LEARNTEK

Angular

WHY TO USE ANGULAR?

- ANGULAR IS A JAVASCRIPT FRAMEWORK WHICH ALLOWS YOU TO BUILD REACTIVE SINGLE PAGE APPLICATIONS(SPAS)

FEATURES OF ANGULAR

- Consistency
- Maintainability
- Code Reusability
- Catch Errors Early
- Command Line Interface
- Productivity

ANGULAR ENVIRONMENT SETUP

- **Software Requirements for Angular Environment Setup:**

1. IDE (for writing code)
2. Node.js (Node.js is an open-source cross-platform JavaScript run-time environment)
3. NPM (NPM is node.js package manager for JavaScript programming language. It is automatically installed when we install node.js)
4. Typescript (It is the programming language)
5. Angular CLI (It is a tool that allows us to create a project, build and run the project in the development server directly using the command line command)

ANGULAR CLI

- npm install -g @angular/cli
- ng new my-first-project
- cd my-first-project
- ng serve

ANGULAR COMPONENTS AND BINDING

- Creating Angular Components
- Nested Angular Components
- Different Bindings in Angular

a. Interpolation Binding → {{ }}

b. Property Binding → [ ]

c. Event Binding → ( )

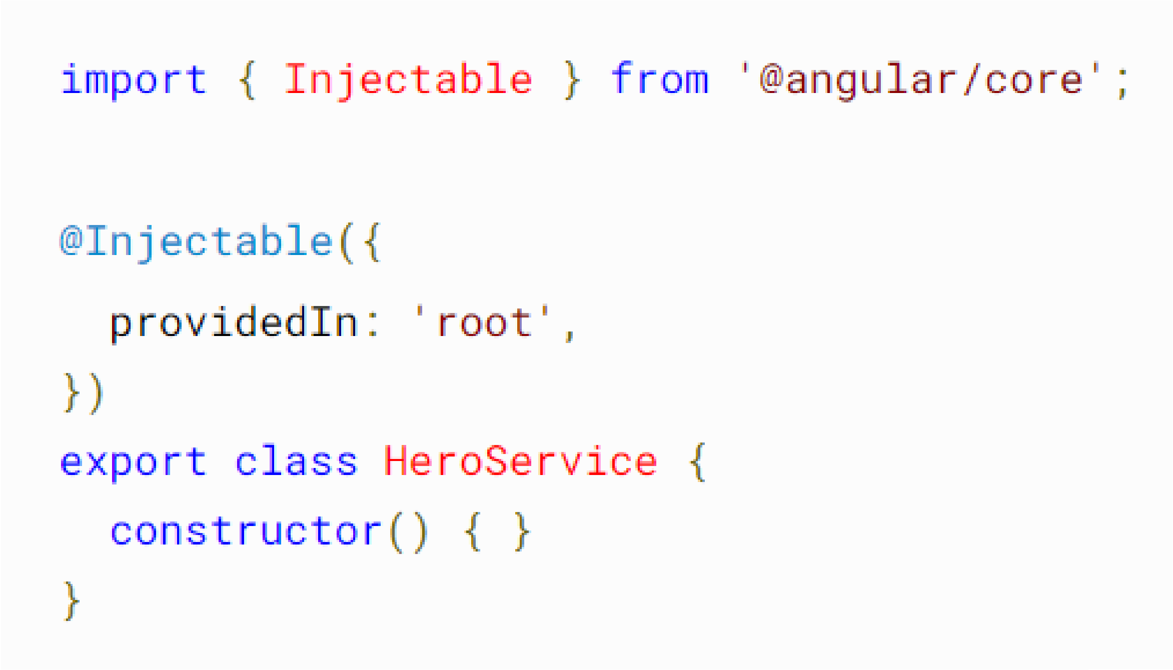
Accessing HTML DOM Elements

- Using Local References in templates
- @ViewChild('Any Variable in TS' , {static : true}) usrVariableName : ElementRef
  - Declare a local template variable in HTML file
- Ng-content

  - select="selector"

    Only select elements from the projected content that match the given CSS selector.

