ANGULAR

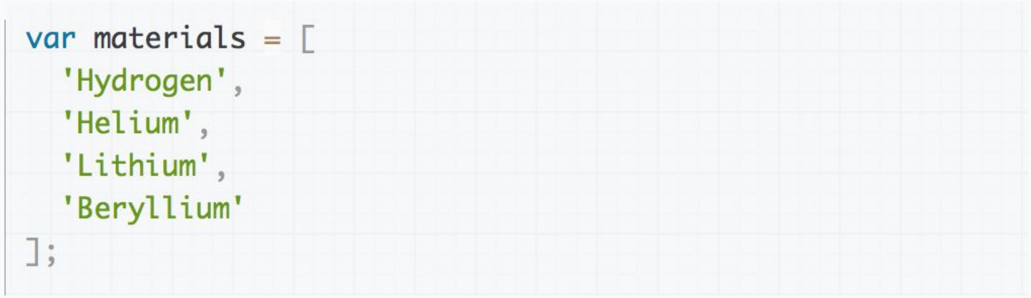
Introduction with code along

Intro

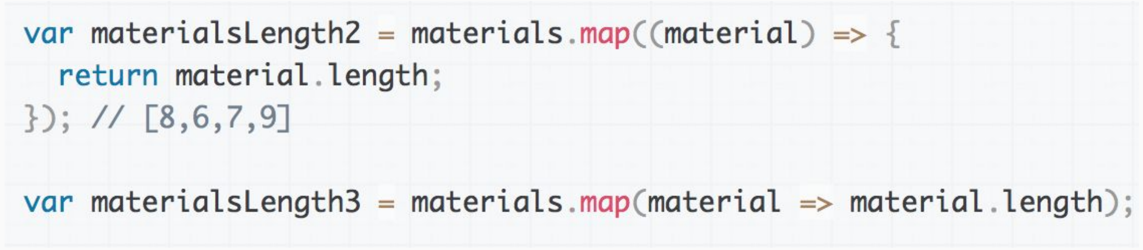
- Intro to ES6 and Typescript
- Angular development ecosystem (VS Code, ALS and NPM)
- Anatomy of an Angular App (Decorators, DI and Components)
- Template, Internal Directives and Binding
- Routing and Server communication
- Build and Release.

ES 6 Features (Ecmascript 2015)

- Block Scopes
- Template Strings
- Classes
- Default Params
- Destructuring
- Arrow Functions
- Modules



```
var materials = [
  'Hydrogen',
  'Helium',
  'Lithium',
  'Beryllium'
];
```



```
var materialsLength2 = materials.map((material) => {
  return material.length;
}); // [8,6,7,9]

var materialsLength3 = materials.map(material => material.length);
```

TypeScript

TypeScript is a free and open-source programming language developed and maintained by Microsoft.

It is a strict superset of JavaScript, and adds optional static typing and class-based object-oriented programming to the language.

Angular is written in Typescript

- Type Inference and Data Types
- Interfaces
- Classes and Access Modifiers
- Static and Instance Members
- Function Overloading
- Generics
- Decorators

Development Ecosystem

Must have: Git, Node and Typescript Compiler

Editors: VS Code, Atom, Sublime, WebStorm, Eclipse etc.

