ANGULAR

One framework.

Web, Mobile & Desktop.

Agenda

> What is Angular?
> Motivation
> TypeScript?
> Single Page Application (SPA)
> Development Environment Setup
> Hello World!
> Bootstrapping the app
> Building Blocks & Architecture

What is Angular?

What is Angular ?

An Open-Source JavaScript Framework, used to build Single Page based Web Application (SPA),

Developed by Google,

Release date Nov 2022,

Current version 15 (stable).

Motivation

Motivation

- Speed & Performance,
- Smaller application,
- Modular application,
- Cross-platform Web, Mobile & Desktop,
- SPA,
- RESTful API,
- Uses TypeScript,
- Rxjs API Integration,
- & It has a huge community on GitHub.

TypeScript ?

TypeScript ?

Free & Open-Source programming language,

Developed & maintained by Microsoft,

A superset of JavaScript,

Full Object-Oriented Programming with features like classes, interfaces & modules…

Support for ECMAScript 5 and uses arrow function syntax,

Compiles to plain JavaScript.

Single Page Application (SPA)

Single Page Application ?

One and Only HTML page for the entire application,

Dynamically update the single page with new data if needed, as the user interacts with the app,

Show and Hide some components during the interaction process,

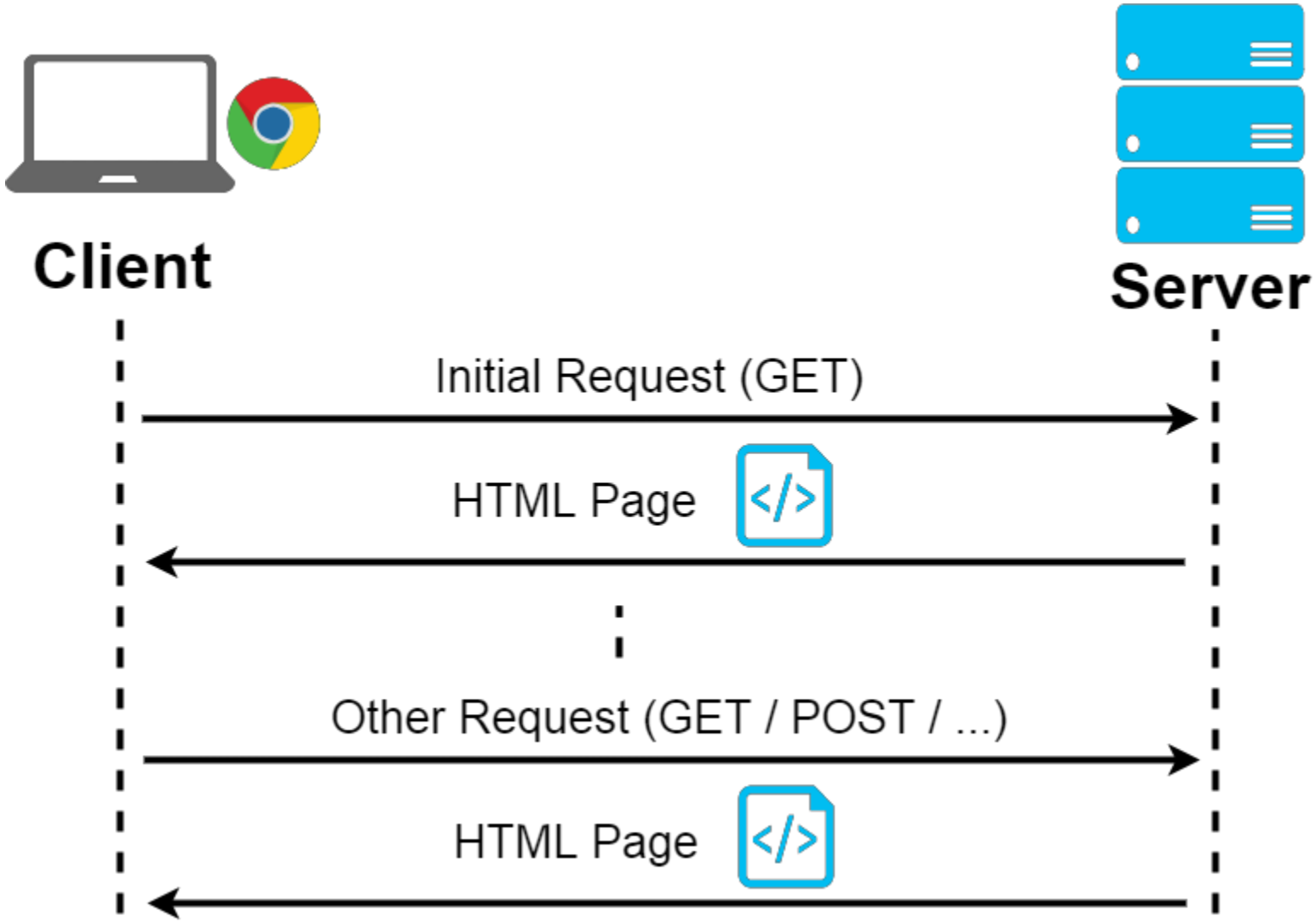
No reloading or refreshing during navigation,

