



EBook Gratis

APRENDIZAJE angular-cli

Free unaffiliated eBook created from
Stack Overflow contributors.

#angular-cli

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con angular-cli.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Generando y sirviendo un proyecto Angular a través de un servidor de desarrollo.....	2
CLI angular - Los pasos básicos.....	3
Capítulo 2: Despliegue del proyecto angular-cli en el servidor apache tomcat 8.0.14.....	4
Introducción.....	4
Examples.....	4
Se tomaron pasos necesarios antes de implementar el proyecto angular-cli para la compilaci.....	4
Comando de compilación Angular-Cli para construir un paquete de proyecto para la implement.....	4
Creación del archivo war para el despliegue de producción del proyecto angular-cli en el s.....	4
Configuración del apache tomcat para el despliegue del proyecto angular-cli.....	5
Capítulo 3: Trabajar con angular-cli: Generando componentes, directivas, tuberías, servici.....	6
Introducción.....	6
Sintaxis.....	6
Parámetros.....	6
Examples.....	7
Uso "generar comando".....	7
Componentes generadores.....	8
Directivas generadoras.....	10
Servicios de generacion.....	11
Tubos generadores.....	12
Módulos generadores.....	12
Creditos.....	14

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [angular-cli](#)

It is an unofficial and free angular-cli ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official angular-cli.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con angular-cli

Observaciones

Esta sección proporciona una descripción general de qué es angular-cli y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de angular-cli, y vincular a los temas relacionados. Dado que la Documentación para angular-cli es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

Nota: las versiones de CLI angular están en rápido desarrollo. Esta documentación apunta a la última versión.

Prerrequisitos

Para ejecutar y trabajar con la última versión, Angular CLI y el proyecto tienen dependencias que requieren el nodo v6.9.0 o superior.

Preparar

Asegúrese de que esté instalada una versión de nodo que sea compatible con la CLI

Instale el CLI angular a nivel mundial. Instala la última versión.

`npm i @angular/cli -g` **O** `yarn global add @angular/cli`, dependiendo del administrador de paquetes en uso.

`ng help comando` `ng help` proporcionará los comandos ejecutables.

Generando y sirviendo un proyecto Angular a través de un servidor de desarrollo

Para crear un nuevo proyecto `ng new [project-name]` que inicializa git. Instale paquetes para herramientas a través de npm. Crea el proyecto con éxito.

`cd [project-name]` y ejecute `npm start` **O** `ng serve`

Ejecutará el servidor en el puerto predeterminado dado.

La aplicación se actualizará de acuerdo con los cambios realizados en el directorio. El puerto HTTP predeterminado y el que usa el servidor LiveReload se pueden cambiar usando el comando a continuación

```
ng serve --host 0.0.0.0 --port 4201 --live-reload-port 49153
```

CLI angular - Los pasos básicos

1. Deberá instalar node.js - <https://nodejs.org/en/>
2. npm install -g @angular/cli: instala la CLI ejecutando este comando en el terminal
3. ng new projectname: después de ejecutar este comando en el terminal, creará una nueva subcarpeta llamada projectname en su carpeta actual.
4. cd projectname - vaya a la subcarpeta, donde está su proyecto
5. ng serve - ejecuta el proyecto. Estará disponible en <http://localhost:4200>

Lea [Empezando con angular-cli en línea](https://riptutorial.com/es/angular-cli/topic/9275/empezando-con-angular-cli): <https://riptutorial.com/es/angular-cli/topic/9275/empezando-con-angular-cli>

Capítulo 2: Despliegue del proyecto angular-cli en el servidor apache tomcat 8.0.14

