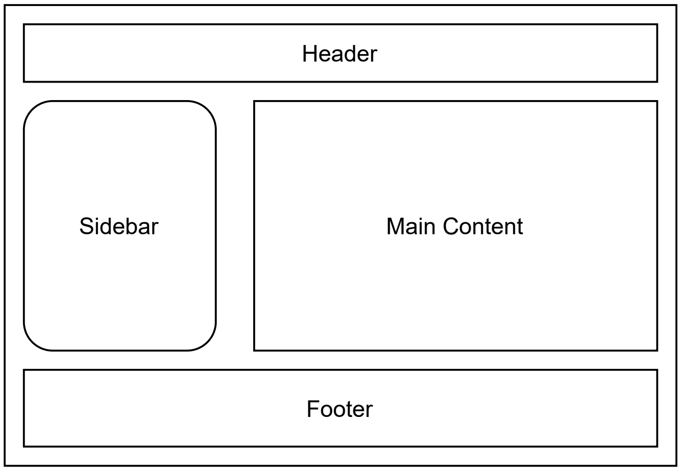
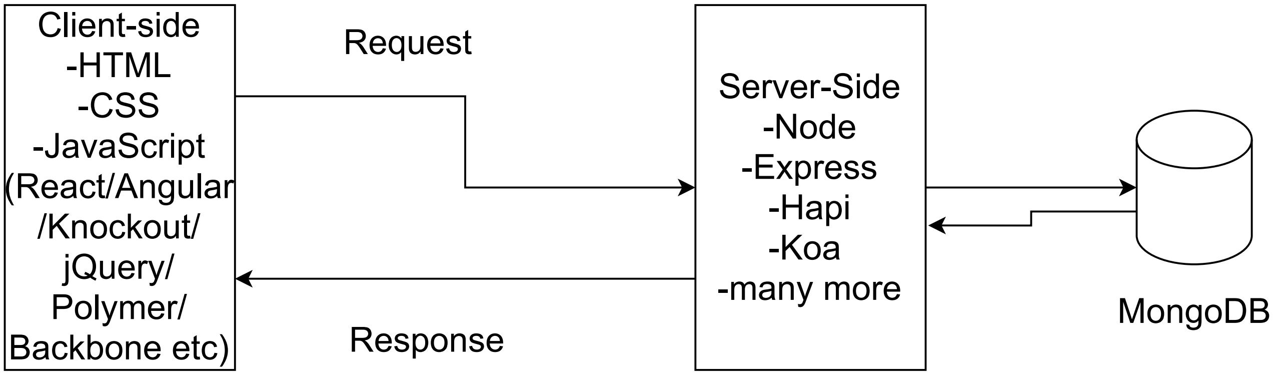
Webpage Structure



MEAN - MongoDB ExpressJS AngularJS NodeJS

JSON - JavaScript Object Notation

MERN / MEAN Full Stack Diagram



JavaScript Libraries -

1- React - Component based architecture (reusable piece of code), Data changes, DOM manipulation, SPA, data binding, template structure, state management, performant app etc.
 - react-router and react-router-dom
 - redux react-redux

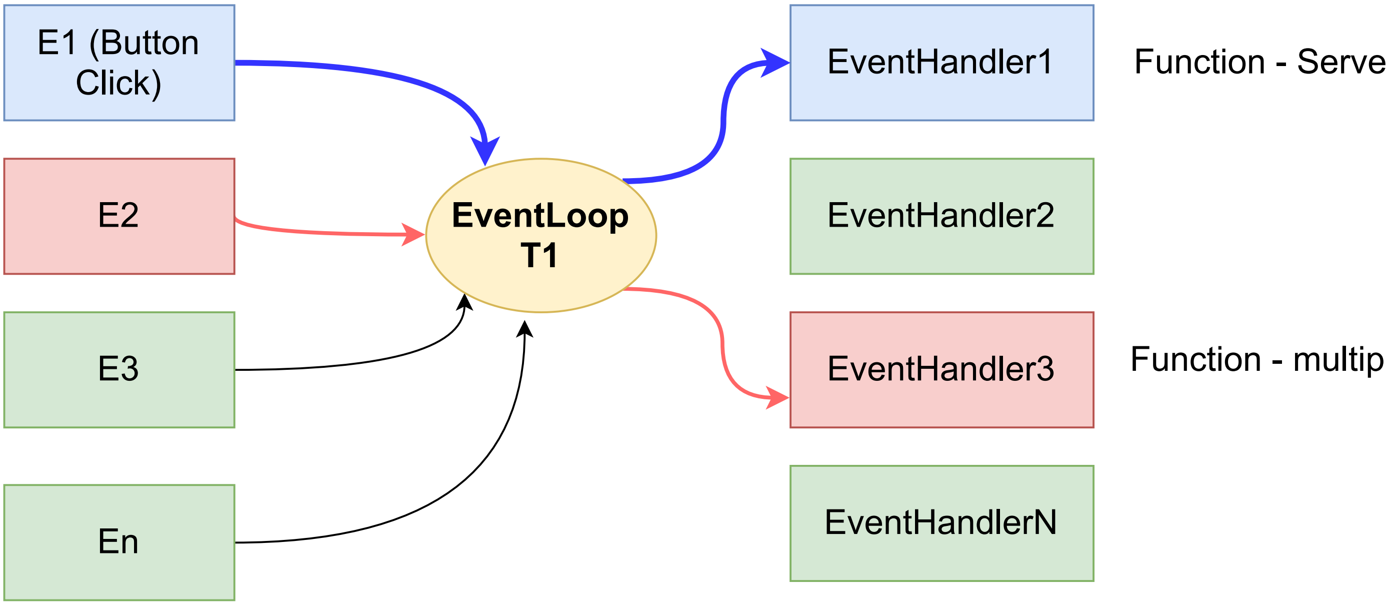
2 - Jquery - DOM Manipulation, Animation, XHR call (remote server calls)

3- Knockout - 2 way data binding, MVVM Pattern

4- BackboneJS - MVC Pattern with 2 way data binding

5 - Angular Framework - provides complete skeleton to create enterprise applications e.g Banking System

Event Queue



E1 (Button Click)

