

JavaScript

- ES6 / ES2015 - ECMA Specification for JavaScript development
- latest features,
- filter, reduce, map, forEach
- arrow functions
- block scope - let / const
- promises
- classes
- rest / spread operators
- destructuring
- object literals
- Template literals
- Modules - import / exports
- for ... of
- async ... await
- for ... in

- Browser Environment - Vanilla JavaScript, React, Angular etc
- Server Environment - NodeJS -
 - : Node Runtime Environment
 - : Node Package Manager (npm) - npm install
 - : Node Native Modules

TypeScript

- Superset of JavaScript
 - : Primitive : Number, Boolean, String
 - : Reference : Objects & Arrays
 - : Functions :
 - : Additional Types : null, undefined, void, unknown, enums, any, tuple,
 - : Complex Types : Class, Interface
 - : type
- Compiled to JavaScript
- Type definitions
- OOPS concepts
- Type related errors at compile time
- Code Optimization
- Typescript Compiler - typescript - tsc

Single Page App-

- browser will not refresh
- the page content will change based on URL
- loads all file in one go
- part of view changed without reloading the page
- App Performance
- Less Network bandwidth usages
- Great user experience
- Server load decreases

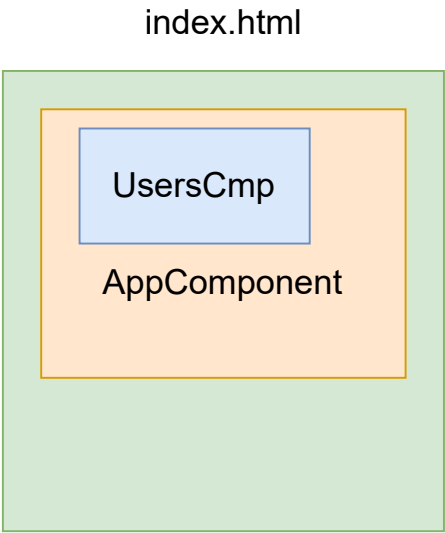
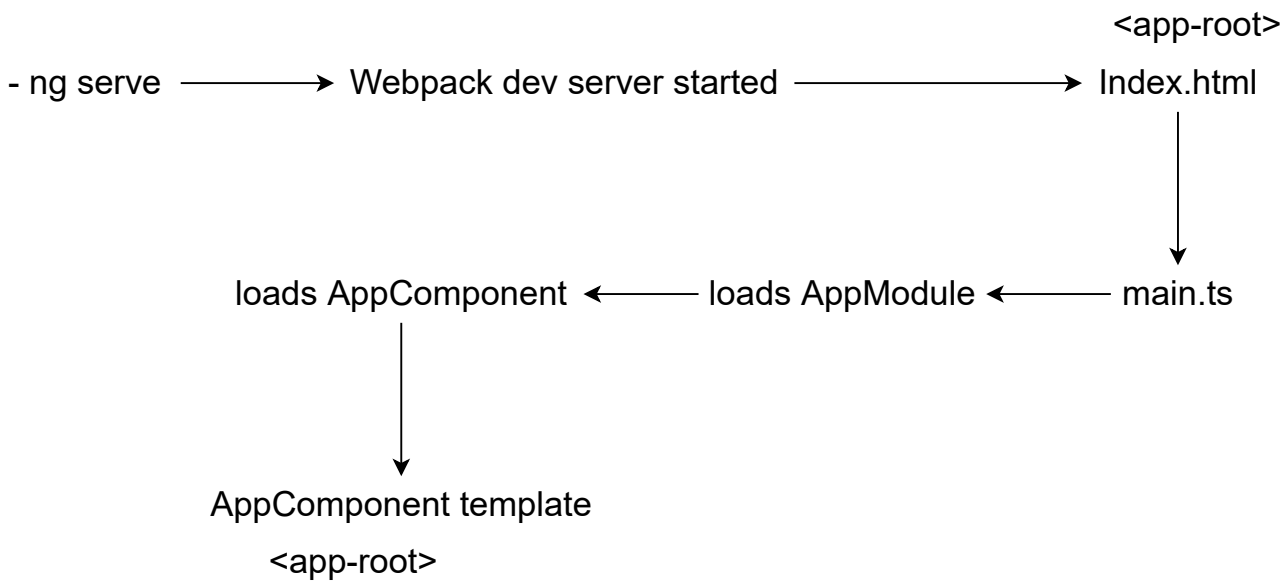
AJAX | XHR | fetch() : Promise API

- jQuery.ajax()
- document.getElementById()
- jQuery()