Services and Dependency Injection in Angular



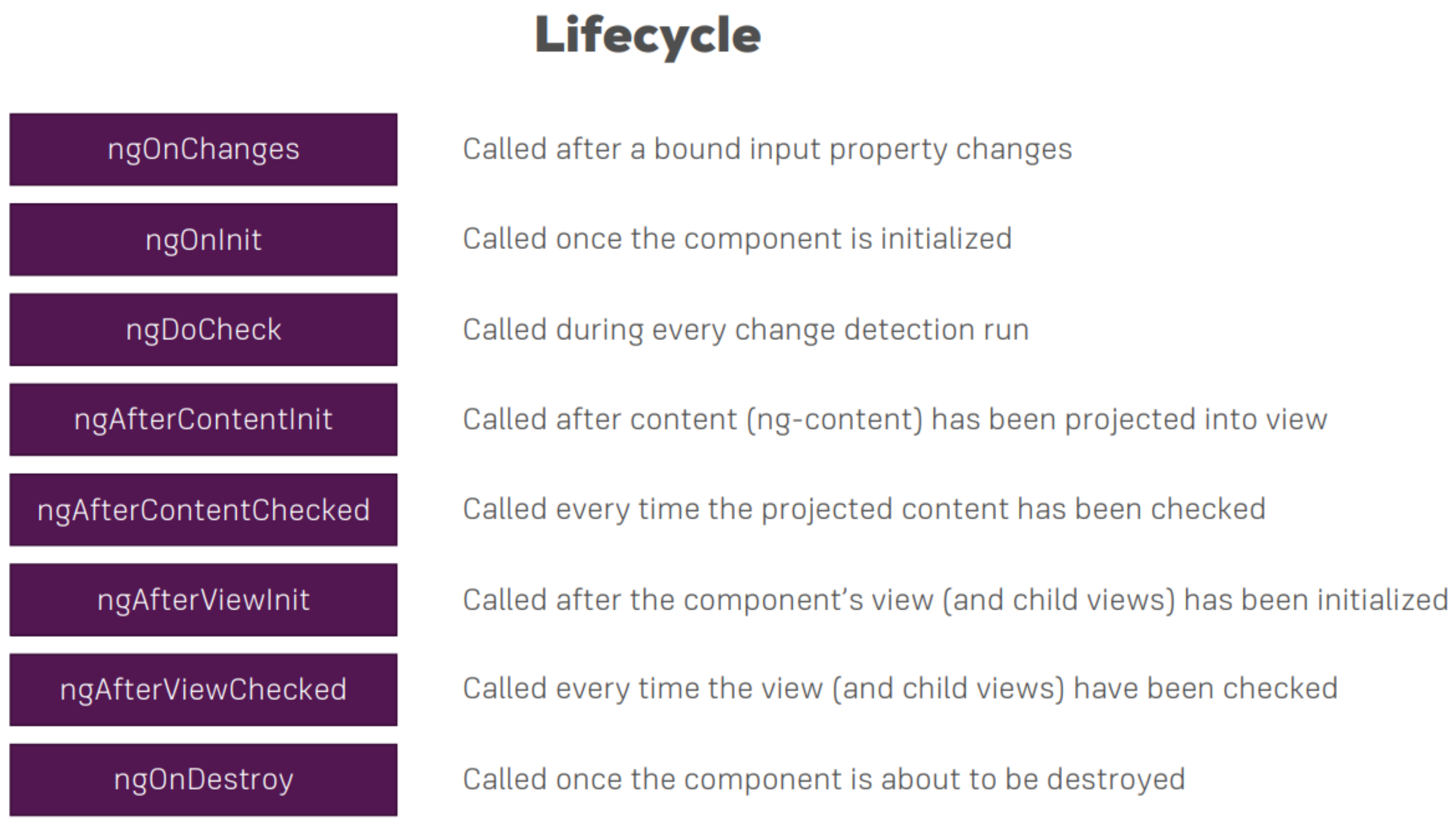
```
import { Injectable } from '@angular/core';


@Injectable({
  providedIn: 'root',
})
export class HeroService {
  constructor() { }
}
```

The @Injectable() decorator specifies that Angular can use this class in the DI system.