git

npm

ANGULAR CLI

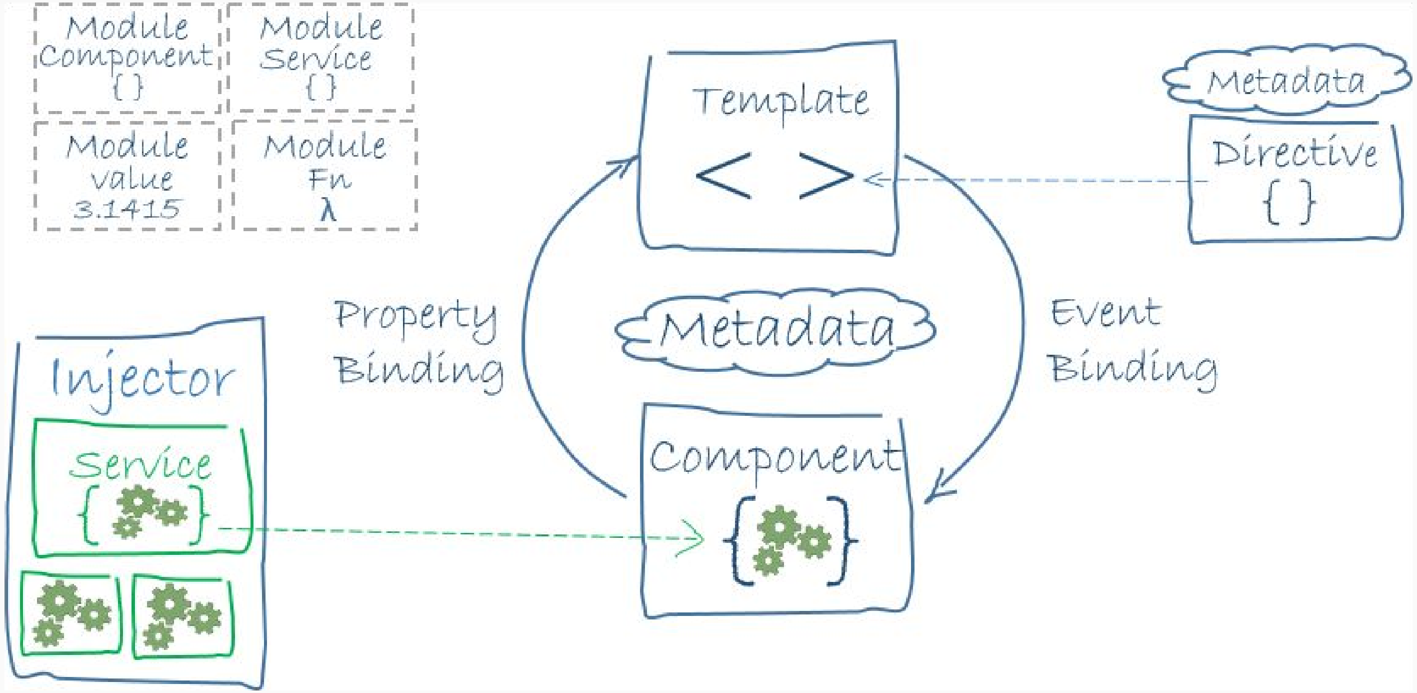
TS

node JS

Angular Language Service

Visual Studio Code Extension