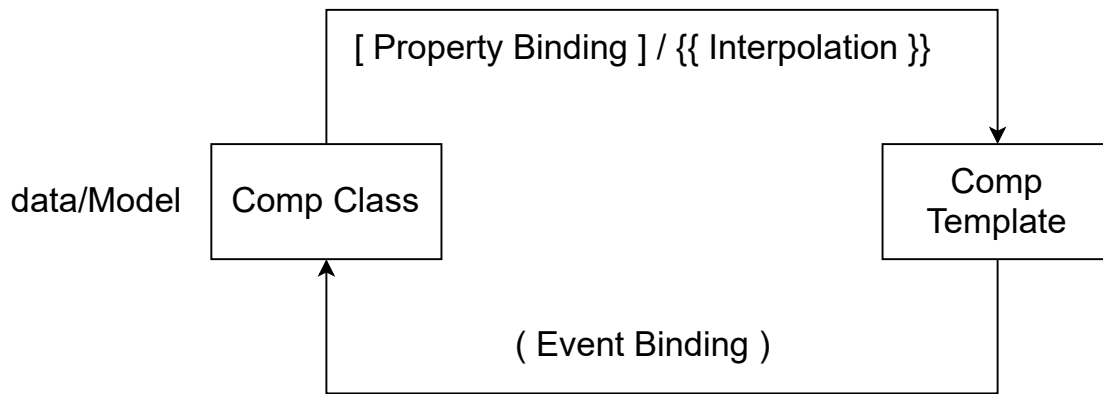
Client-side JavaScript Lib/Framework

- jQuery : DOM Manipulation, AJAX, Animations
- AngularJS : v1.x; MV* at client side, DI, Two-way Databinding, templates, module approach, small in size, SPA,
- ***Angular : Super-heroic framework**
 - : **Component based (reusability)**
 - : **SPA**
 - : **Mobile friendly : + Ionic | + bootstrap | + angular material comp**
 - : **CLI Support :**
 - : **Performance** - less time to load on browser; Lazy loading; AoT : Reduces bundle size; 'Tree-shaking' - removes irrelevant code from app bundle; 40%, IVY, ngcc, app optimization techniques.
 - : **Security : Authentication - RouteLink, XSS, CSRF / XSRF HttpClient Module, HTTP Level vulnerabilities**
 - : **Routing : Route guards, Lazy loading, nested routing, parameters, query params etc**
 - : **DI - Design Pattern; Service; loosely coupled code**
 - : **2 way data binding - [(ngModel)] - Banana in the box**
 - : **Use of template**
 - : **Lazy loading : Loads the feature on demand**
 - : **Remote Server Calls : HttpClient - get, post, patch, delete**
 - : **Ease of Testing, maintaining, scalability - MVC at client-side, TestBed; jasmine, karma, protector etc**
 - : **Shadow DOM - Web-components (HTML5 API) - small part of actual DOM; encapsulation**
 - : **DOM Manipulation : Directives, Renderer, ElementRef, ViewChild, ContentChild etc**
 - : **State Management : Services are singleton**
 - : **Modular Approach : Module - Container for inter-related code; encapsulation**
 - : **Change Detection - Zone.JS - Digest Cycle - to detect changes on DOM**
 - : **RxJS - Observables and operators - creational, filtering, combined, utility etc ; Async calls**
 - : **and many more**
- ***Ember** : frequent change in internal API's
- React : Virtual DOM, Diffing Algo, Cross platform support, Component based, Dynamic Data handling, small bundle size, Declarative way of writing code, Faster, scalable, simple to learn, Facebook Team; **react-router, redux, flux**
- Preact - 3Kb, based on React concepts
- Vue : Evan You 2 way data binding Virtual DOM vuex vue-router Template syntax small

Angular App Bootstrap Process



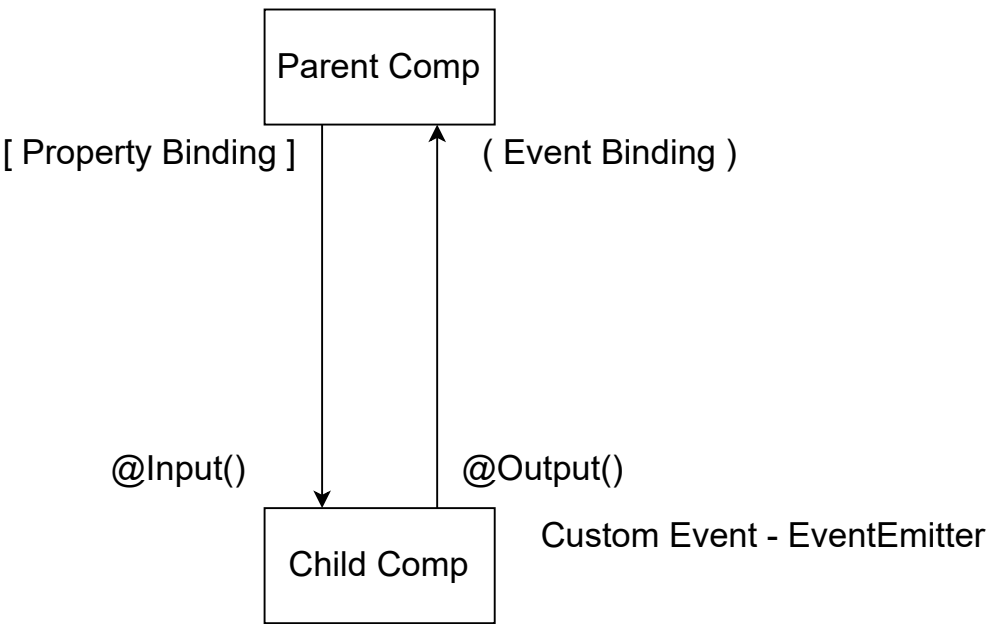
Data Binding



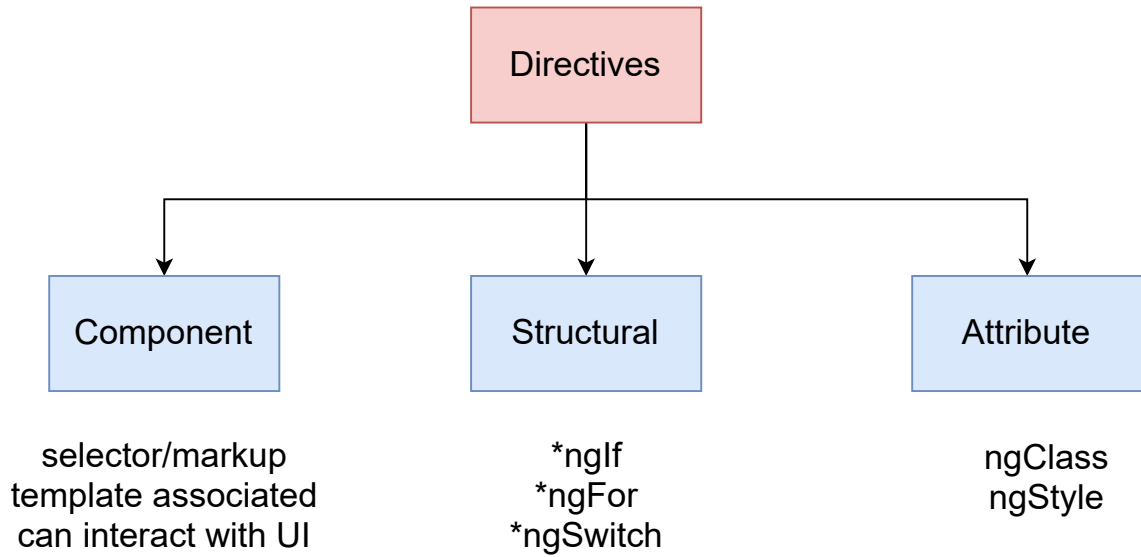
2 way data binding

- [(ngModel)] : Banana in the box
- [PB] + (EB)