E2

E3

En

EventLoop T1

EventHandler1

EventHandler2

EventHandler3

EventHandlerN

Function - Serve

Function - multip

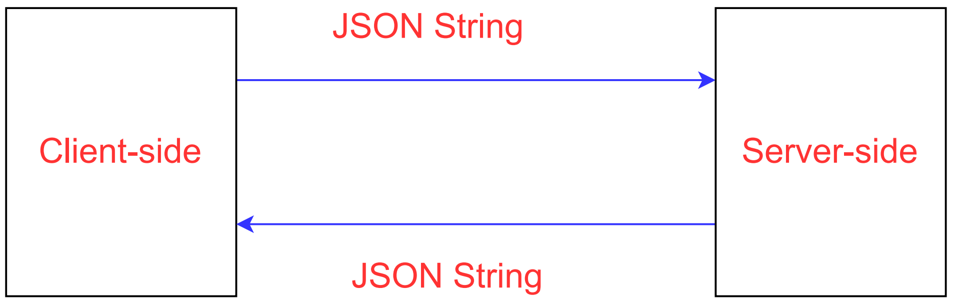
Handling Async JS

- Callback functions
- Promises
- Async...await
- Observables

r Call - 3S

ly the [100*100] - 1S

JavaScript Objects

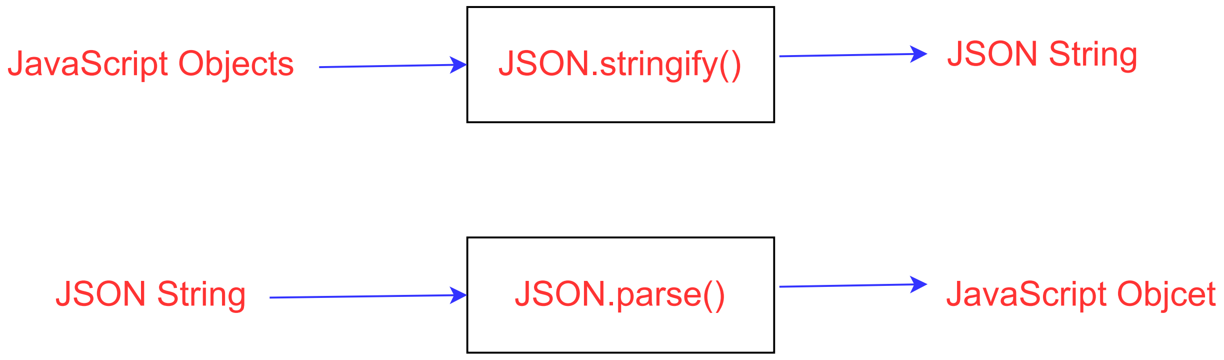


Client-side

JSON String

Server-side

JSON String



JavaScript Objects

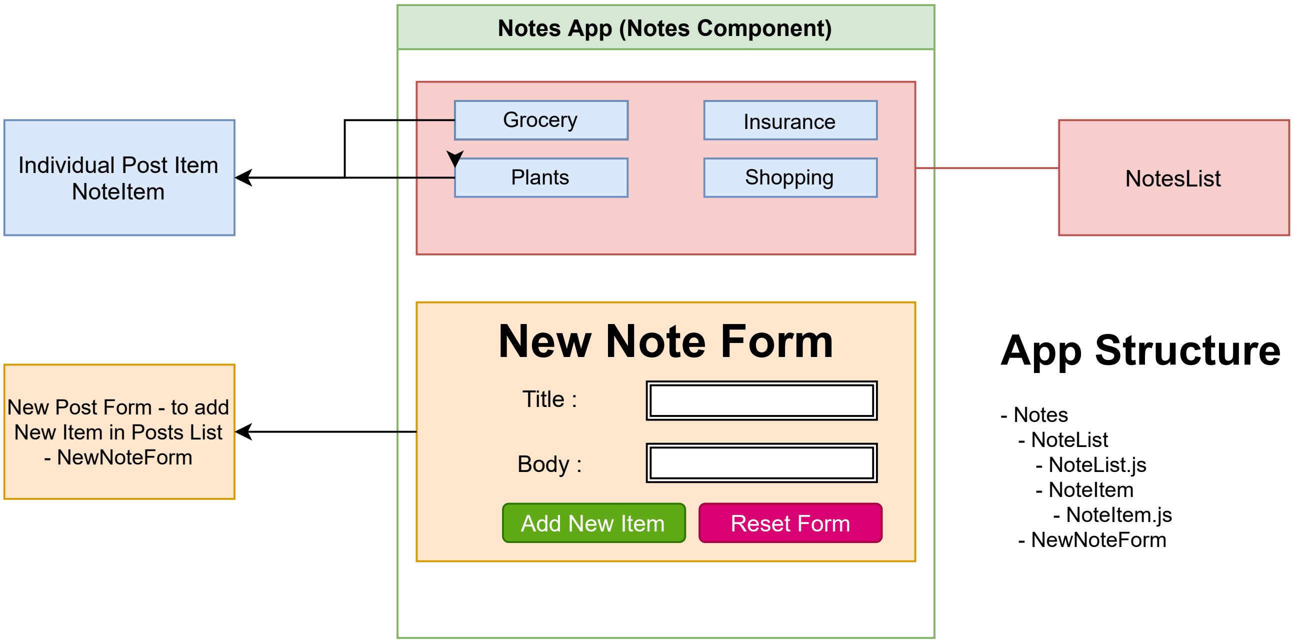
JSON.stringify()

JSON String

JSON String

JSON.parse()

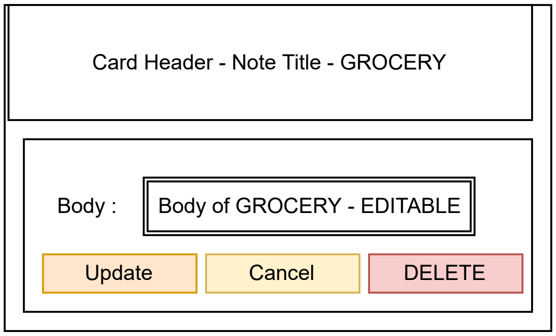
JavaScript Objcet



App Structure

- Notes
  - NoteList
    - NoteList.js
    - NoteItem
      - NoteItem.js
  - NewNoteForm

Edit Note Component


Card Header - Note Title - GROCERY
Body :
Body of GROCERY - EDITABLE
Update
Cancel
DELETE