Anatomy of an Angular application



Modules

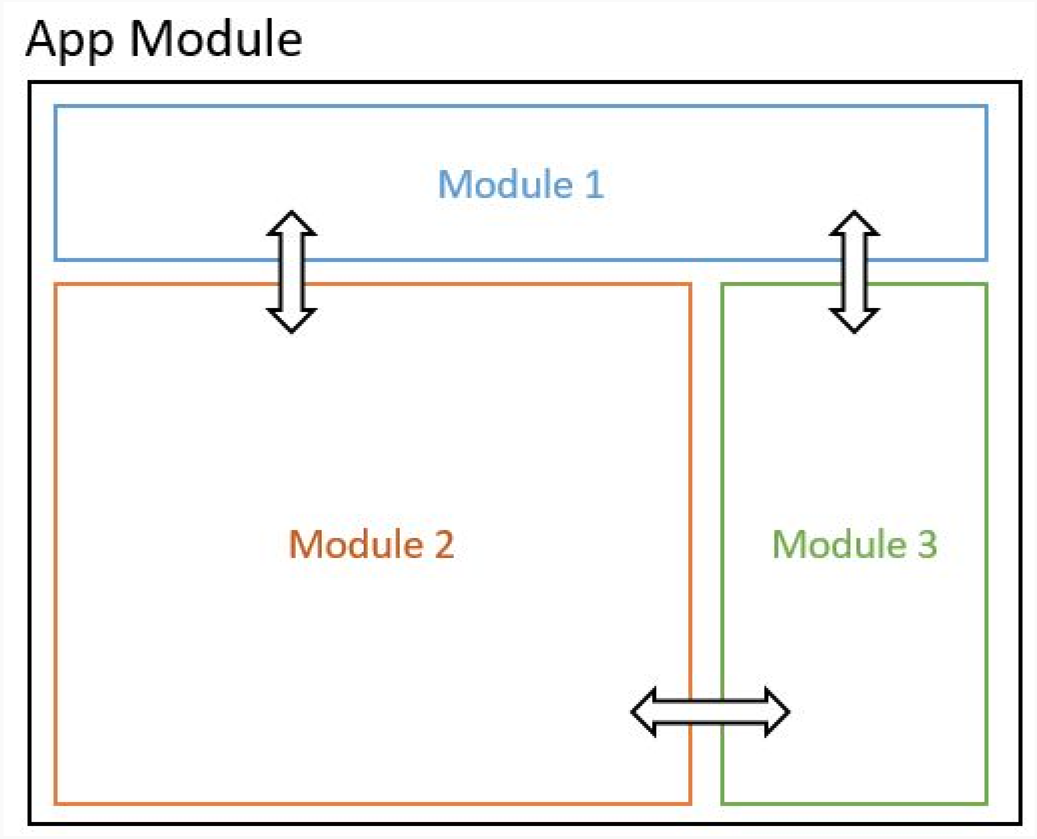
Top most logical container

Provides selective export

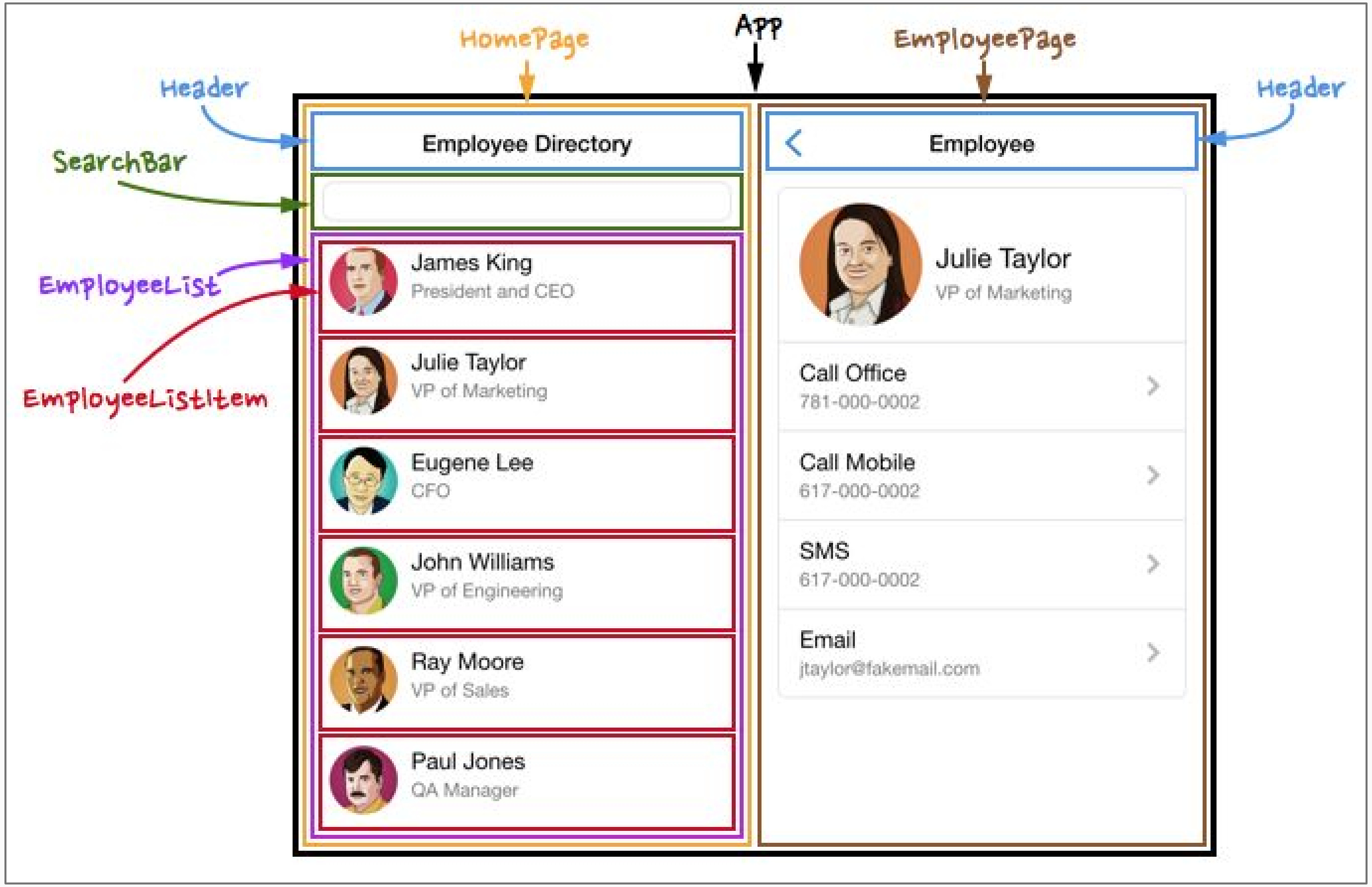
