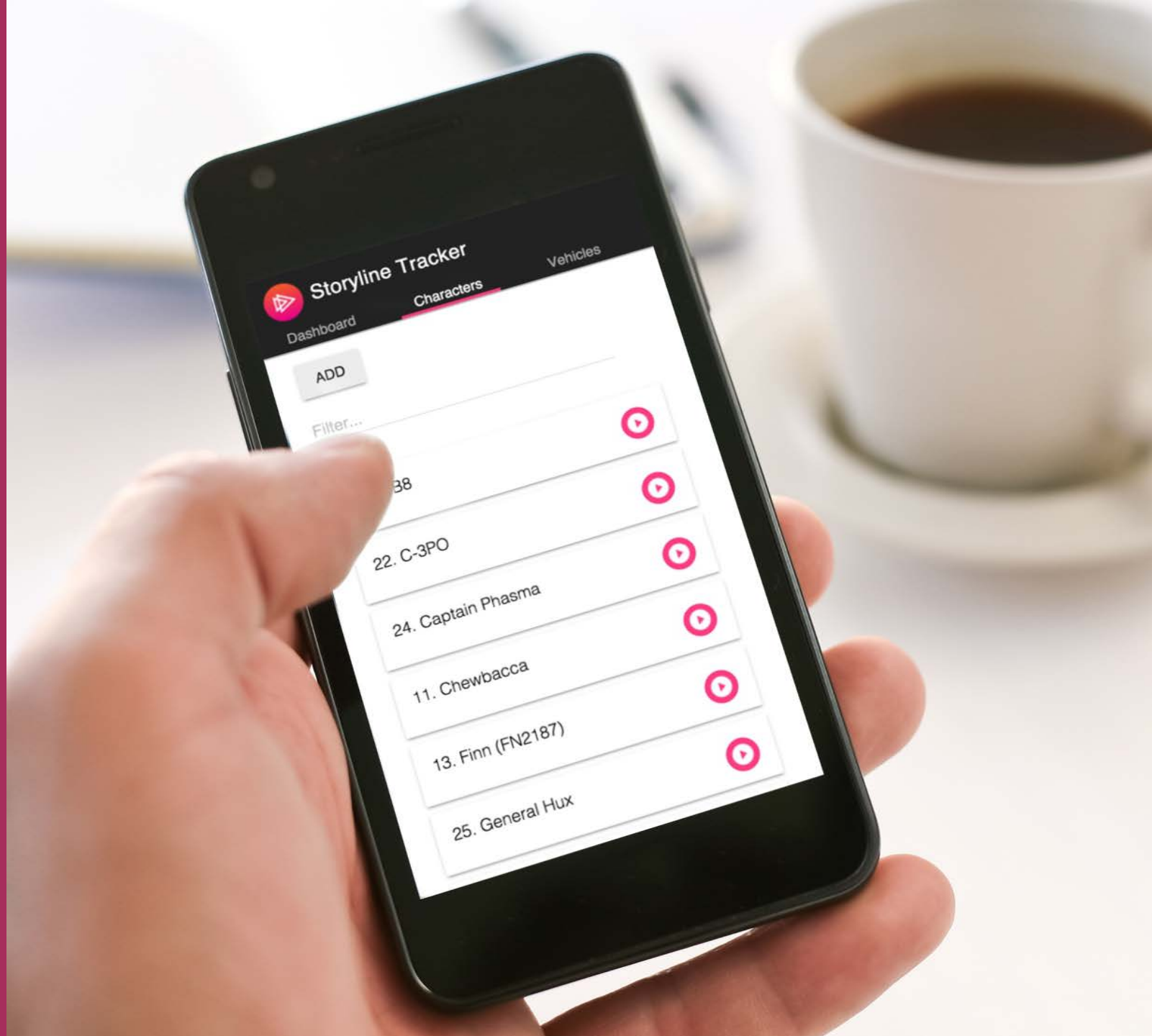
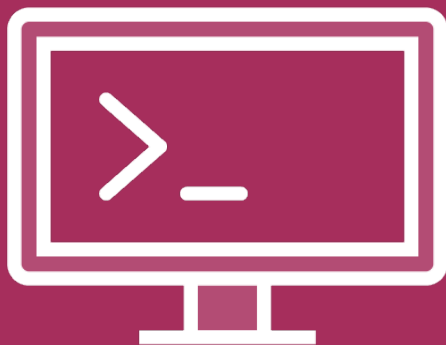
```
$ npm install source-map-explorer --save-dev
```

```
$ node_modules/.bin/source-map-explorer dist/main.bundle.js
```





