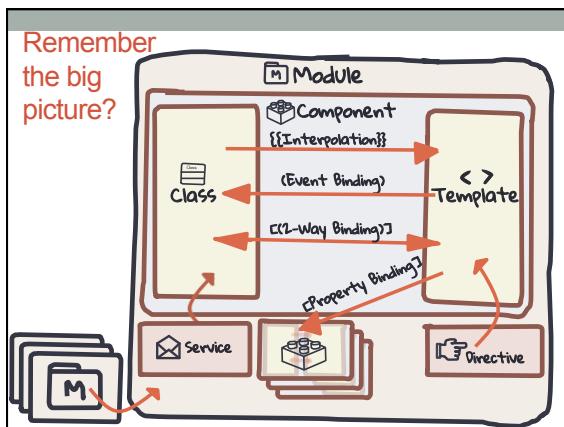


Services

(aka. Injectables. aka. Providers)

tl;dr

- Services are for sharing data or functionality across components.
- They must be added to the providers array of a module or component annotation and must be injected to be used.
- Provide them as low as possible but not so low they can't be shared
- Create services with ng generate service which will add the all-important @Injectable annotation



What's a service?

AJAX RESTful

Nope! None of the above.

So what is a service then?

A service is a holder of objects and activities that would be useful to one or more external things.

Vacuum bottom	Balance valves
Backwash	Company name
Pool service	
Clean baskets	Repair pump
Clean filter	Employees []
Test chemicals	Balance chlorine
Phone number	

Angular services aren't what you think at first

Services are "holders of wonderful things"

- Data
- Functionality
- They are bundles of stuff that can be injected into components

Anything to be shared between components goes in a service

Services should be view-agnostic

- They shouldn't know anything about any view
- They can't write to a view

Why?

- a. That's the class's job
- b. Services should do ONE thing
- c. Otherwise they couldn't work with multiple views

How to add a service to your component

Services break the pattern of inclusion

- There's no @Service decorator
- They're usually grouped with other things in a module, but you don't technically add them to a module

You add services in three steps ...

1. Import them

```
import { FooService } from '../shared/foo.service';
```
2. Provide them in the component or module

```
@Component({
  providers: [ FooService ]
}) ...
```
3. Use DI to inject them in the constructor

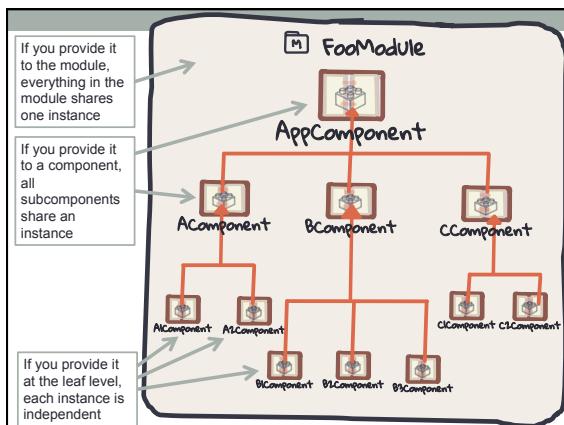
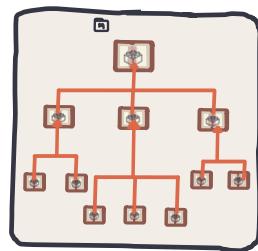
```
constructor(private _fooService: FooService) { ...}
```



Not the only way to do it, but remember this nifty shorthand for injecting into the constructor.

Where you provide them is very important!

- For tight cohesion you want to provide them as low as possible.
- But if you get too low, you have to provide two different times and thus it is two different instances
- If you register high enough, it's a singleton and can share data



How to write a service

Use the angular CLI, of course!

```
ng generate service <serviceName>
- Note:
  - ng generate service doesn't create a subdirectory
  - Don't ng generate fooService, just foo
  - Doesn't affect the module at all (unlike other blueprints).

$ ng generate service foo
installing service
  create src/app/foo.service.spec.ts
  create src/app/foo.service.ts
  WARNING Service is generated but not provided, it must be provided to be used
$ 
```

↑
Reinforcing that we need to inject it in the component or the module before it can be used!

And your service might look like this...

```
import { Injectable } from '@angular/core';
@Injectable()
export class FooService {
  someProperty = {};
  doSomething(inputValue) {
    const outputValue = "foo";
    return outputValue;
  }
}
```

Injectable must be imported

Must be marked as @Injectable

Any member of the class will be exposed as part of the service.

tl;dr

- Services are for sharing data or functionality across components.
- They must be added to the providers array of a module or component annotation and must be injected to be used.
- Provide them as low as possible but not so low they can't be shared
- Create services with ng generate service which will add the all-important @Injectable annotation