Can be lazy loaded

Contains:

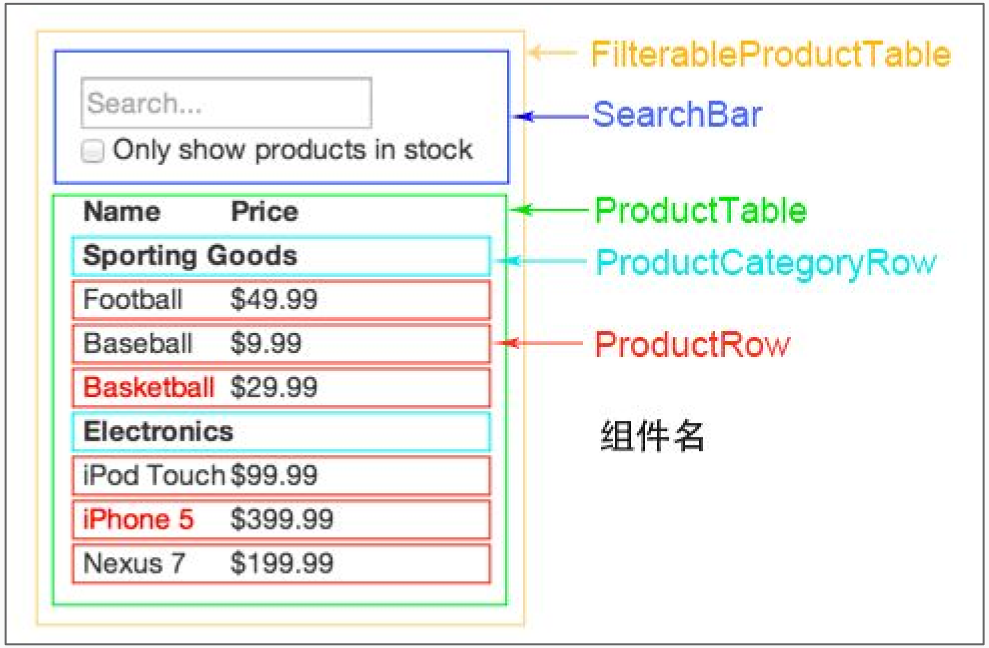
- Components
- Services
- Routes
- Pipes
- Directives etc