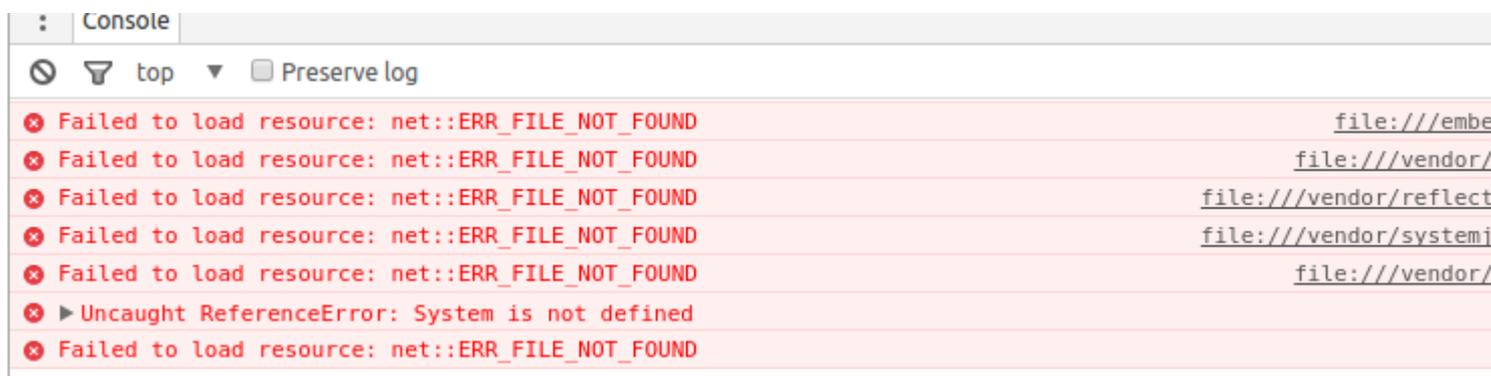
Introducción

Este tema cubriría cómo el proyecto angular-cli está listo para la compilación de producción, cuáles son todos los pasos necesarios tomados antes del despliegue, cómo crear un archivo de guerra para el despliegue del proyecto y, finalmente, cómo configurar el apache tomcat para el despliegue del proyecto angular-cli.

Examples

Se tomaron pasos necesarios antes de implementar el proyecto angular-cli para la compilación de producción.

Antes de desplegar el proyecto angular en el servidor, necesitamos construir un proyecto angular para la producción. También debemos cambiar la ruta de enrutamiento en el archivo index.html de `<base href="/">` to `<base href="./">` si no se realiza, su proyecto no se cargará correctamente, habrá algunos error de enrutamiento que dice el archivo 404 no encontrado.



Comando de compilación Angular-Cli para construir un paquete de proyecto para la implementación de producción

```
ng build -prod
```

El comando anterior con una opción adicional como `-prod` generaría el paquete de proyecto de compilación de producción. Una vez que el comando anterior se ejecute en el directorio raíz de su proyecto, aparecerá un directorio llamado `dist` . En el que aparece todo el conjunto de compilación de producción de su proyecto.

Creación del archivo war para el despliegue de producción del proyecto angular-cli en el servidor tomache apache

Una vez que el directorio **dist** esté listo con sus paquetes construidos de producción. Simplemente abra el directorio **dist** y abra el símbolo del sistema escriba el siguiente comando para crear el archivo **war** para implementar su proyecto en el servidor tomache apache.

```
jar cvf dist.war .
```

Una vez que los comandos jar anteriores se ejecutan. **Generaría** un archivo **dist.war** dentro del directorio **dist** .

Configuración del apache tomcat para el despliegue del proyecto angular-cli.

1. Corte / Copie el archivo **dist.war** del directorio **dist** y colóquelo en el directorio **apapp tomcat webapp** .
2. Vaya a la carpeta **bin de** apache tomcat y haga doble clic en el archivo **startup.bat** .
3. Ahora el servidor tomcat ejecutará el archivo **dist.war** y arrancará el servidor **catalina tomcat**.
4. Una vez que el servidor **Catalina Tomcat** se inicie, abra el navegador web y escriba **localhost: 8080 / dist** y toque la tecla enter para que su proyecto se ejecute en la ventana del navegador web.

Lea [Despliegue del proyecto angular-cli en el servidor apache tomcat 8.0.14 en línea:](https://riptutorial.com/es/angular-cli/topic/9780/despliegue-del-proyecto-angular-cli-en-el-servidor-apache-tomcat-8-0-14)

<https://riptutorial.com/es/angular-cli/topic/9780/despliegue-del-proyecto-angular-cli-en-el-servidor-apache-tomcat-8-0-14>

Capítulo 3: Trabajar con angular-cli: Generando componentes, directivas, tuberías, servicios, etc.

Introducción

La herramienta angular-cli puede ayudarte a organizar diferentes partes de una aplicación angular (componentes, directivas, tuberías, servicios, clases, guardas, interfaces, enumeraciones y módulos).

