edWisor

Angular - Services and DI

- Aditya Kumar
Chief Technology Officer, edwisor.com

We will cover with the following concepts

1) Why services?

2) What is DI and how services function?

Components are only meant for handling the user interaction

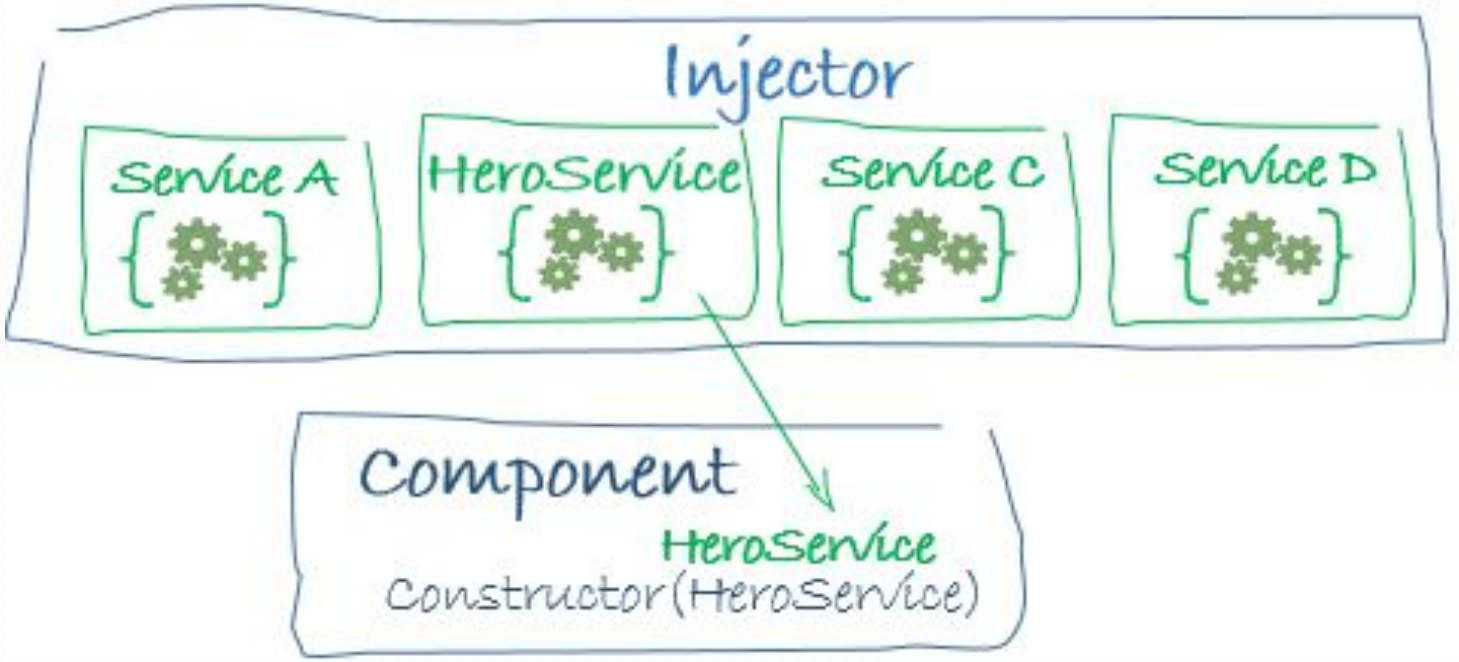
1) The idea is to keep the components as lean and simple. They should only deal with the functionality of that particular view or feature

2) Another reason why services are a separate entity than component is re-usability. Angular wanted to encourage re-usability but in more easy way than just nesting components or sharing things at component level. Hence Services!

3) Heavy lifting tasks like talking to servers, maintaining data and state information, keeping logs etc should not be a concern for component. Hence its all delegate to services.

4) Services are injected into the component using DI logic.

DI is how angular brings in the things needed by the components



https://angular.io/guide/architecture-services

Injector keeps a track of service instances needed

1) Angular creates an application wide injector at the time application is launched/bootstrapped.

2) Injector keeps a track of all service instances and tries to re-use them as much as possible.

3) When you specify a service name in providers array (of module or component), you are registering a dependency and that's where the process starts.

4) When Angular discovers that a component depends on a service, it first checks if the injector already has any existing instances of that service. If a requested service instance does not yet exist, the injector makes one using the registered provider, and adds it to the injector before returning the service to Angular. When all requested services have been resolved and returned, Angular can call the component's constructor with those services as argument

The next steps are ...

Security and optimizations