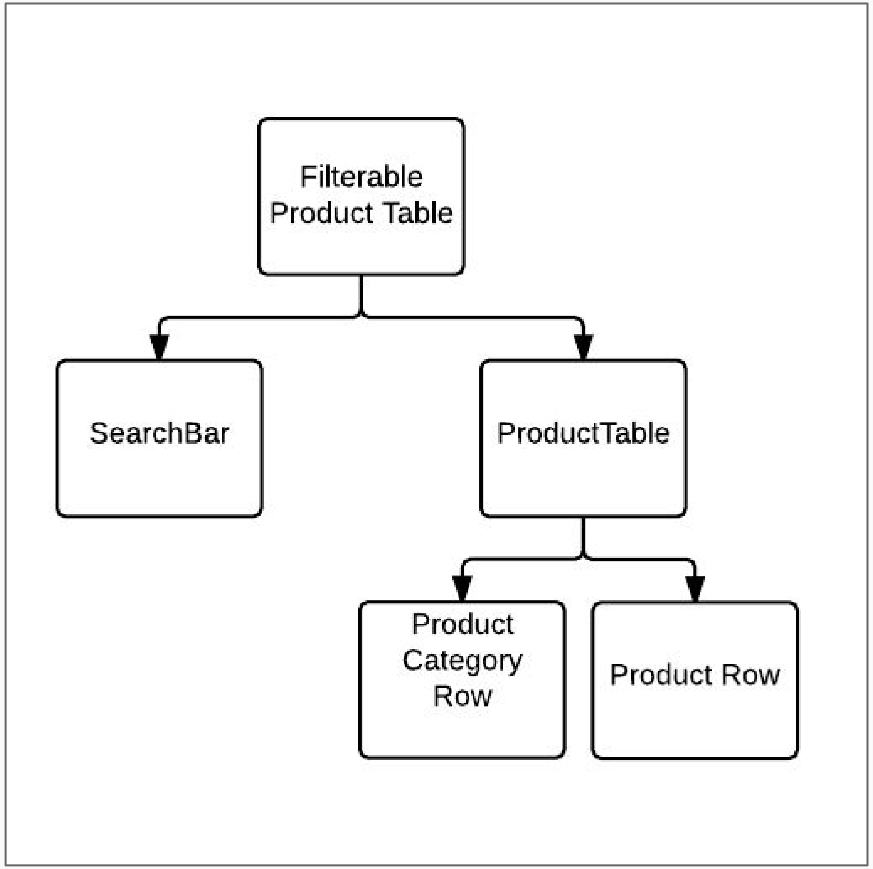


Sample Contacts App

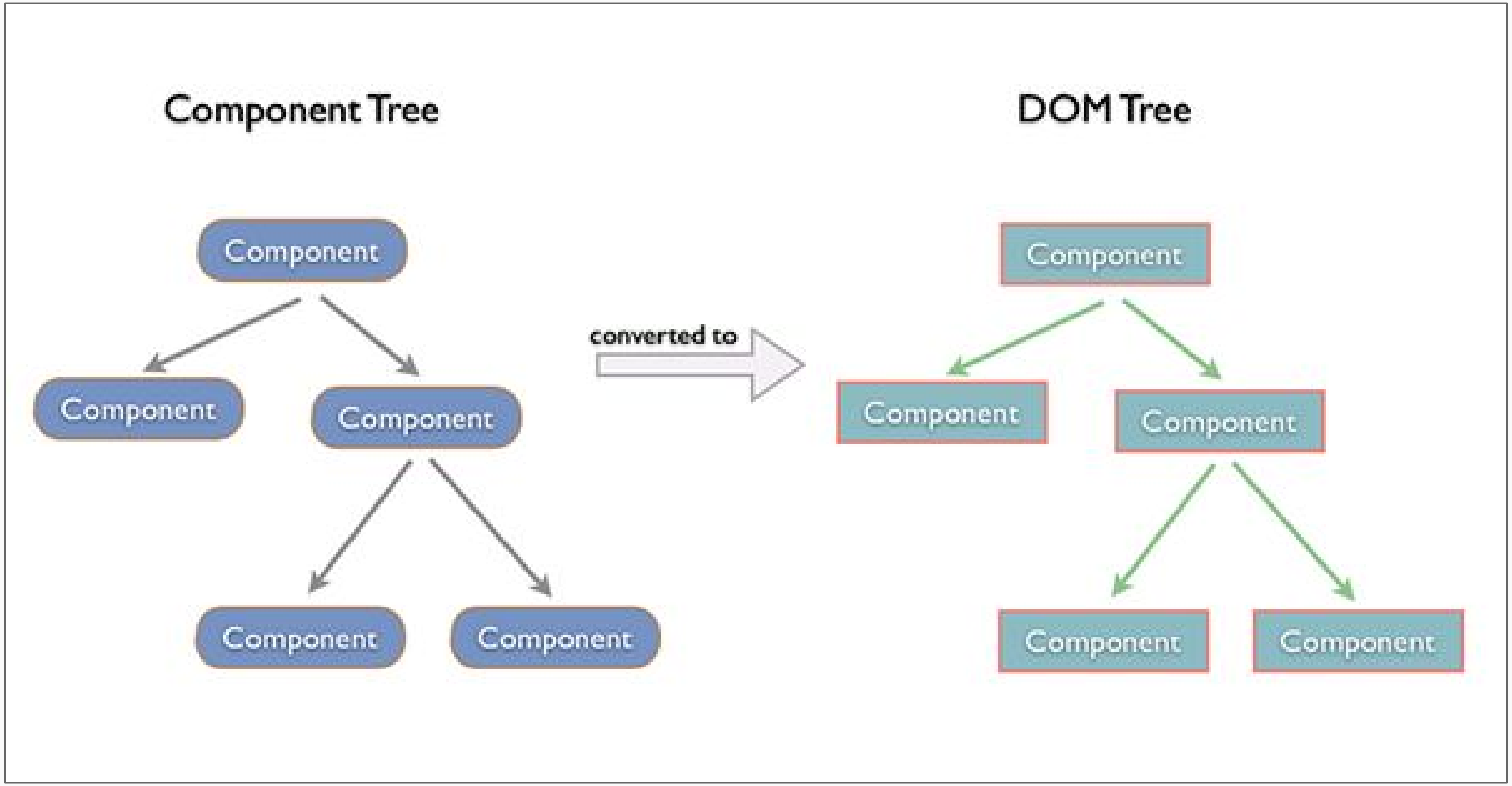


UI Component Hierarchy





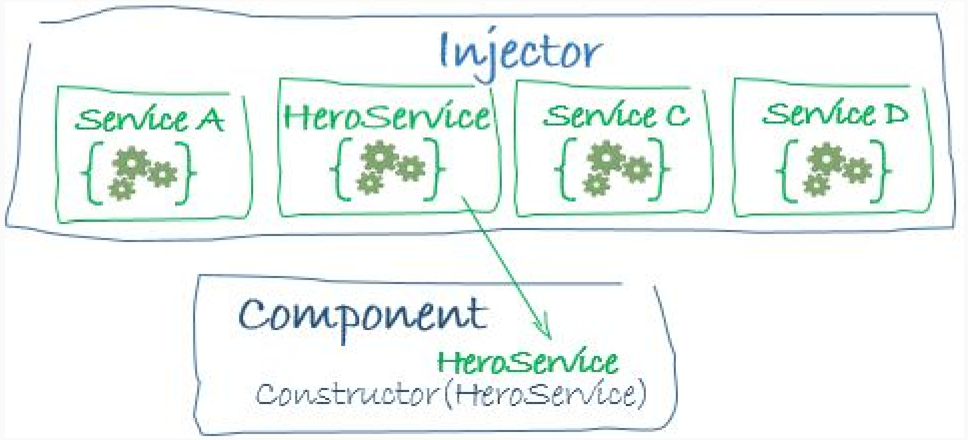
Eventual DOM Tree



Decorators & Dependency Injection

```
import { NgModule }      from '@angular/core';
import { BrowserModule } from
'@angular/platform-browser';
@NgModule({
  imports:      [ BrowserModule ],
  providers:    [ Logger ],
  declarations: [ AppComponent ],
  exports:      [ AppComponent ],
  bootstrap:    [ AppComponent ]
})
export class AppModule { }
```

```
constructor(private service: HeroService) { }
```