Notes App -

Note - { id: 1, title: "Shopping", body : "buy the jeans"}

- Bootstrap Library - CSS classes

ES6 Modules resolution -

- Babel - .babelrc

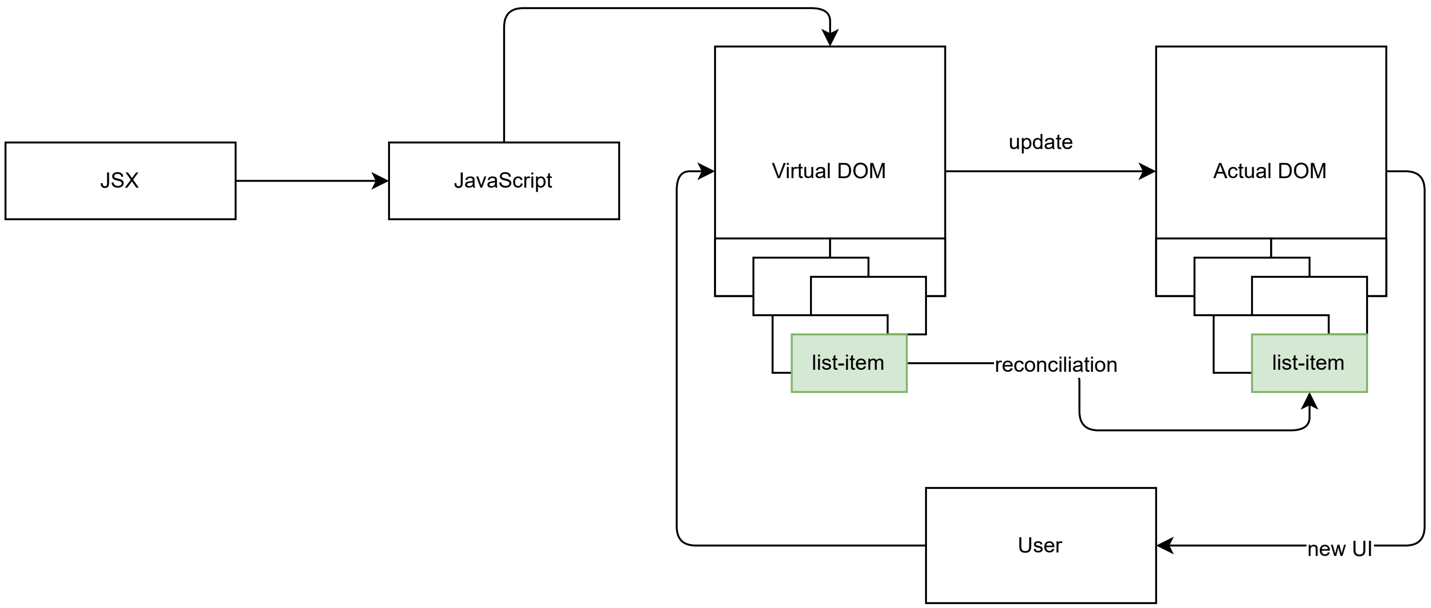
- Webpack - webpack.config.js

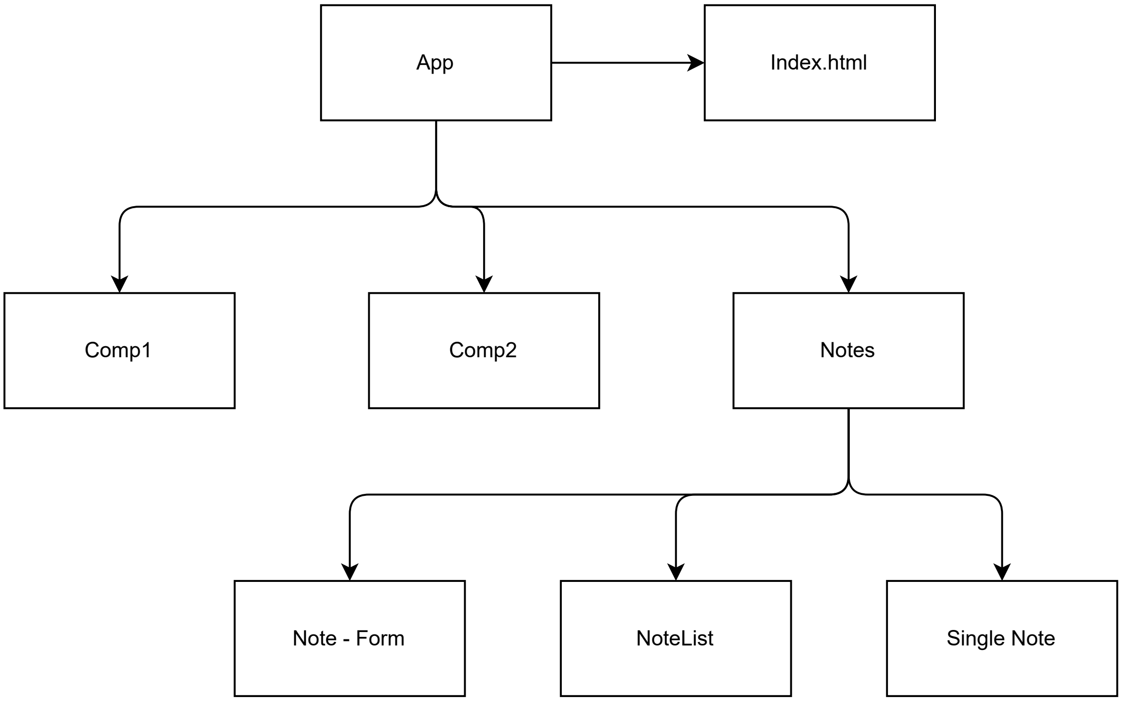
-.esm / ejs - NA

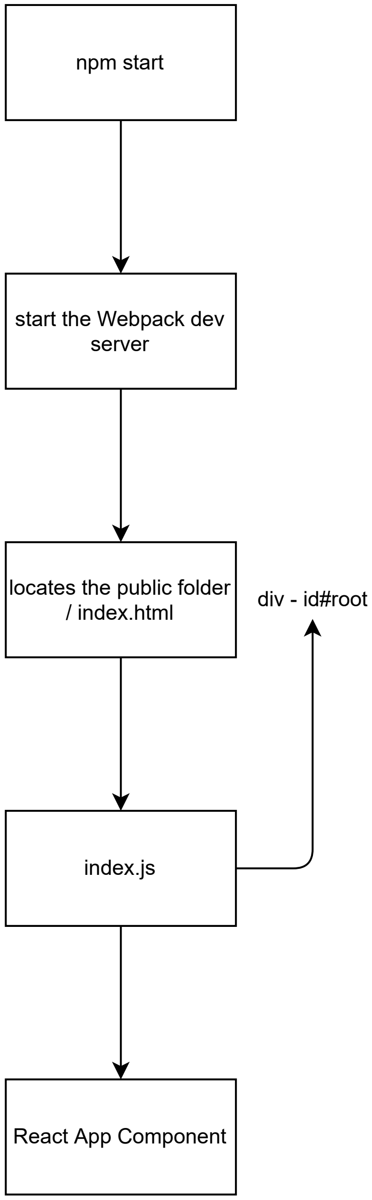
-SystemJS -

- RequireJS - config.js

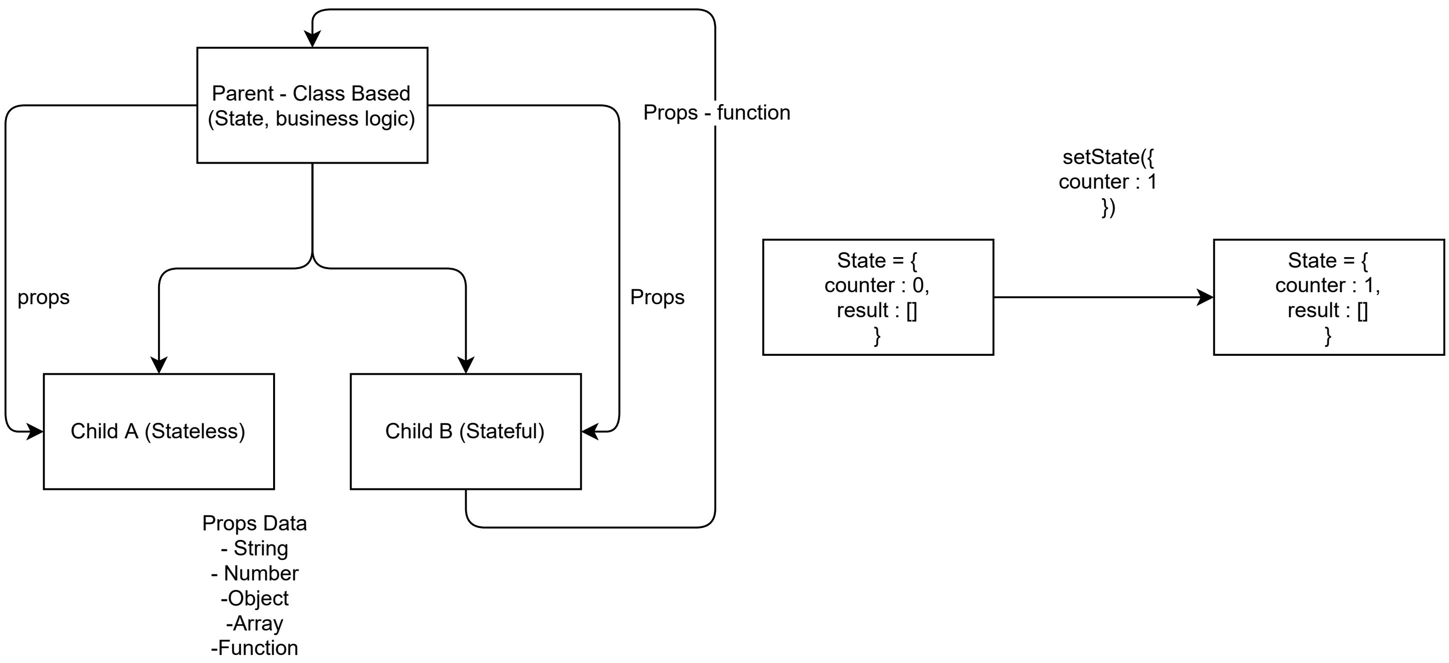
Diffing Algorithm




App
Index.html
Comp1
Comp2
Notes
Note - Form
NoteList
Single Note


npm start
start the Webpack dev server
locates the public folder / index.html
div - id#root
index.js
React App Component