# Demo



# Build Targets and Environments

---



# Environment

Indicates which file to use between  
environment.prod.ts and environment.ts

# Target

Defines how (and if) the files are optimized

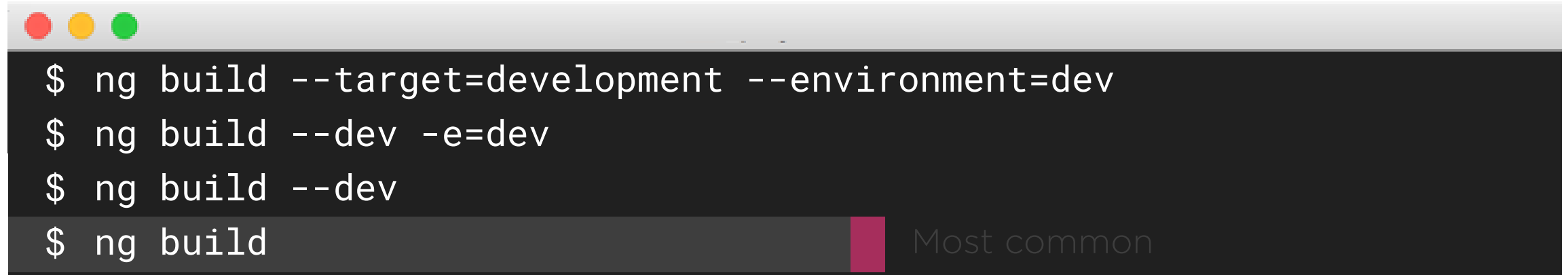


# Comparing Dev and Prod Build Targets

	ng build	ng build --prod
<b>Environment</b>	environment.ts	environment.prod.ts
<b>Cache-busting</b>	only images referenced in css	all build files
<b>Source maps</b>	generated	not generated
<b>Extracted CSS</b>	global CSS output to .js	yes, to css file(s)
<b>Uglification</b>	no	yes
<b>Tree-Shaking</b>	no	yes
<b>AOT</b>	no	yes
<b>Bundling</b>	yes	yes



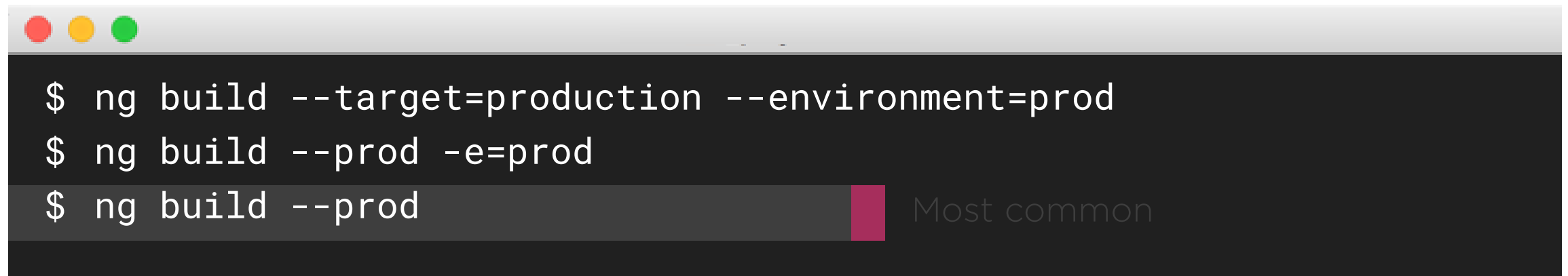
# Build Target and Environment Aliases



```
$ ng build --target=development --environment=dev
$ ng build --dev -e=dev
$ ng build --dev
$ ng build
```

Most common

A terminal window with a dark background and light gray text. It shows four commands for building in development mode. The first command is the full syntax. The second and third are shorthand versions. The fourth is the most common shorthand. A pink highlight bar is under the fourth command, and the text 'Most common' is to its right.



```
$ ng build --target=production --environment=prod
$ ng build --prod -e=prod
$ ng build --prod
```

Most common

A terminal window with a dark background and light gray text. It shows three commands for building in production mode. The first command is the full syntax. The second and third are shorthand versions. The third is the most common shorthand. A pink highlight bar is under the third command, and the text 'Most common' is to its right.