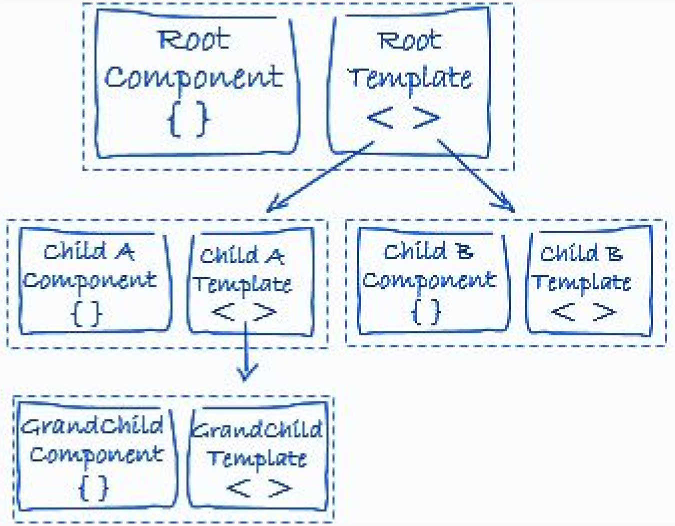


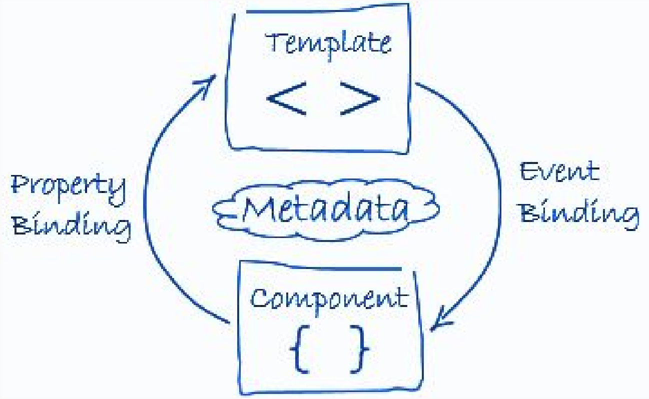
```
@Component({
  selector: 'app-root',
  templateUrl:'./app.component.html',
  styleUrls: ['./app.component.sass']
})
```

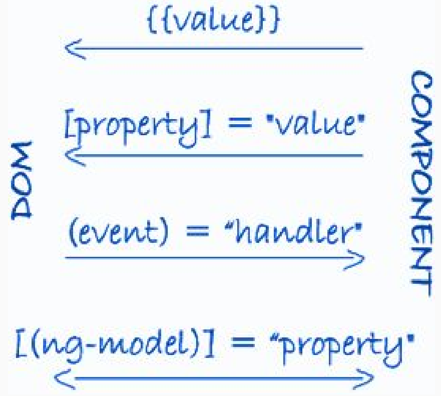
Components,

Templates &

Binding



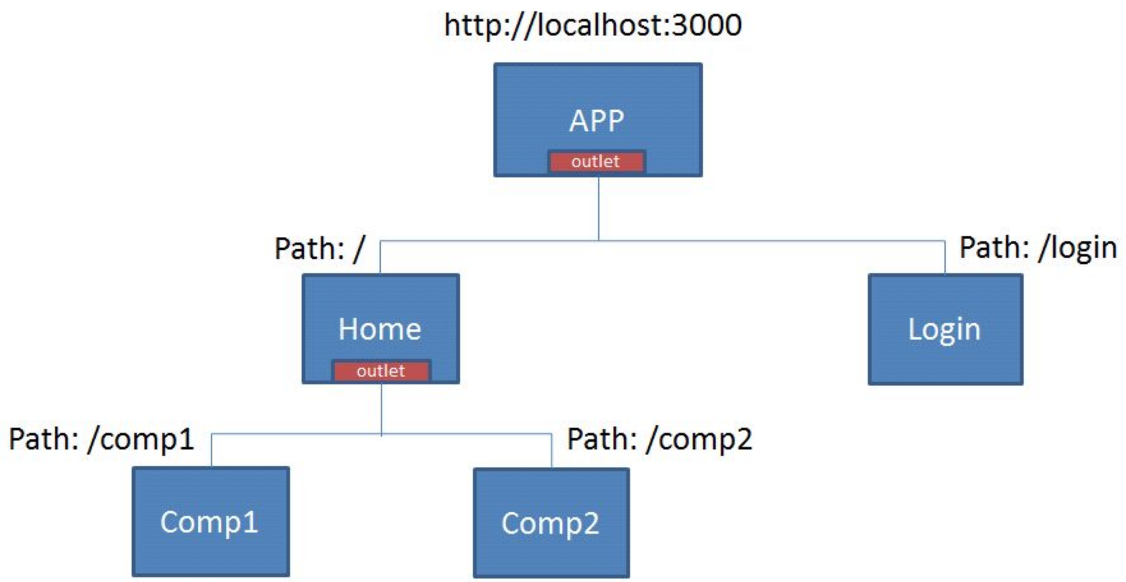




Routing

The Angular Router enables navigation from one view to the next as users perform application tasks.

It can interpret a browser URL as an instruction to navigate to a client-generated view.