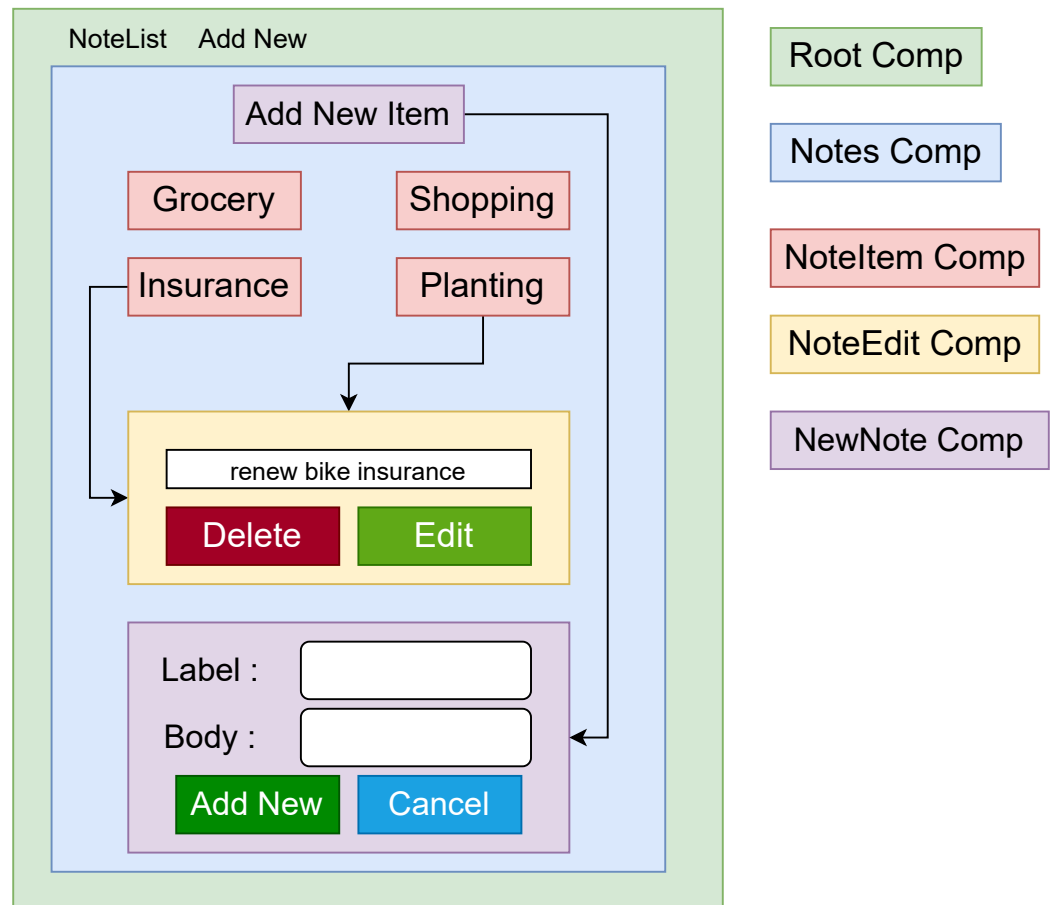
Component Communication using @Input & @Output



Directives : Introduction



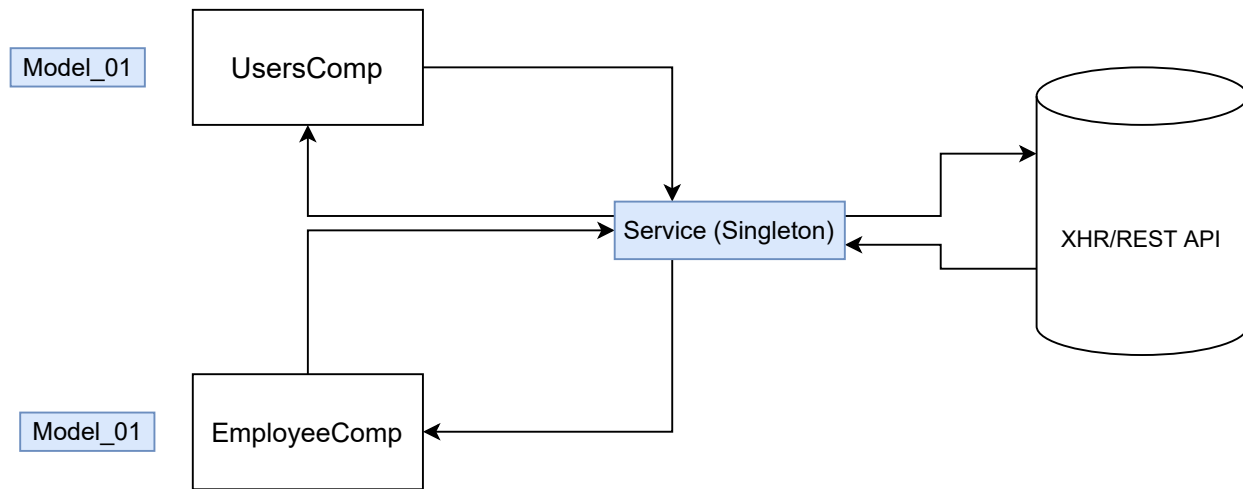
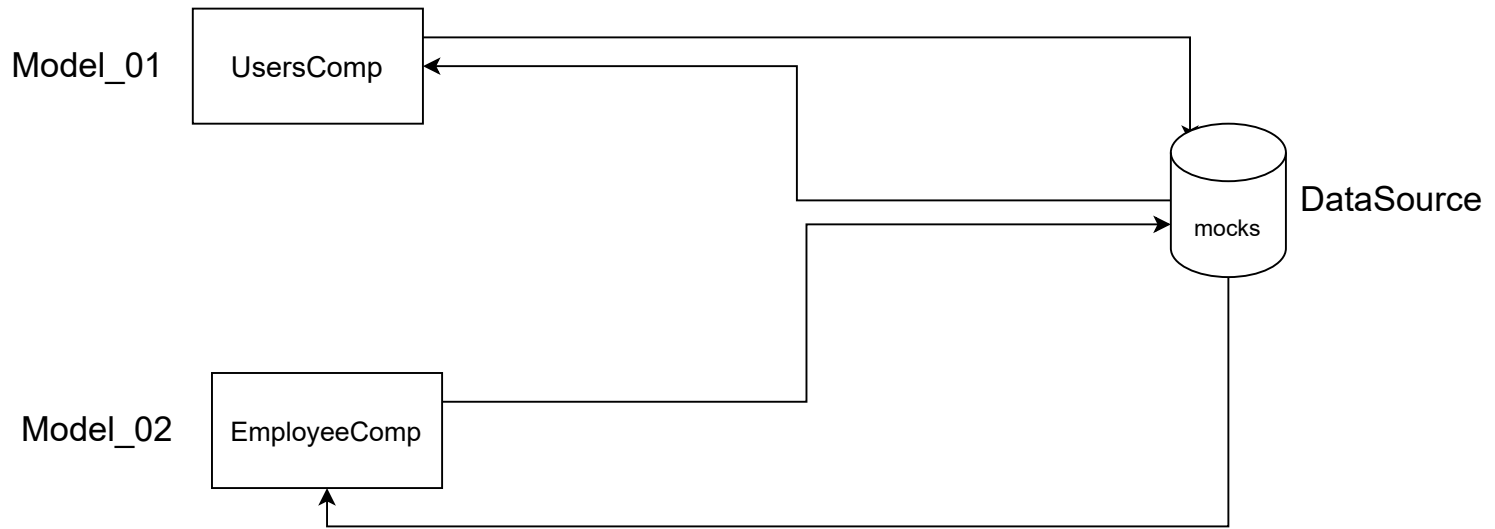
Notes App