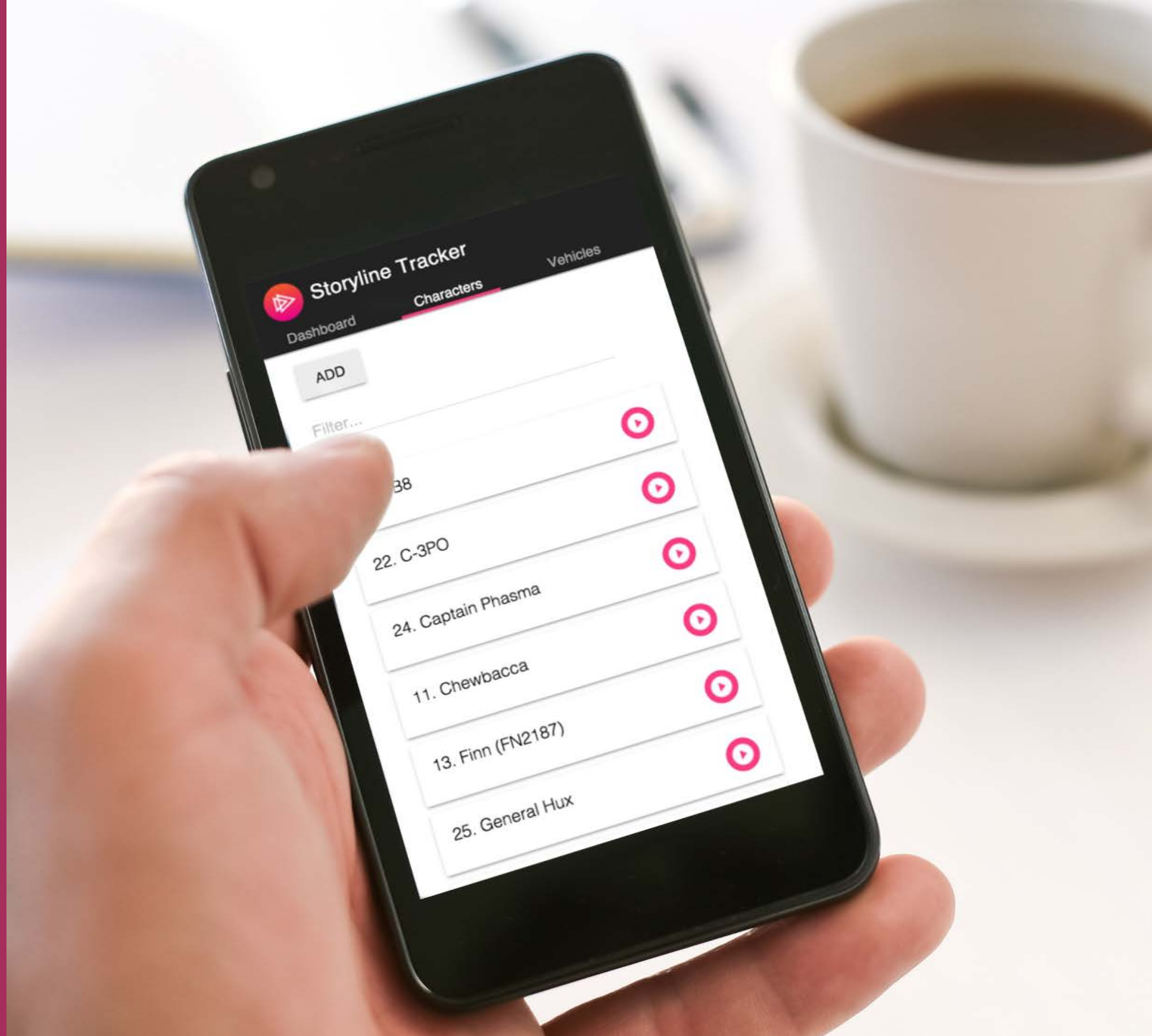
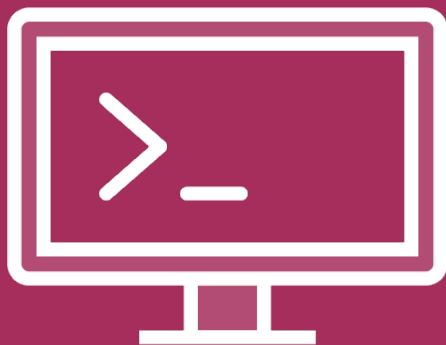