Sintaxis

- `ng generar [componente | directiva | servicio | pipa clase | enumerar interfaz | guardia módulo] [nombre] [banderas ...]`
- `ng g [c | d | s | p | cl | e | i | g | m] [nombre] [banderas ...]`

Parámetros

Parámetro	Descripción
<code>component</code> o <code>c</code>	Se utiliza para generar componente
<code>directive</code> o <code>d</code>	Se utiliza para generar directivas.
<code>service</code> o <code>s</code>	Se utiliza para generar servicios.
<code>pipe</code> o <code>p</code>	Se utiliza para generar tuberías.
<code>class</code> o <code>cl</code>	Se utiliza para generar clases.
<code>enum</code> o <code>e</code>	Se utiliza para generar enumeraciones.
<code>interfaces</code> o <code>i</code>	Se utiliza para generar interfaces.
<code>guard</code> o <code>g</code>	Se utiliza para generar guardias.
<code>module</code> o <code>m</code>	Se utiliza para generar módulos.
<code>--flat</code> o <code>-f</code>	Se utiliza para habilitar / deshabilitar la creación de directorios
<code>--inline-template</code> o <code>-it</code>	Se utiliza para habilitar / deshabilitar plantillas html en línea en componentes
<code>--inline-style</code> o <code>-is</code>	Se utiliza para habilitar / deshabilitar estilos en línea en

Parámetro	Descripción
	componentes
<code>--prefix o -p</code>	Se usa para deshabilitar o cambiar el prefijo
<code>--spec o -s</code>	Se usa para habilitar / deshabilitar la creación de archivos <code>.spec</code>
<code>--skip-import</code>	Se utiliza para omitir la importación del módulo.
<code>--app o -a</code>	Se utiliza para especificar el nombre de la aplicación para usar
<code>--module o -m</code>	Se utiliza para especificar el módulo de declaración.
<code>--view-encapsulation o -ve</code>	Se utiliza para especificar la estrategia de encapsulación de vista en componentes
<code>--change-detection o -cd</code>	Se utiliza para especificar la estrategia de detección de cambios en los componentes.
<code>--routing o -r</code>	Se utiliza para especificar si se debe generar el archivo del módulo de enrutamiento

Examples

Uso "generar comando"

Puede usar el comando `ng generate o ng g` para generar bloques de construcción angulares (componentes, servicios, tuberías, etc.).

Puede encontrar todos los **planos** posibles en la siguiente tabla:

Andamio	Uso	Acortado
Componente	<code>ng generate component component-name</code>	<code>ng gc component-name</code>
Directiva	<code>ng generate directive directive-name</code>	<code>ng gd directive-name</code>
Tubo	<code>ng generate pipe pipe-name</code>	<code>ng gp pipe-name</code>
Servicio	<code>ng generate service service-name</code>	<code>ng gs service-name</code>
Clase	<code>ng generate class class-name</code>	<code>ng g cl class-name</code>
Guardia	<code>ng generate guard guard-name</code>	<code>ng gg guard-name</code>
Interfaz	<code>ng generate interface interface-name</code>	<code>ng gi interface-name</code>
Enumerar	<code>ng generate enum enum-name</code>	<code>ng ge enum-name</code>

Andamio	Uso	Acortado
Módulo	<code>ng generate module module-name</code>	<code>ng gm module-name</code>

Entonces, por ejemplo, si ejecuta `ng generate component user-list - angular-cli:`

- cree `user-list` directorio de la `user-list` en la carpeta `src/app` o en la carpeta donde ejecutó el comando.
- dentro de ese directorio genere 4 archivos (`user-list.component.ts` , `user-list.component.html` , `user-list.component.css` y `user-list.component.spec.ts`)
- agregue `user-list` como una declaración en el decorador `@NgModule` del módulo más cercano.

Componentes generadores

Para agregar un componente con un selector `[prefix]-user-list` , ejecute:

```
$ ng g c user-list

installing component
  create src/app/user-list/user-list.component.css
  create src/app/user-list/user-list.component.html
  create src/app/user-list/user-list.component.spec.ts
  create src/app/user-list/user-list.component.ts
  update src/app/app.module.ts
```

el prefijo evita las colisiones de nombres de elementos con componentes en otras aplicaciones y con elementos HTML nativos. Así, por ejemplo, si el prefijo es `app` - componente generado tendrá `app-user-list` selector.