The metadata, providedIn: 'root', means that the HeroService is visible throughout the application.

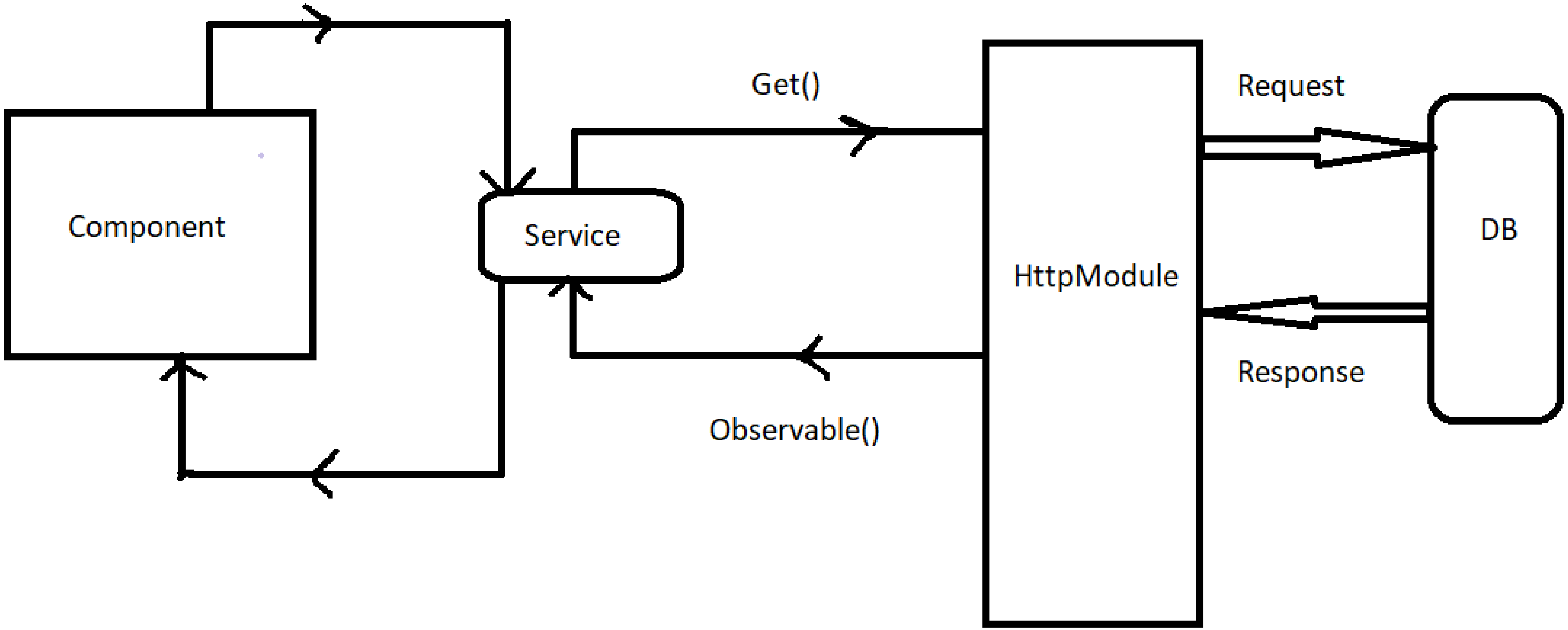
Angular Life Cycle Hooks

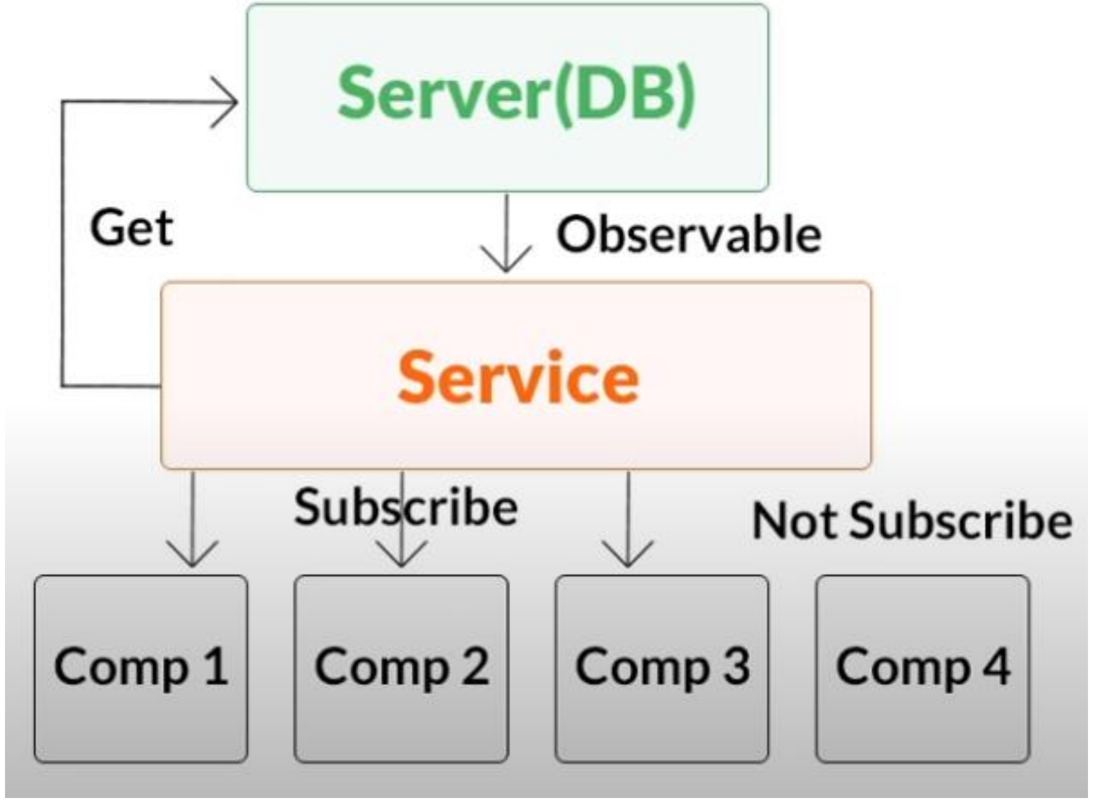


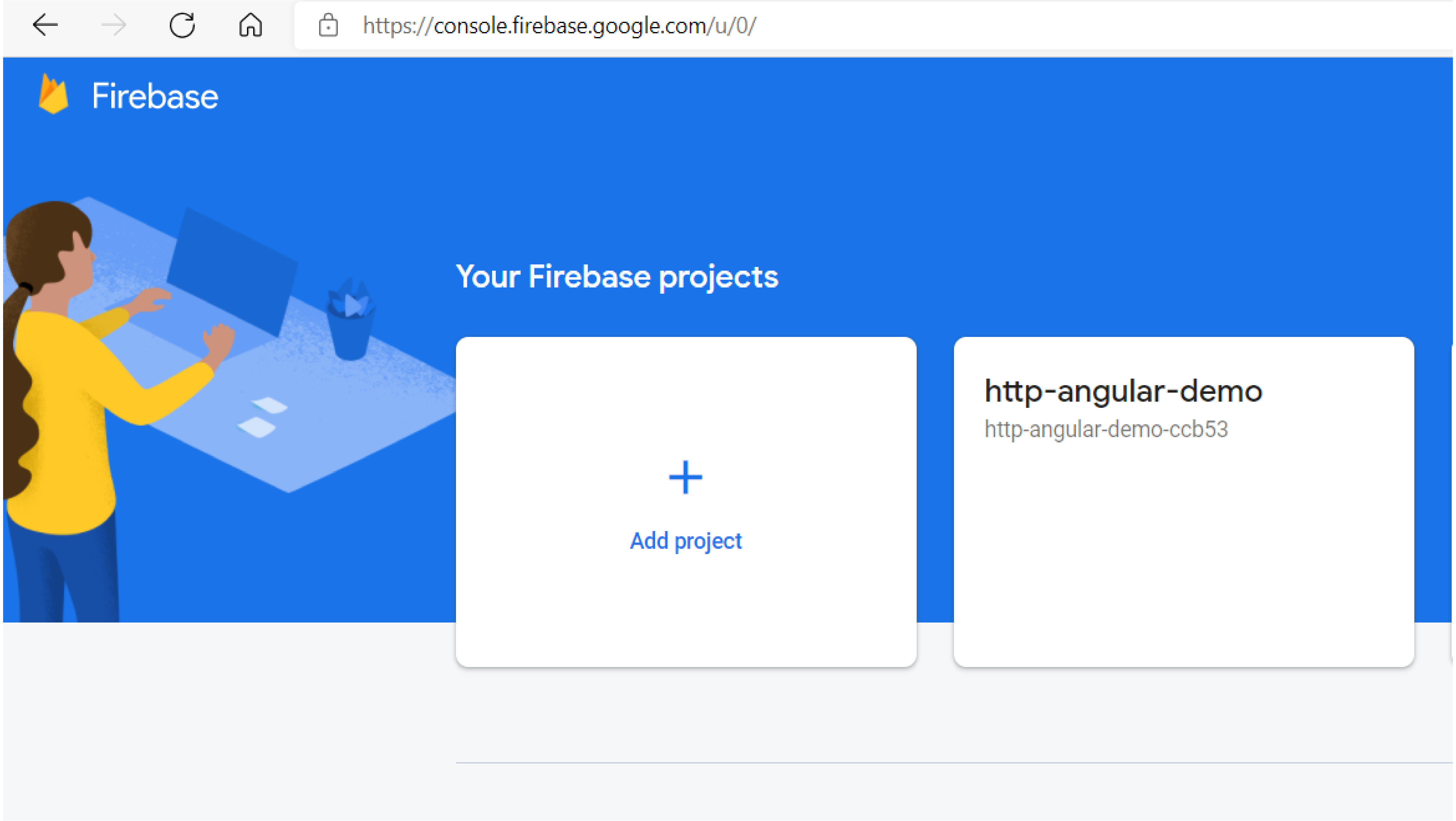
Lifecycle

ngOnChanges	Called after a bound input property changes
ngOnInit	Called once the component is initialized
ngDoCheck	Called during every change detection run
ngAfterContentInit	Called after content (ng-content) has been projected into view
ngAfterContentChecked	Called every time the projected content has been checked
ngAfterViewInit	Called after the component’s view (and child views) has been initialized
ngAfterViewChecked	Called every time the view (and child views) have been checked
ngOnDestroy	Called once the component is about to be destroyed