More responsive & fluid app as a result,

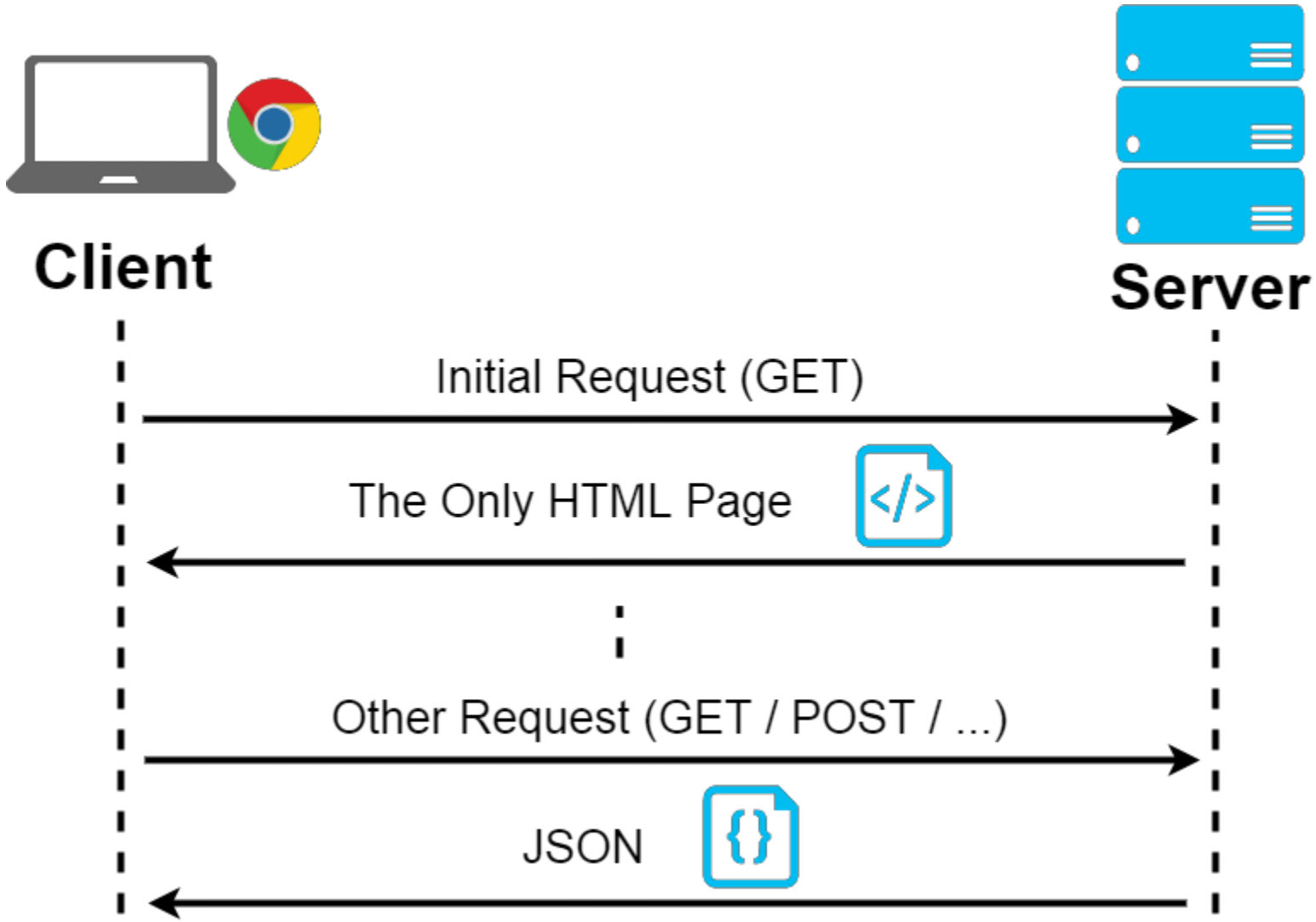
Full separation between the presentation logic & business logic,

Asynchronous Requests (AJAX), JSON response & RESTful API.

