## Service

* Service is a broad category encompassing any value, function, or feature that an app needs.
* A service is typically a class with a narrow, well-defined purpose.
* A service should do something specific and do it well.
* Angular distinguishes components from services to increase modularity and reusability.
* A component can delegate certain tasks to services, such as fetching data from the server, validating user input, or logging directly to the console.
* By defining such processing tasks in an injectable service class, you make those tasks available to any component.
* You can also make your app more adaptable by injecting different providers of the same kind of service, as appropriate in different circumstances.
* Angular does help you by making it easy to factor your application logic into services and make those services available to components through dependency injection.

Dependency injection (DI)



DI is wired into the Angular framework and used everywhere to provide new components with the services or other things they need. Components consume services; that is, you can inject a service into a component, giving the component access to that service class.

To define a class as a service in Angular, use the @[Injectable](https://angular.io/api/core/Injectable)() decorator

* The injector is the main mechanism. Angular creates an application-wide injector for you during the bootstrap process, and additional injectors as needed. You don't have to create injectors.
* An injector creates dependencies, and maintains a container of dependency instances that it reuses if possible.
* A provider is an object that tells an injector how to obtain or create a dependency.

A dependency doesn't have to be a service—it could be a function, for example, or a value.

When Angular creates a new instance of a component class, it determines which services or other dependencies that component needs by looking at the constructor parameter types. For example, the constructor of BookListComponent needs BookService.