Component Communication



Parent - Class Based (State, business logic)

props

Props

Props - function

Child A (Stateless)

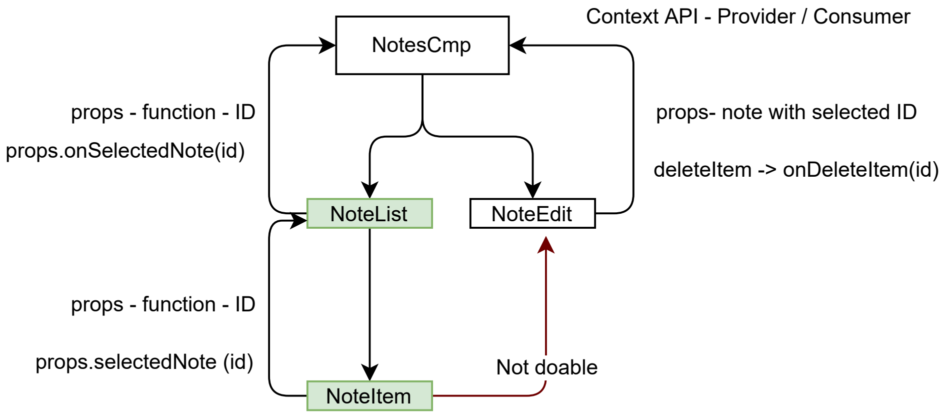
Child B (Stateful)

Props Data
- String
- Number
-Object
-Array
-Function

State = {
counter : 0,
result : []
}

setState({
counter : 1
})

State = {
counter : 1,
result : []
}



NotesCmp

Context API - Provider / Consumer

props - function - ID

props.onSelectedNote(id)

props- note with selected ID

deleteItem -> onDeleteItem(id)

NoteList

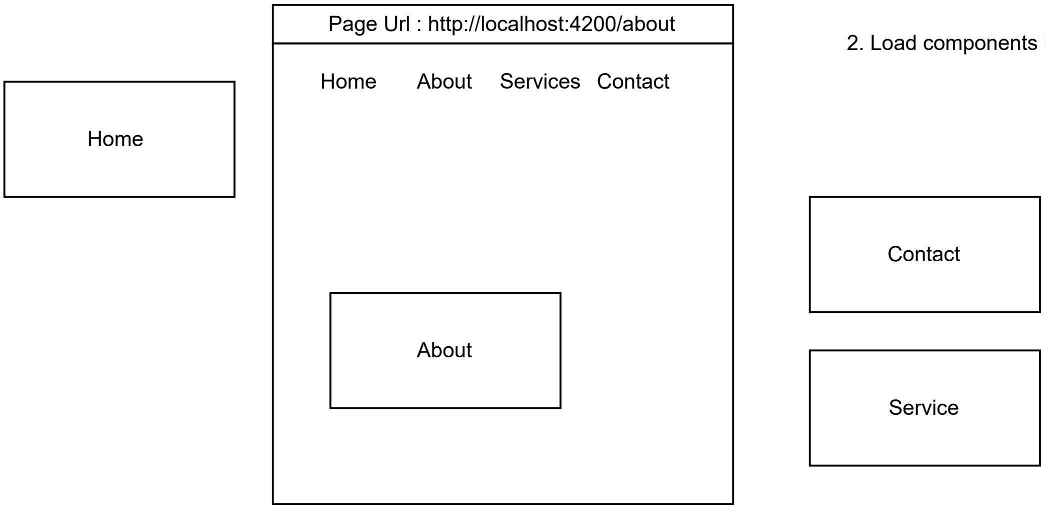
NoteEdit

props - function - ID

props.selectedNote (id)

