# Servicios angular

# Servicios

## Inyección de dependencia en Angular

Las dependencias son servicios u objetos que una clase necesita para realizar su función. La inyección de dependencia, o DI, es un patrón de diseño en el que una clase solicita dependencias de fuentes externas en lugar de crearlas.

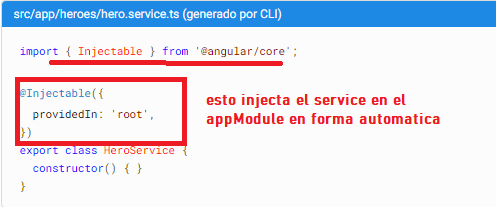
El marco DI de Angular proporciona dependencias a una clase en la creación de instancias. Utilice Angular DI para aumentar la flexibilidad y modularidad en sus aplicaciones.

## Creación de un servicio inyectable

Para generar una nueva HeroService clase en la src/app/héroes

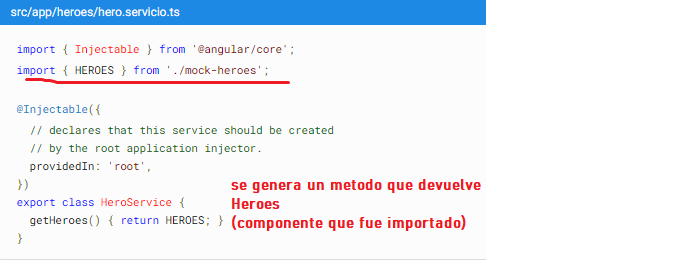
carpeta, use el siguiente comando de [CLI angular .](https://angular.io/cli)

ng generate service heroes/hero



El decorador especifica que Angular puede usar esta clase en el sistema DI. Los metadatos, significa que el es visible en toda la aplicación.@[Injectable](https://angular.io/api/core/Injectable)()providedIn: 'root'HeroService

A continuación, para obtener los datos simulados del héroe, agregue un getHeroes()método que devuelva los héroes de mock.heroes.ts.



Para mayor claridad y facilidad de mantenimiento, se recomienda que defina componentes y servicios en archivos separados.

Si combina un componente y un servicio en el mismo archivo, es importante definir primero el servicio y luego el componente. Si define el componente antes que el servicio, Angular devuelve un error de referencia nula en tiempo de ejecución.

## Uso de servicios en otros servicios

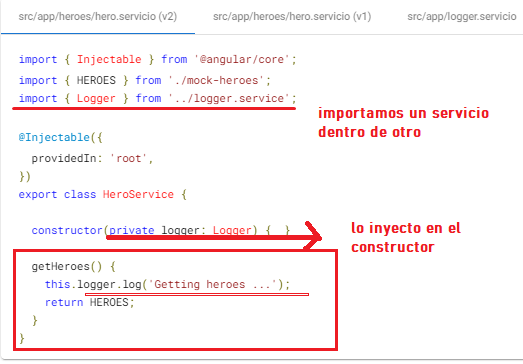
Cuando un servicio depende de otro servicio, siga el mismo patrón que inyectando en un componente. En el siguiente ejemplo HeroServicedepende de un Logger servicio para reportar sus actividades.

Primero, importe el Loggerservicio. A continuación, inyecte el Loggerservicio en HeroService constructor()especificándolo private logger: Logger entre paréntesis.

Cuando crea una clase que constructor()tiene parámetros, especifique el tipo y los metadatos sobre esos parámetros para que Angular pueda inyectar el servicio correcto.

Aquí, constructor()especifica un tipo de Loggery almacena la instancia de Logger en un campo privado llamado logger.

Las siguientes pestañas de código presentan el Loggerservicio y dos versiones de HeroService. La primera versión de HeroServiceno depende del Loggerservicio. La segunda versión revisada depende del Logger servicio.



Fuente : <https://angular.io/guide/dependency-injection>