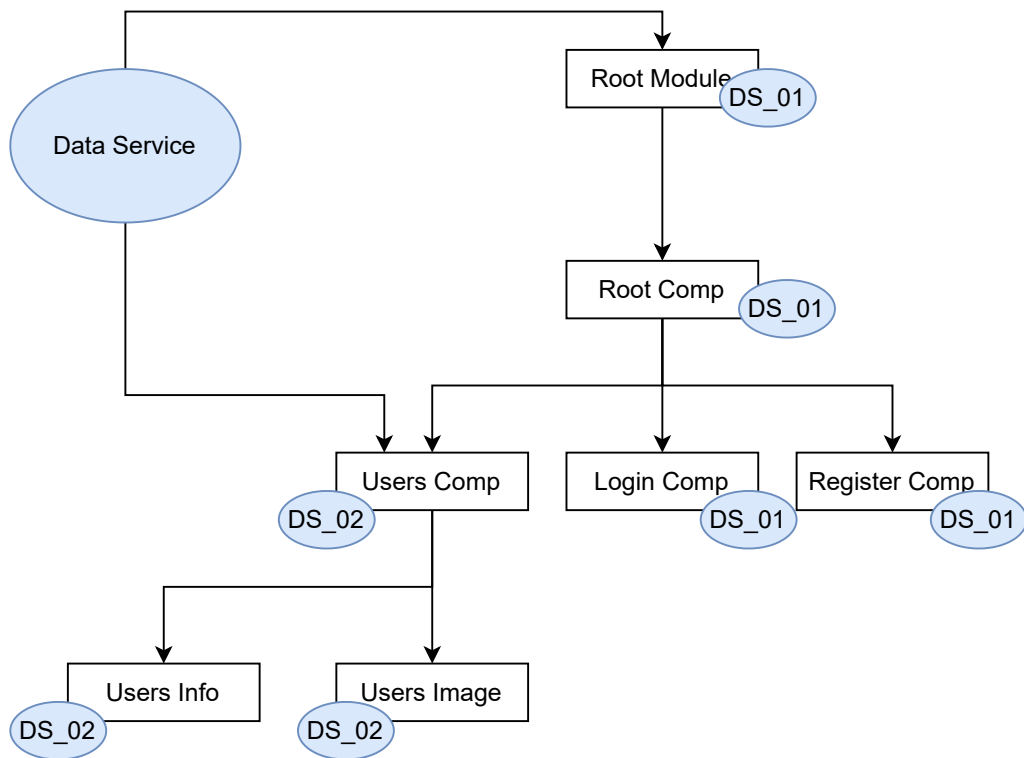
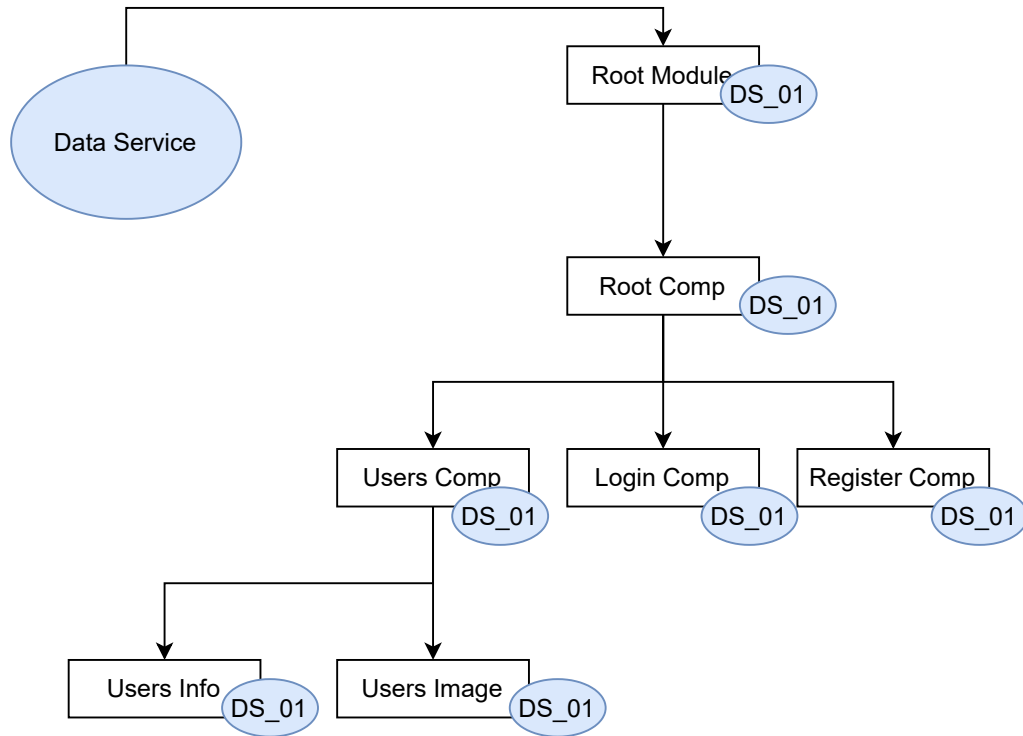
Model Details -

```
let notes = [  
  {id : 1, title : "grocery", body: "buy the pulses"},  
  {id : 2, title : "shopping", body: "buy the jeans"},  
  {id : 3, title : "insurance", body: "renew car insurance"},  
  {id : 4, title : "planting", body: "pot the plants"},  
]
```