NoteItem

Not doable

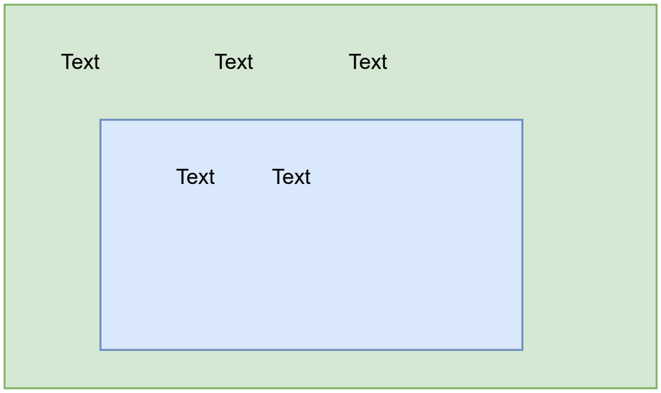


1. Set up Links to change the URL

2. Load components based on the URL

Routing

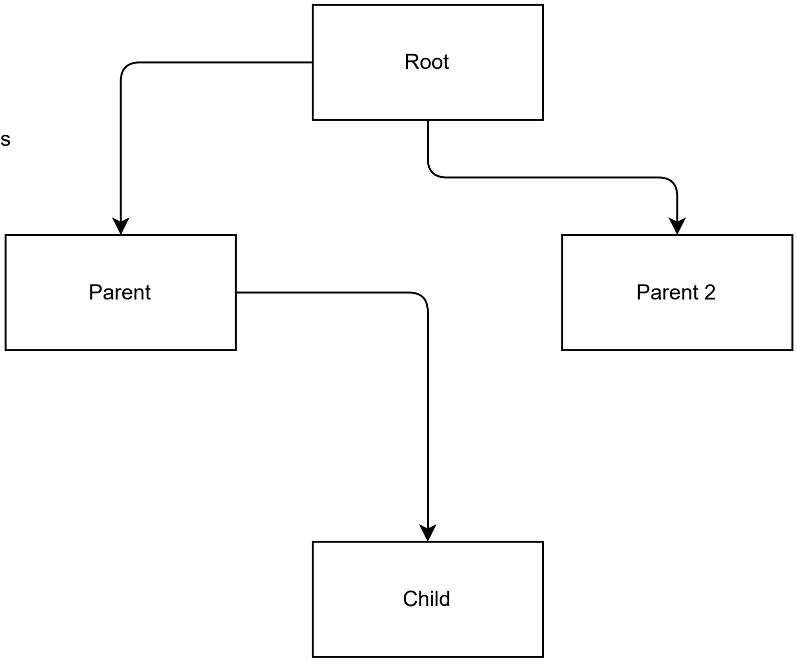
1. Check base href in index.html
2. Create configuration for loading components using **Routes**
3. Tell angular about route configuration using **RouterModule.forRoot()**
4. Set up the router links using **routerLink**
5. provide space to load the component template using **<router-outlet>**
6. To navigate the user programmatically **- Inject the Router Service**
7. **Child Routing/Nested Routing** http://localhost:4200/products/overview
8. **Route parameter** - http://localhost:4200/product/3/overview
9. **Query Parameter** - http://localhost:4200/product?name=iphone
10. **ActivatedRoute** Service can handle query and route parameters