# Common ng build Options

Options	Alias	Description
--sourcemap	-sm	Generate a source map
--aot		Ahead of Time compilation
--watch	-w	Watch and rebuild
--environment	-e	Build environment
--target	-t	Build target
--dev		Shortcut for dev env and target
--prod		Shortcut for prod env and target



# Demo



# Serving Angular

---







# Serving

Compiled output is served from memory

Efficient for reloads



# Getting Help

Add the **--help** flag



```
$ ng serve --help
```

A terminal window with a dark background and a light gray title bar. The title bar has three colored window control buttons (red, yellow, green) on the left. The terminal displays the command `$ ng serve --help` in white text.

# Common `ng serve` Options

All build options are available on `serve`, plus the following ...

## Options

## Alias

## Description

`--open`

`-o`

Opens in the default browser

`--port`

`-p`

Port to listen to when serving

`--live-reload`

`-lr`

Reload when changes occur

`--ssl`

Serve using HTTPS

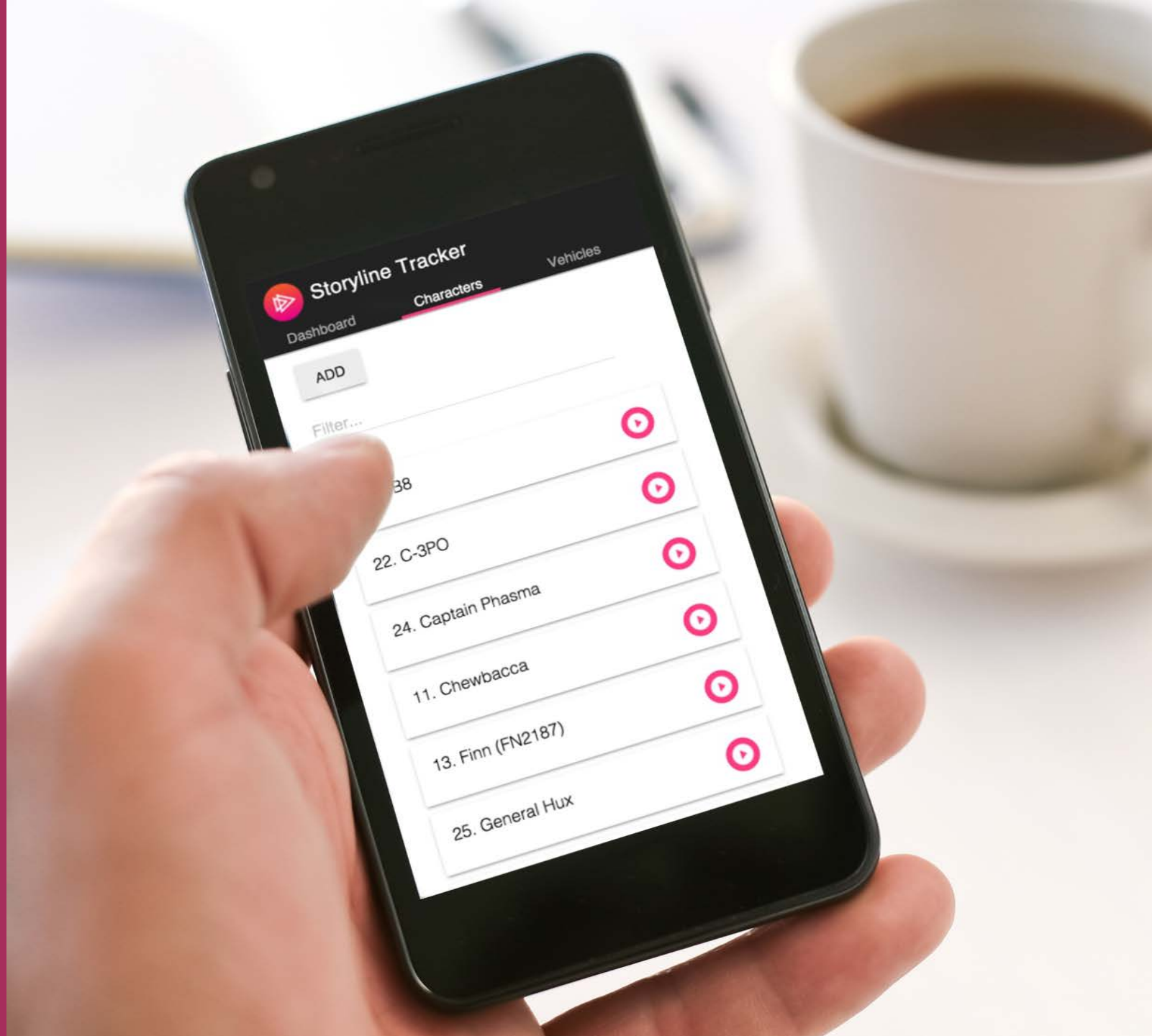
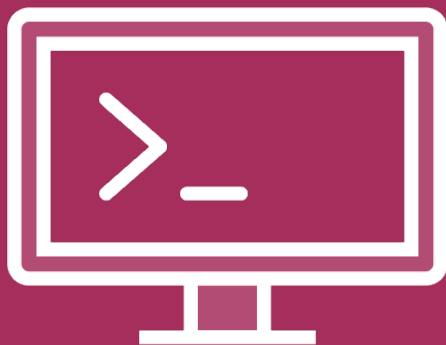
`--proxy-config`