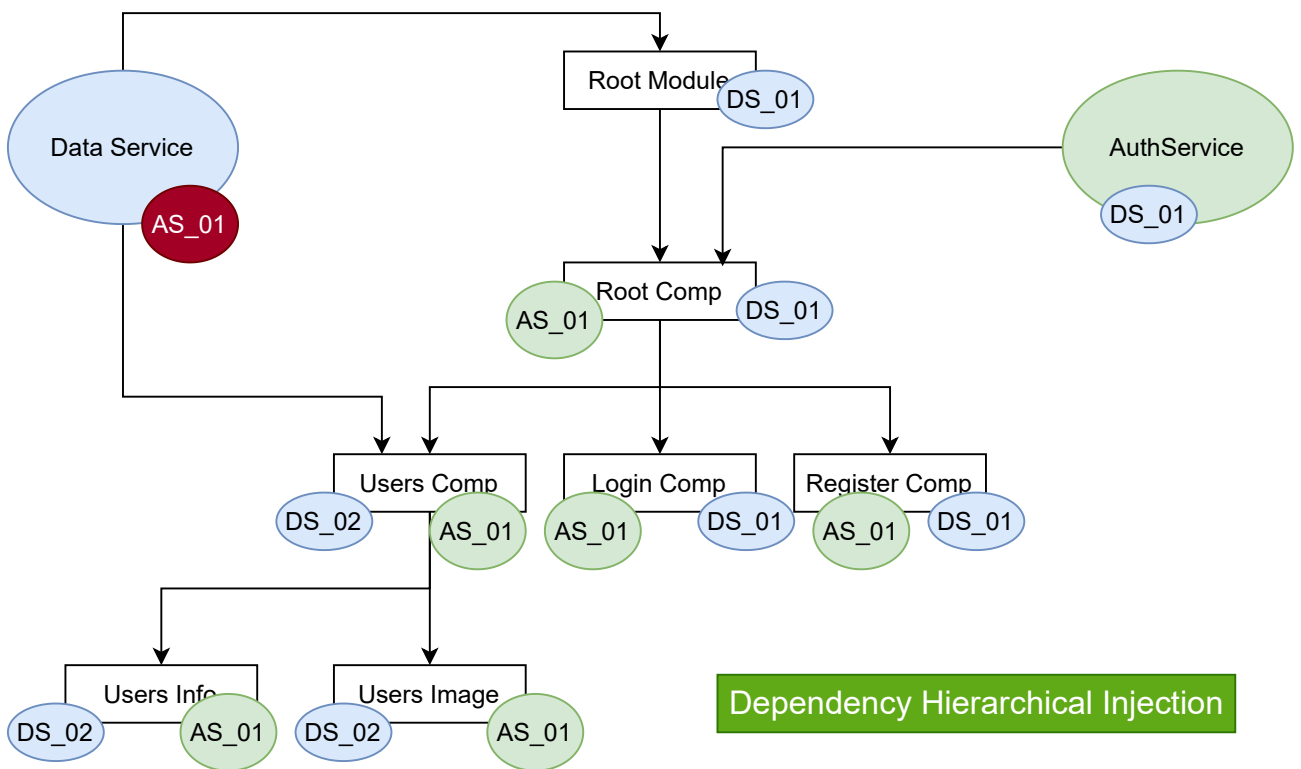
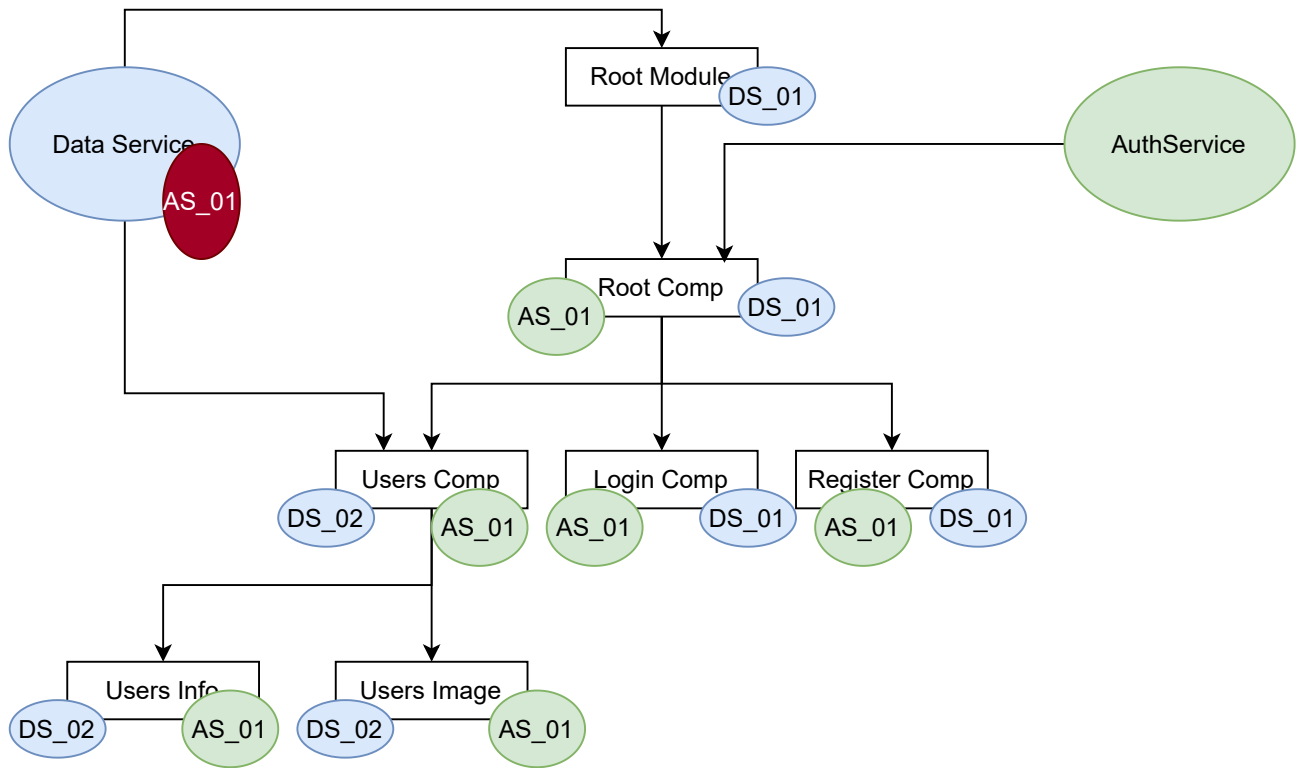
Multiple Model Problem