- Para evitar el uso de prefijos, agregue `--prefix false` o `-p false` flag

```
$ ng g c user-list --prefix false
```

```
import { Component } from '@angular/core';

@Component({
  selector: 'user-list',
  templateUrl: './user-list.component.html',
  styleUrls: ['./user-list.component.css']
})
export class UserListComponent {}
```

- Para evitar la creación de archivos `.spec` , agregue `--spec false` o `-sp false` flag

```
$ ng g c user-list --spec false

installing component
  create src/app/user-list/user-list.component.css
  create src/app/user-list/user-list.component.html
  create src/app/user-list/user-list.component.ts
  update src/app/app.module.ts
```

- Para usar plantillas html en línea en lugar de plantillas externas, agregue `--inline-template` o `-it` flag

```
$ ng g c user-list --inline-template

installing component
  create src/app/user-list/user-list.component.css
  create src/app/user-list/user-list.component.spec.ts
  create src/app/user-list/user-list.component.ts
  update src/app/app.module.ts
```

- Para usar estilos en línea en lugar de estilos externos, agregue `--inline-style` o `-is` flag

```
$ ng g c user-list --inline-style

installing component
  create src/app/user-list/user-list.component.html
  create src/app/user-list/user-list.component.spec.ts
  create src/app/user-list/user-list.component.ts
  update src/app/app.module.ts
```

- Para evitar la creación de carpetas, agregue `--flat` o `-f` flag

```
$ ng g c user-list --flat

installing component
  create src/app/user-list.component.css
  create src/app/user-list.component.html
  create src/app/user-list.component.spec.ts
  create src/app/user-list.component.ts
  update src/app/app.module.ts
```

También puede combinar las banderas enumeradas anteriormente. Por ejemplo, para crear solamente `.component.ts` archivo sin `.css`, `.html`, `.spec` archivos y carpetas utilice el siguiente comando.

```
$ ng g c user-list -f -it -is -sp false

installing component
  create src/app/user-list.component.ts
  update src/app/app.module.ts
```

Todos `generate component` banderas de `generate component`:

Descripción	Bandera	Acortado	Valor por defecto
Impedir la creación de carpetas	<code>--flat</code>	<code>-f</code>	<code>false</code>
Evitar el uso de prefijos	<code>--prefix false</code>	<code>-p false</code>	<code>true</code>
Evitar la creación de archivos <code>.spec</code>	<code>--spec false</code>	<code>-sp false</code>	<code>true</code>
Habilitar plantillas html en línea	<code>--inline-template</code>	<code>-it</code>	<code>false</code>

Descripción	Bandera	Acortado	Valor por defecto
Habilitar estilos en línea	--inline-style	-is	false

Directivas generadoras

Para agregar una directiva con un selector `[prefix]Highlight` , ejecute:

```
$ ng g d highlight

installing directive
  create src/app/highlight.directive.spec.ts
  create src/app/highlight.directive.ts
  update src/app/app.module.ts
```

- Para evitar el uso de prefijos, agregue `--prefix false` o `-p false` flag

```
$ ng g d highlight --prefix false
```

```
import { Directive } from '@angular/core';

@Directive({
  selector: '[highlight]'
})
export class HighlightDirective {}
```

- Para evitar la creación de archivos `.spec` , agregue `--spec false` o `-sp false` flag

```
$ ng g d highlight --spec false

installing directive
  create src/app/highlight.directive.ts
  update src/app/app.module.ts
```

- Para habilitar la creación de carpetas, agregue `--flat false` o `-f false` flag

```
$ ng g d highlight --flat false

installing directive
  create src/app/highlight/highlight.directive.spec.ts
  create src/app/highlight/highlight.directive.ts
  update src/app/app.module.ts
```

También puede combinar las banderas enumeradas anteriormente. Por ejemplo, para crear solo el archivo `highlight.directive.ts` dentro de la carpeta de `highlight` sin el archivo `.spec` , use el siguiente comando.

```
$ ng g d highlight -f false -sp false

installing directive
  create src/app/highlight/highlight.directive.ts
  update src/app/app.module.ts
```

Todos `generate directive` banderas `generate directive` :

Descripción	Bandera	Acortado	Valor por defecto
Habilitar la creación de carpetas	<code>--flat false</code>	<code>-f false</code>	<code>true</code>
Evitar el uso de prefijos	<code>--prefix false</code>	<code>-p false</code>	<code>true</code>
Evitar la creación de archivos <code>.spec</code>	<code>--spec false</code>	<code>-sp false</code>	<code>true</code>

Servicios de generacion

Para agregar un servicio con un nombre `UserService` , ejecute:

```
$ ng g s user

installing service
  create src/app/user.service.spec.ts
  create src/app/user.service.ts
```

- Para evitar la creación de archivos `.spec` , agregue `--spec false` o `-sp false` flag

```
$ ng g s user --spec false

installing service
  create src/app/user.service.ts
```

- Para habilitar la creación de carpetas, agregue `--flat false` o `-f false` flag

```
$ ng g s user --flat false

installing service
  create src/app/user/user.service.spec.ts
  create src/app/user/user.service.ts
```