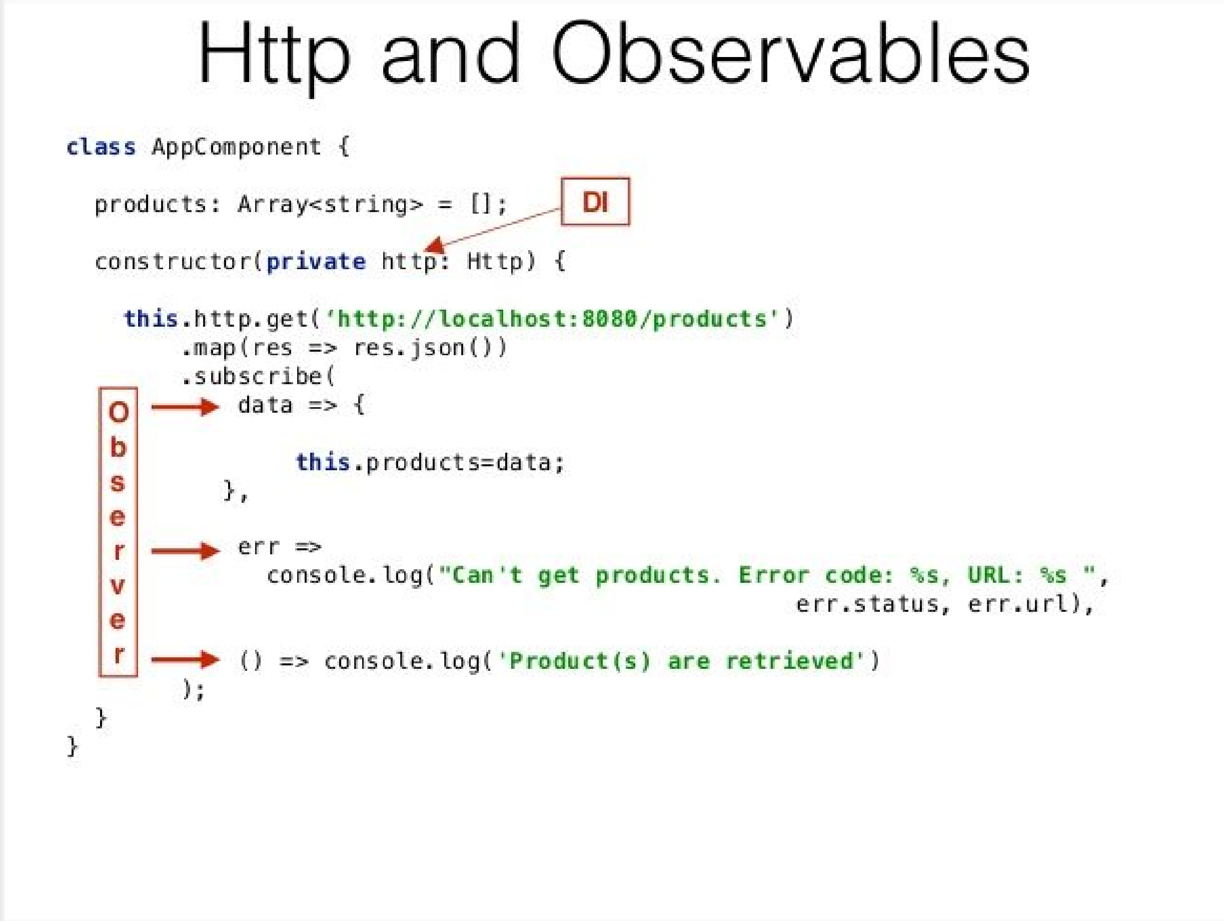


Server Communication

Performs http requests using "***XMLHttpRequest***" as default backend

Http is available as an ***Injectable*** class

Calling a request returns an ***Observable*** which will emit a single Response when a response is received.



Http and Observables

```
class AppComponent {

  products: Array<string> = [];

  constructor(private http: Http) {

    this.http.get('http://localhost:8080/products')
        .map(res => res.json())
        .subscribe(
             data => {

                 this.products=data;
             },

             err =>
               console.log("Can't get products. Error code: %s, URL: %s ",
                                                     err.status, err.url),

             () => console.log('Product(s) are retrieved')
        );
  }
}
```

Build and Release

- Ahead-of-Time (AOT) Compilation: pre-compiles Angular component templates.
- Bundling: concatenates modules into a single file (bundle).
- Inlining: pulls template html and css into the components.
- Minification: removes excess whitespace, comments, and optional tokens.
- Uglification: rewrites code to use short, cryptic variable and function names.
- Dead code elimination: removes unreferenced modules and unused code.
- Pruned libraries: drop unused libraries and pare others down to the features you need.
- Performance measurement: focus on optimizations that make a measurable difference.

Thank you.