`-pc`

Proxy configuration file



# Demo





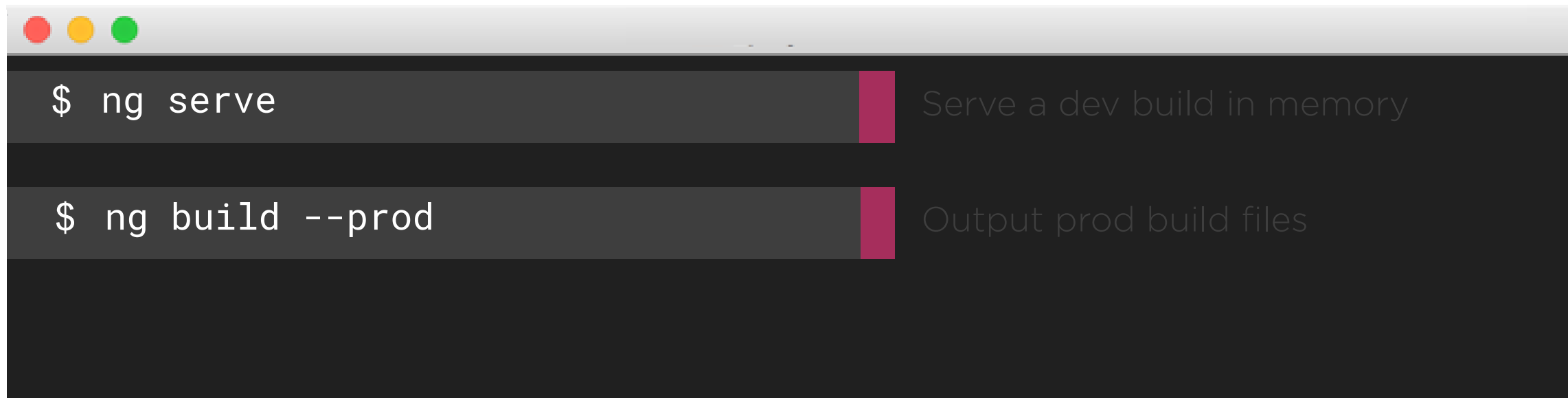
# What Else

Building with external libraries

Angular Material

Ejecting WebPack configuration





## Remember These Commands

Fast development cycles with **ng serve**

When ready to promote your code, use **ng build --prod**



# Building and Serving



**Build to output the production files**

**Many customizations**

**Serve is intended for development**

**Build with external libraries**

**Eject and use WebPack**