También puede combinar las banderas enumeradas anteriormente. Por ejemplo, para crear solo el archivo `user.service.ts` dentro de `user` carpeta del `user` sin el archivo `.spec` , use el siguiente comando.

```
$ ng g s user -f false -sp false

installing service
  create src/app/user/user.service.ts
```

Todos `generate service` banderas de `generate service` :

Descripción	Bandera	Acortado	Valor por defecto
Habilitar la creación de carpetas	<code>--flat false</code>	<code>-f false</code>	<code>true</code>

Descripción	Bandera	Acortado	Valor por defecto
Evitar la creación de archivos <code>.spec</code>	<code>--spec false</code>	<code>-sp false</code>	<code>true</code>

Tubos generadores

Para agregar una tubería con un nombre `searchByName` , ejecute:

```
$ ng g p search-by-name

installing pipe
  create src/app/search-by-name.pipe.spec.ts
  create src/app/search-by-name.pipe.ts
  update src/app/app.module.ts
```

- Para evitar la creación de archivos `.spec` , agregue `--spec false` o `-sp false` flag

```
$ ng g p search-by-name --spec false

installing pipe
  create src/app/search-by-name.pipe.ts
  update src/app/app.module.ts
```

- Para habilitar la creación de carpetas, agregue `--flat false` o `-f false` flag

```
$ ng g p search-by-name --flat false

installing pipe
  create src/app/search-by-name/search-by-name.pipe.spec.ts
  create src/app/search-by-name/search-by-name.pipe.ts
  update src/app/app.module.ts
```

También puede combinar las banderas enumeradas anteriormente. Por ejemplo, para crear solo el archivo `search-by-name.pipe.ts` dentro de la carpeta `search-by-name` sin el archivo `.spec` , use el siguiente comando.

```
$ ng g p search-by-name -f false -sp false

installing pipe
  create src/app/search-by-name/search-by-name.pipe.ts
  update src/app/app.module.ts
```

Todos `generate pipe` banderas de `generate pipe` :

Descripción	Bandera	Acortado	Valor por defecto
Habilitar la creación de carpetas	<code>--flat false</code>	<code>-f false</code>	<code>true</code>
Evitar la creación de archivos <code>.spec</code>	<code>--spec false</code>	<code>-sp false</code>	<code>true</code>

Módulos generadores

Para agregar un módulo llamado `GuestModule` , ejecute:

```
$ ng g m guest

installing module
  create src/app/guest/guest.module.ts
```

- Para habilitar la creación de archivos `.spec` , agregue el `--spec o -sp`

```
$ ng g m guest --spec

installing module
  create src/app/guest/guest.module.spec.ts
  create src/app/guest/guest.module.ts
```

- Para habilitar el enrutamiento, agregue `--routing o -r` flag

```
$ ng g m guest --routing

installing module
  create src/app/guest/guest-routing.module.ts
  create src/app/guest/guest.module.ts
```

También puede combinar las banderas enumeradas anteriormente. Por ejemplo, para crear un módulo con enrutamiento y especificaciones, use el siguiente comando.

```
$ ng g m guest -sp -r

installing module
  create src/app/guest/guest-routing.module.ts
  create src/app/guest/guest.module.spec.ts
  create src/app/guest/guest.module.ts
```

Todos `generate module` indicadores de `generate module` :

Descripción	Bandera	Acortado	Valor por defecto
Habilitar la creación de archivos <code>.spec</code>	<code>--spec</code>	<code>-sp</code>	<code>false</code>
Habilitar enrutamiento	<code>--routing</code>	<code>-r</code>	<code>false</code>

Lea [Trabajar con angular-cli: Generando componentes, directivas, tuberías, servicios, etc. en línea](https://riptutorial.com/es/angular-cli/topic/9482/trabajar-con-angular-cli--generando-componentes--directivas--tuberias--servicios--etc-): <https://riptutorial.com/es/angular-cli/topic/9482/trabajar-con-angular-cli--generando-componentes--directivas--tuberias--servicios--etc->

Creditos

S. No	Capítulos	Contributors
1	Empezando con angular-cli	BogdanC , Community , Daedalon , Kaloyan , Saiyaff Farouk
2	Despliegue del proyecto angular-cli en el servidor apache tomcat 8.0.14	Gurudath G
3	Trabajar con angular-cli: Generando componentes, directivas, tuberías, servicios, etc.	Ketan Akbari , Saka7