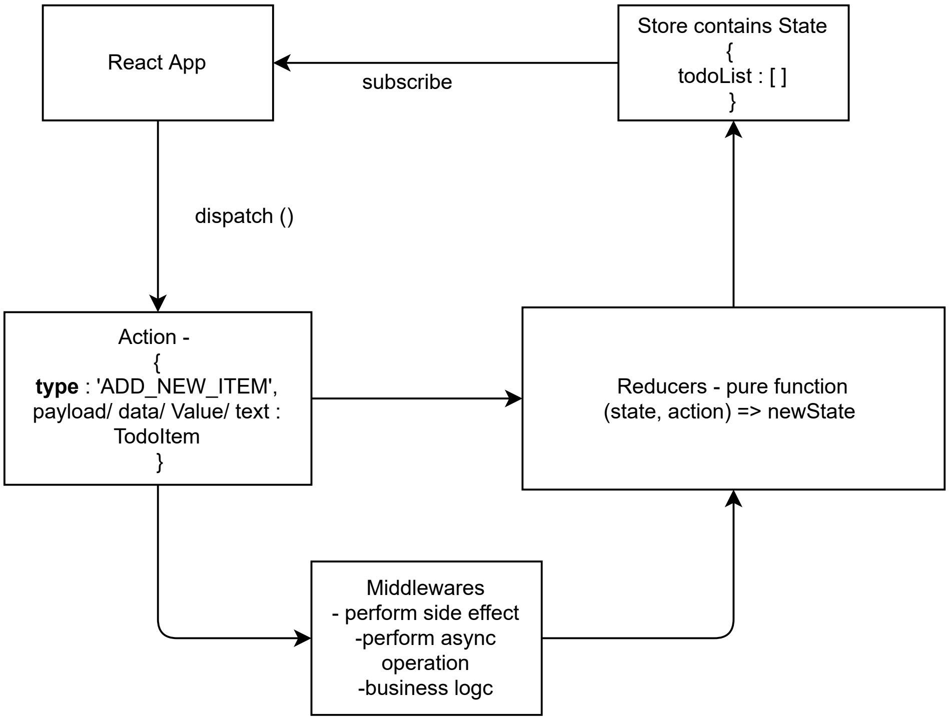


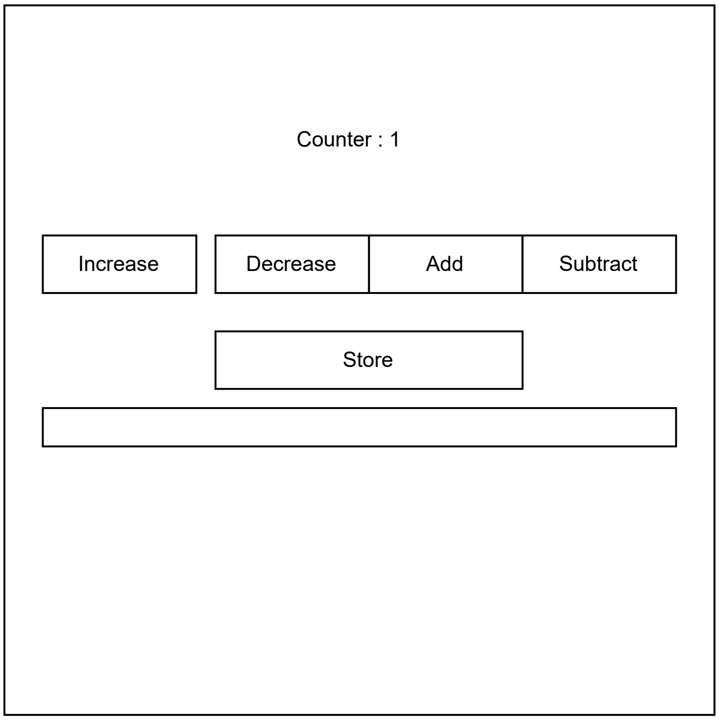
http://localhost:3000/parent/child1

Store maintains State

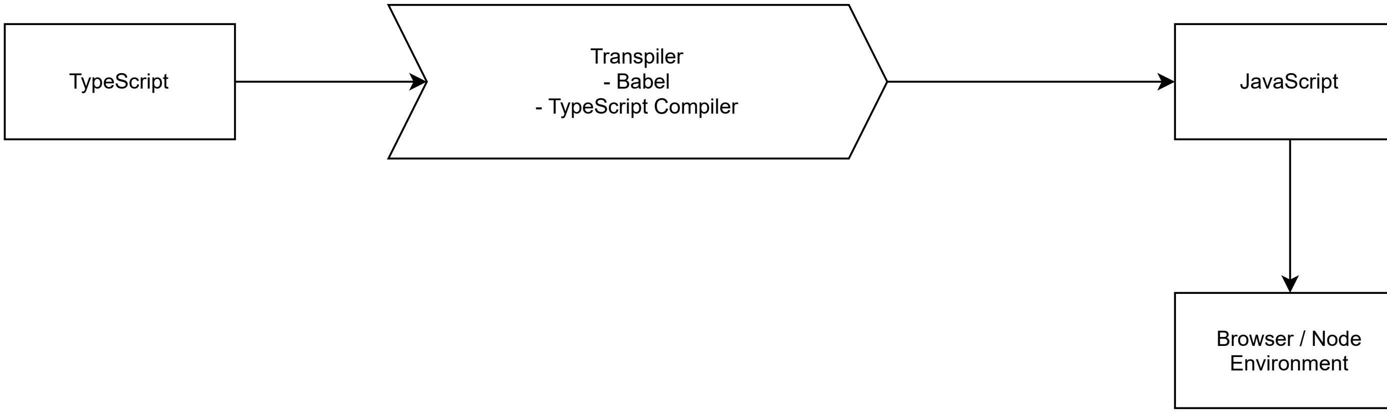
Data flow using props

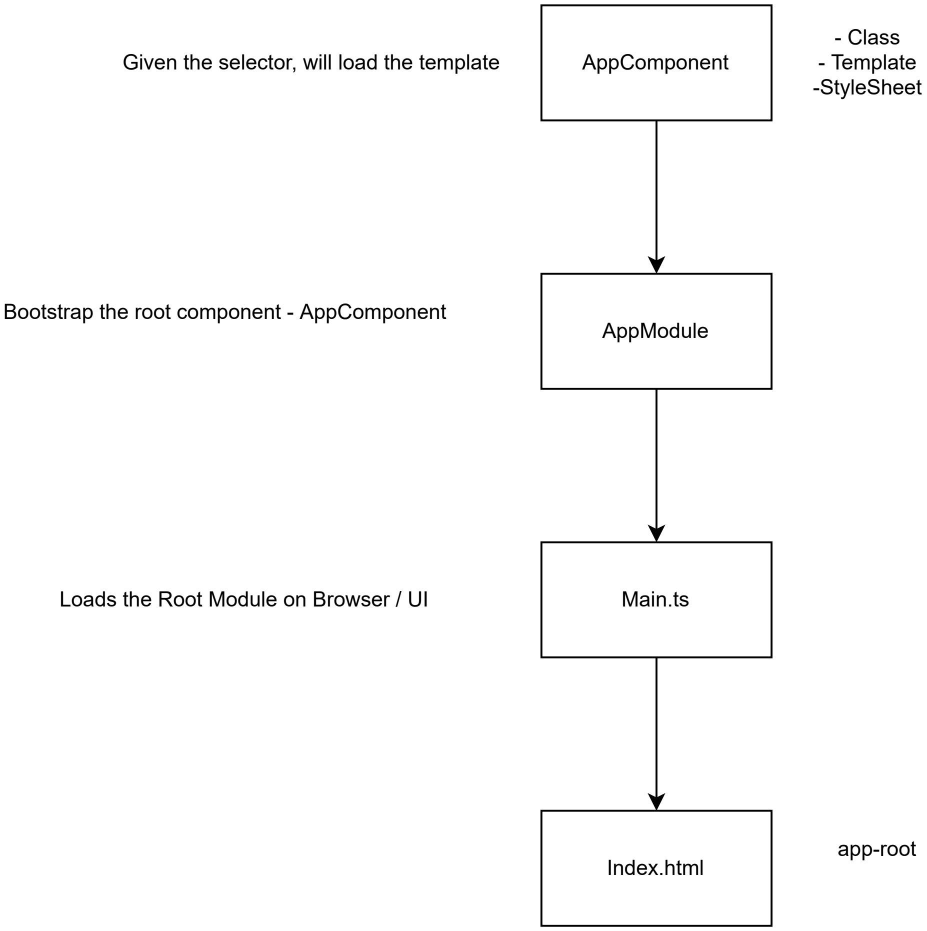



React App
subscribe
Store contains State
{
todoList : [ ]
}
dispatch ()
Action -
{
type : 'ADD_NEW_ITEM',
payload/ data/ Value/ text :
TodoItem
}
Reducers - pure function
(state, action) => newState
Middlewares
- perform side effect
-perform async
operation
-business logc



```
{
counter : 101,
result : [1, 0, 10, 5]
}
```


TypeScript
Transpiler
- Babel
- TypeScript Compiler
JavaScript
Browser / Node
Environment