HTTP & Observables




Server(DB)
Get
Observable
Service
Subscribe
Not Subscribe
Comp 1
Comp 2
Comp 3
Comp 4



https://console.firebase.google.com/u/0/

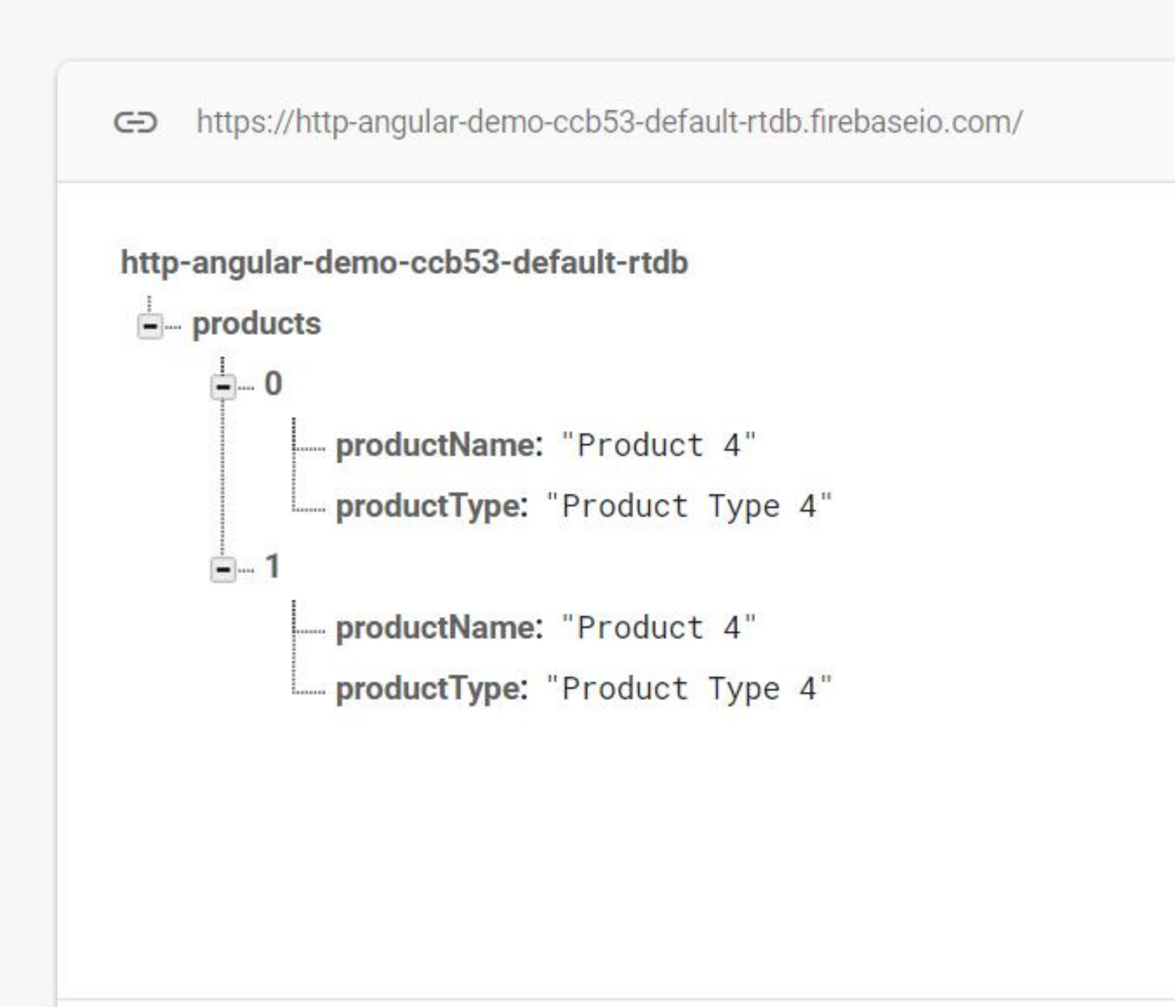
Your Firebase projects

http-angular-demo

http-angular-demo-ccb53

HTTP Requests

- POST/PUT
- GET with async pipe
- DELETE



https://http-angular-demo-ccb53-default-rtdb.firebaseio.com/

```
http-angular-demo-ccb53-default-rtdb
  products
    0
      productName: "Product 4"
      productType: "Product Type 4"
    1
      productName: "Product 4"
      productType: "Product Type 4"
```