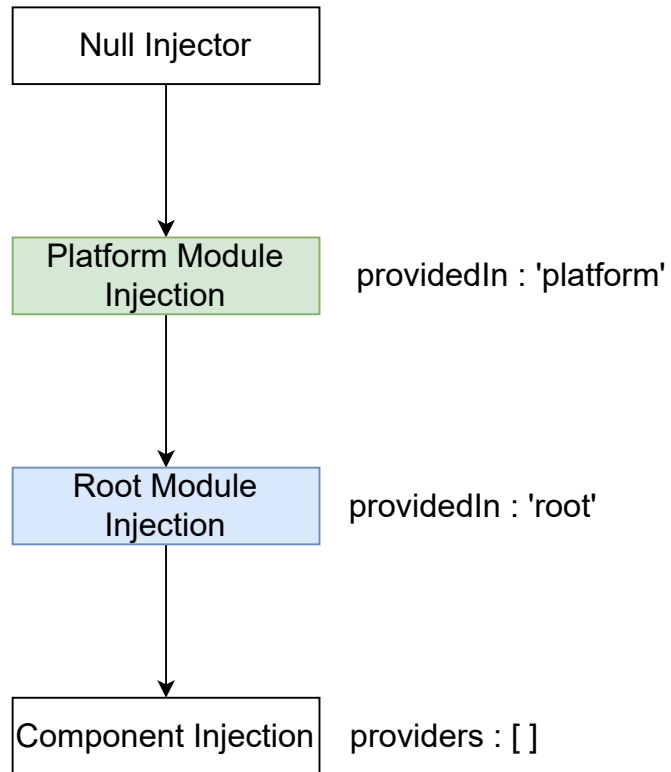
Services DI Scenarios



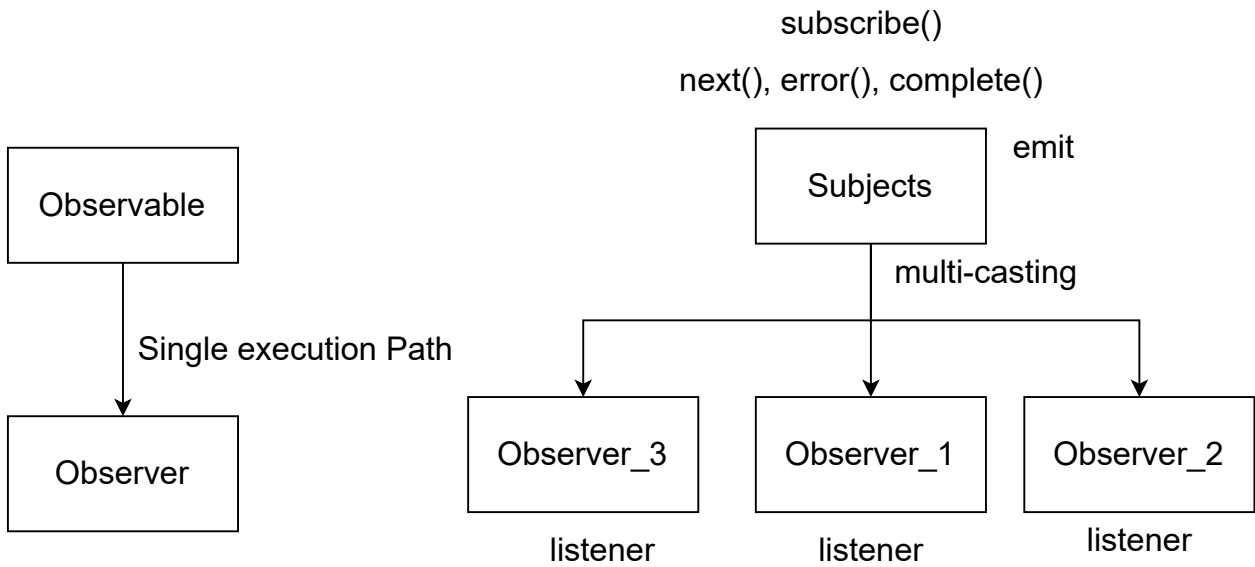
Services DI Scenarios



ngV6 - Tree-shakable services

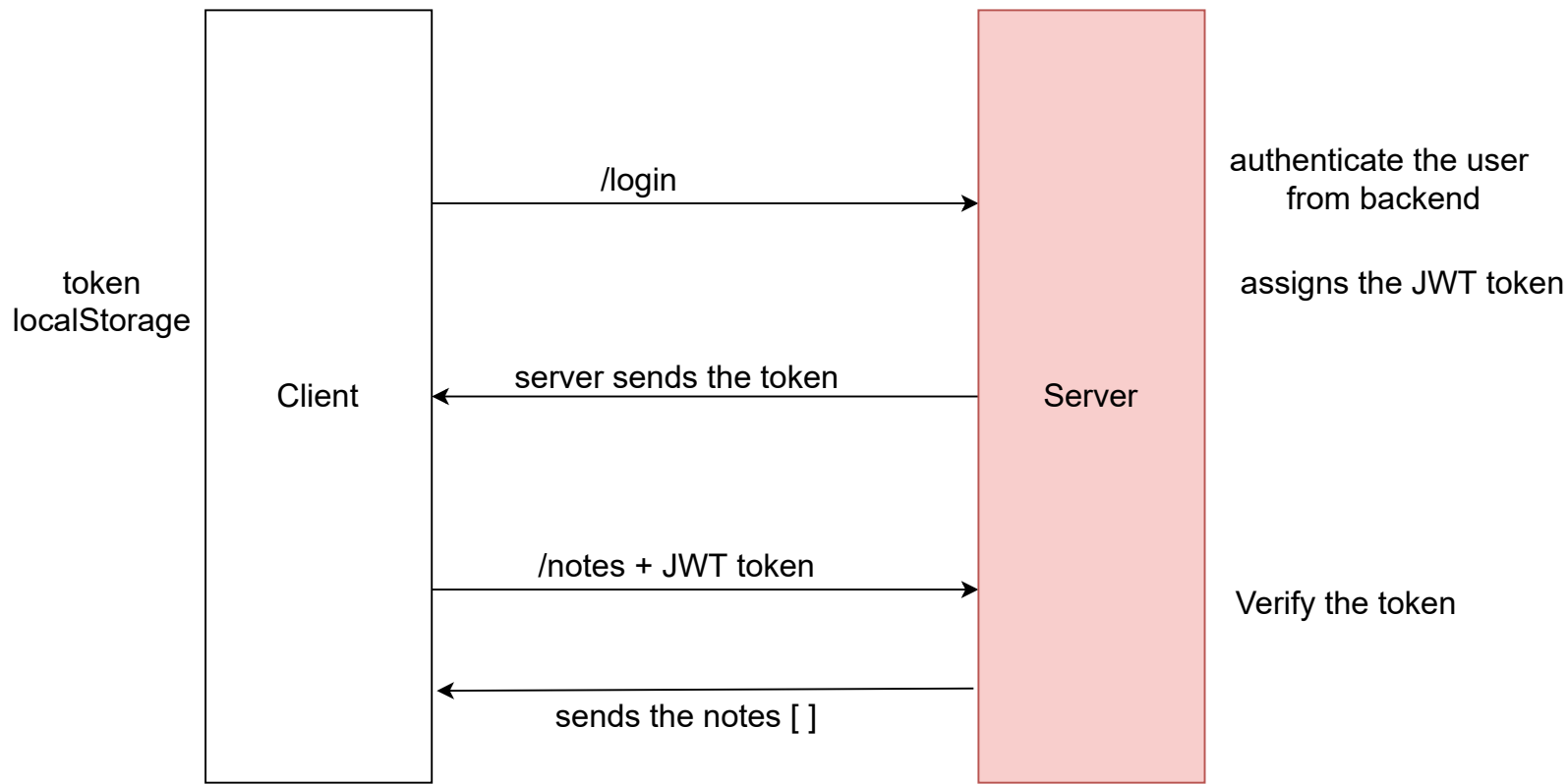


Observables

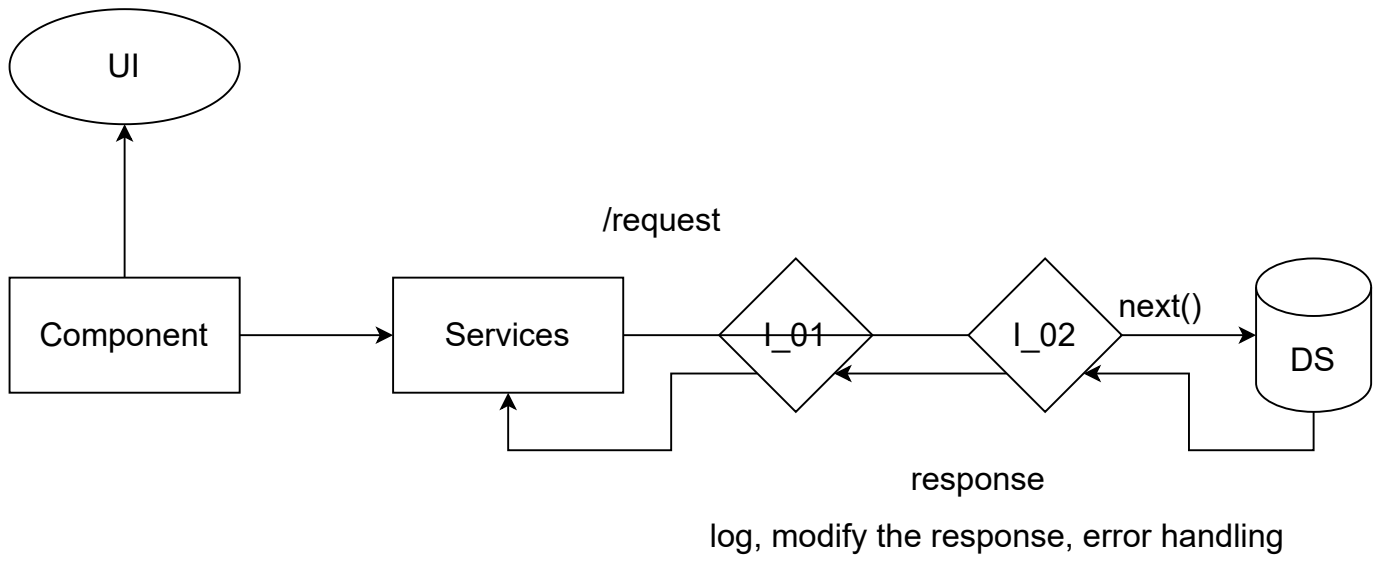


Note : Check the observable reference to know more about observables