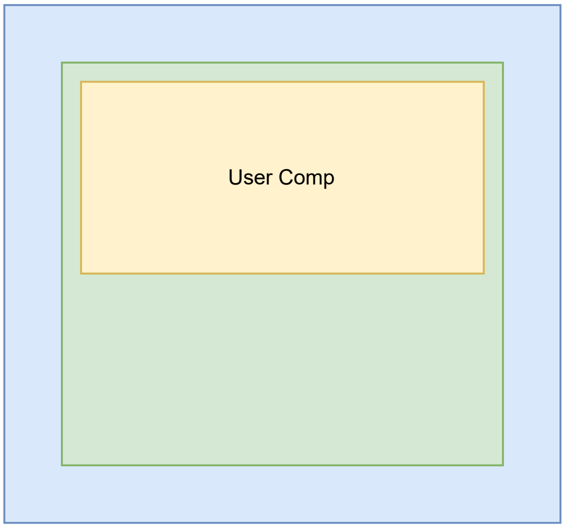
Given the selector, will load the template

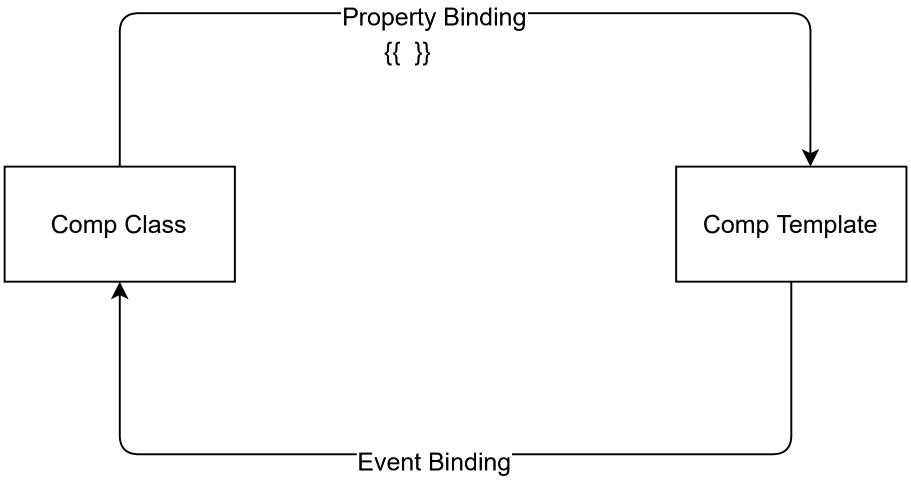
- Class
- Template
-StyleSheet

Bootstrap the root component - AppComponent

Loads the Root Module on Browser / UI

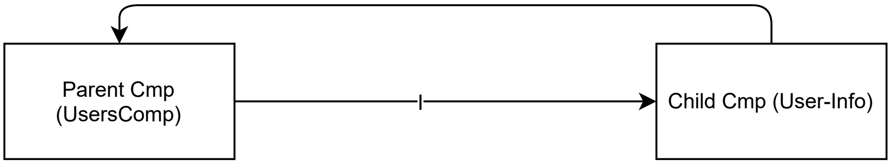
app-root





- Property Binding Syntax - [ ]
- Event Binding Syntax - ( )

- 2 way data Binding
- [Property Binding] + (Event Binding)
- Banana in the Box - [(ngModel)]