Understanding JWT



Understanding Interceptors



Understanding Routing in Angular

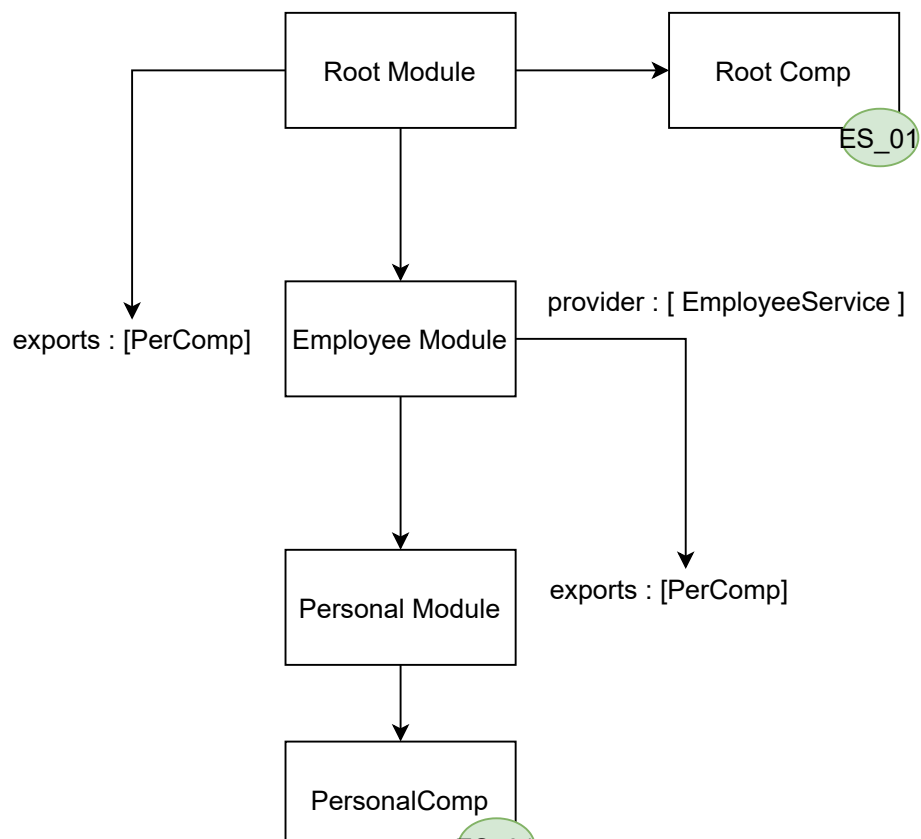
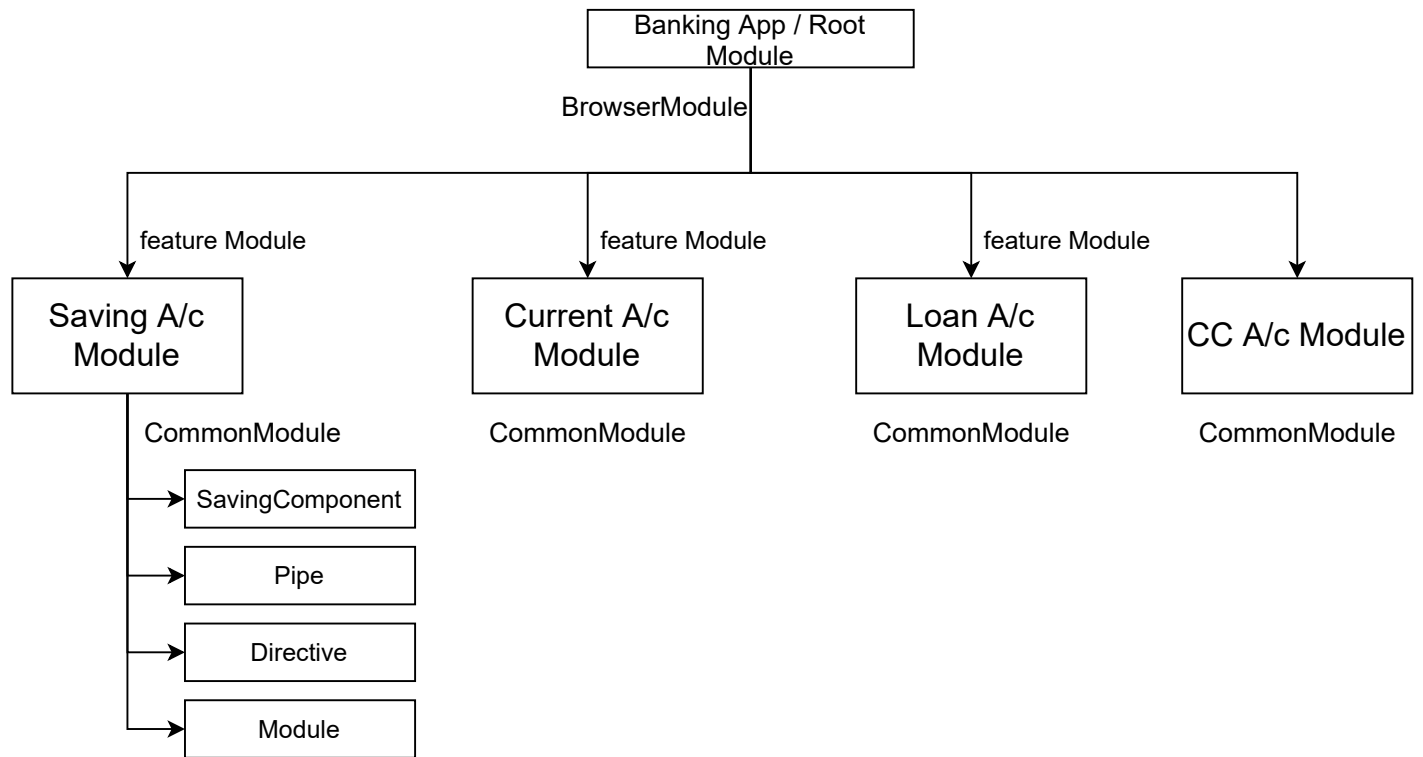
HTML5 Mode - <http://www.example.com/index/about>
HashBang Mode - <http://www.example.com/index#/first>



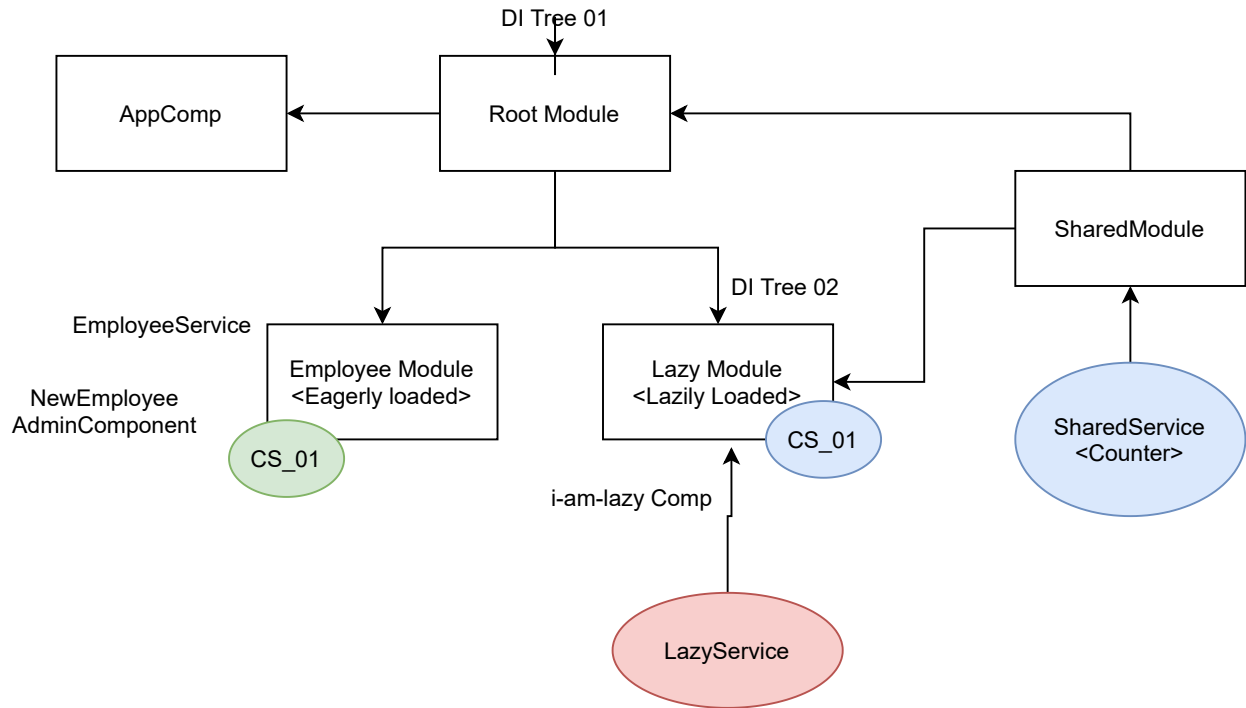
<http://www.exmaple.com/products/overview/99>
<http://www.exmaple.com/products/specification>

<http://www.example.com/product/specification?author=johnDoe>

Module System & Folder Structure

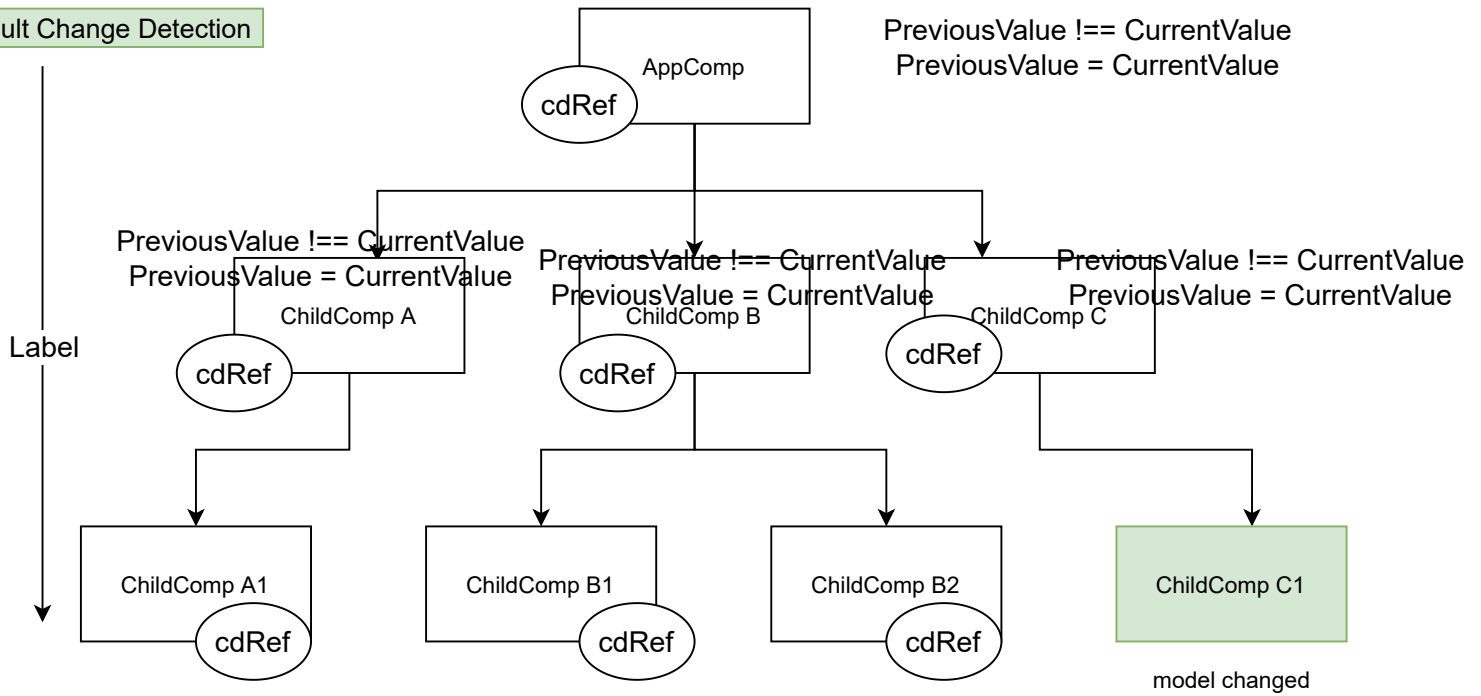


Understanding Shared Module



Change Detection Strategies

default Change Detection



onPush Change Detection