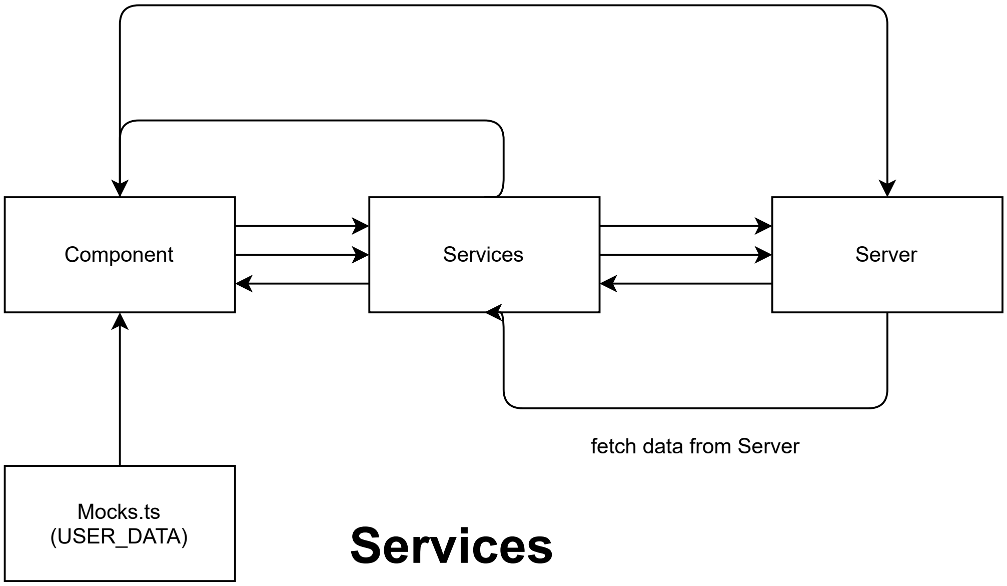


Parent to Child Communication

- @Input () with property binding Syntax

Child to Parent Communication

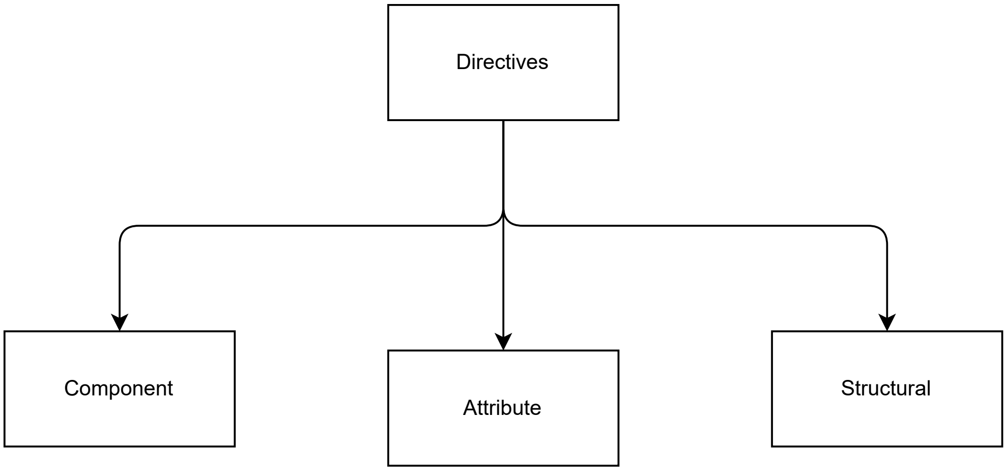
- @Output () with Event binding Syntax



Services

- DI Principle

- Single Responsibility principle



- ngModel

- ngStyle

ngClass

- *ngIf

- *ngFor

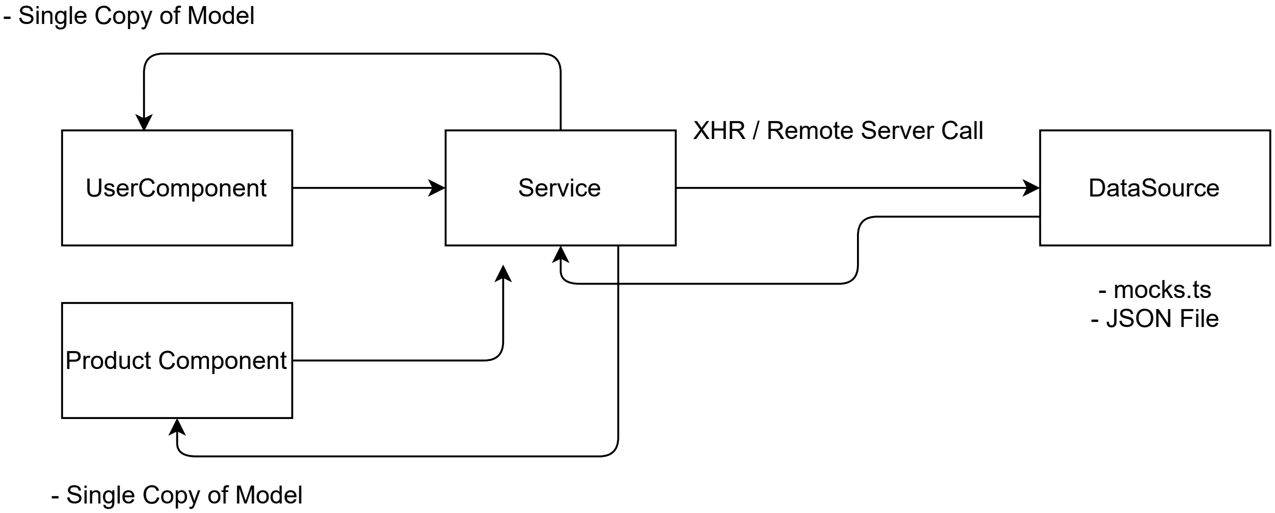
- *ngSwitch

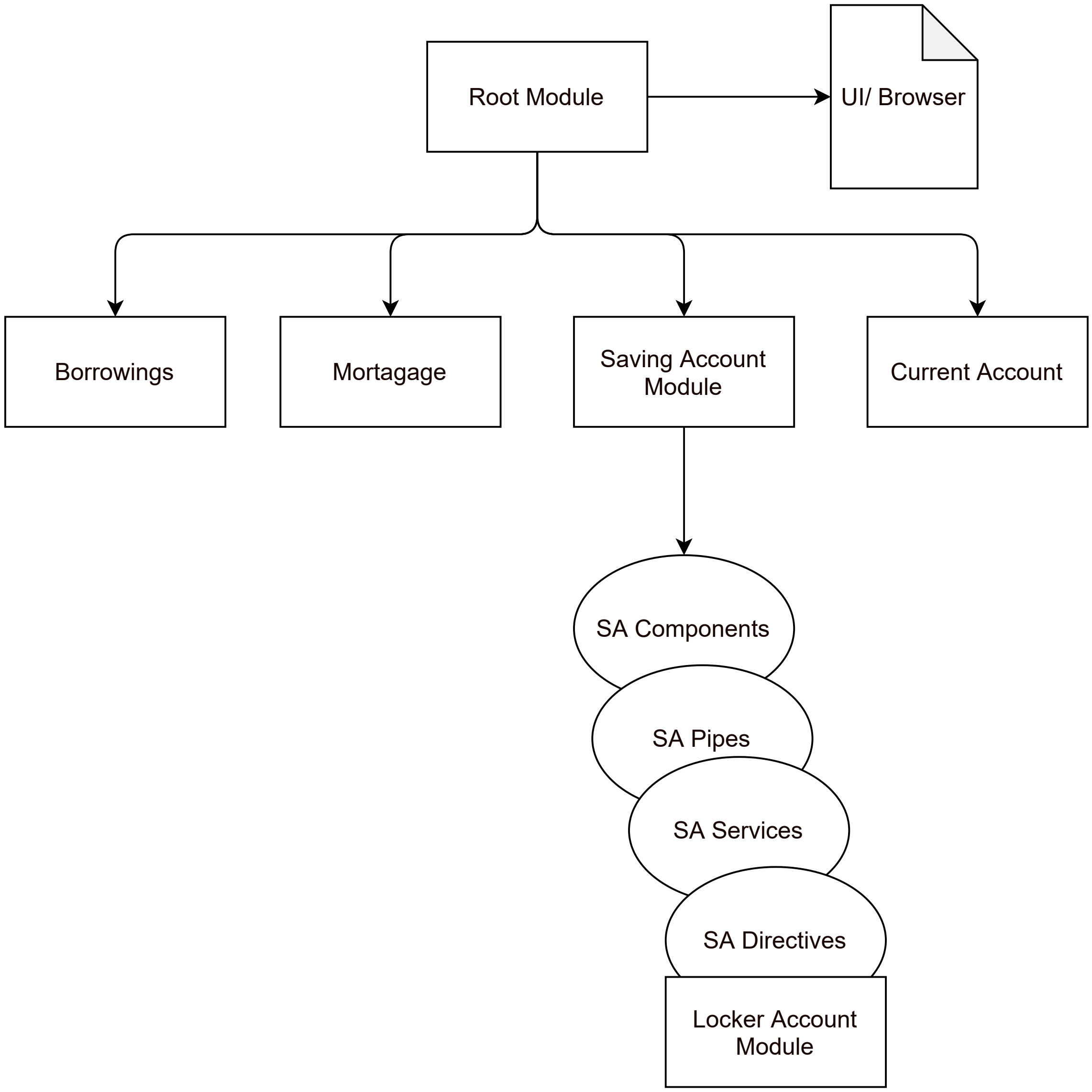
Form and Form Controls Classes

- ngValid / ngInvalid

- ngPristine/ ngDirty

-ngUntouched / ngTouched


- Single Copy of Model
UserComponent
Service
XHR / Remote Server Call
DataSource
- mocks.ts
- JSON File
Product Component
- Single Copy of Model


Root Module
UI/ Browser
Borrowings
Mortagage
Saving Account Module
Current Account
SA Components
SA Pipes
SA Services
SA Directives
Locker Account Module