Traditional way

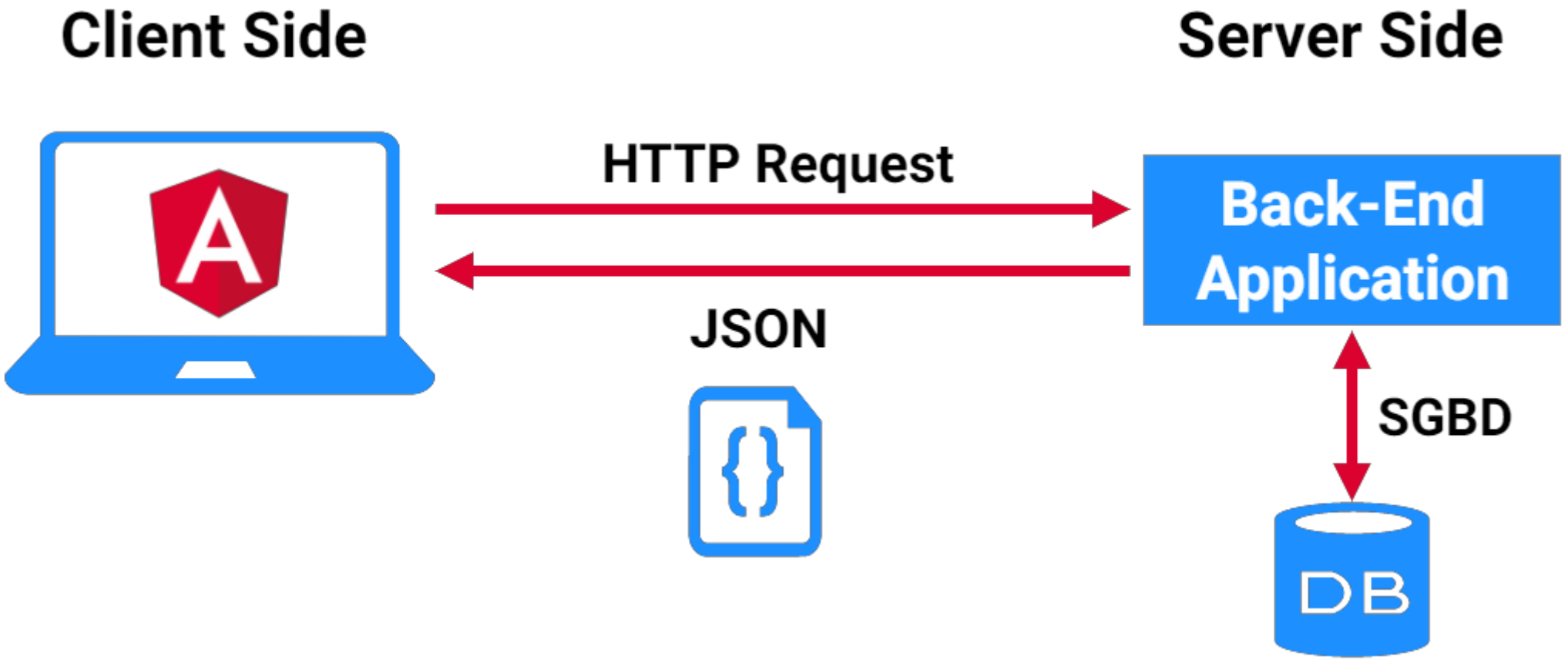


Single Page Application



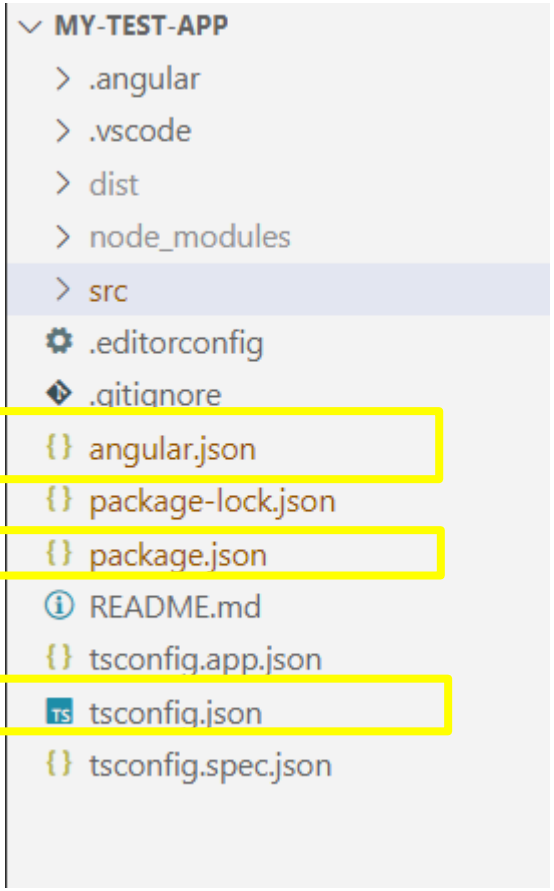
ANGULAR for Client Web App

ANGULAR for Client Web App



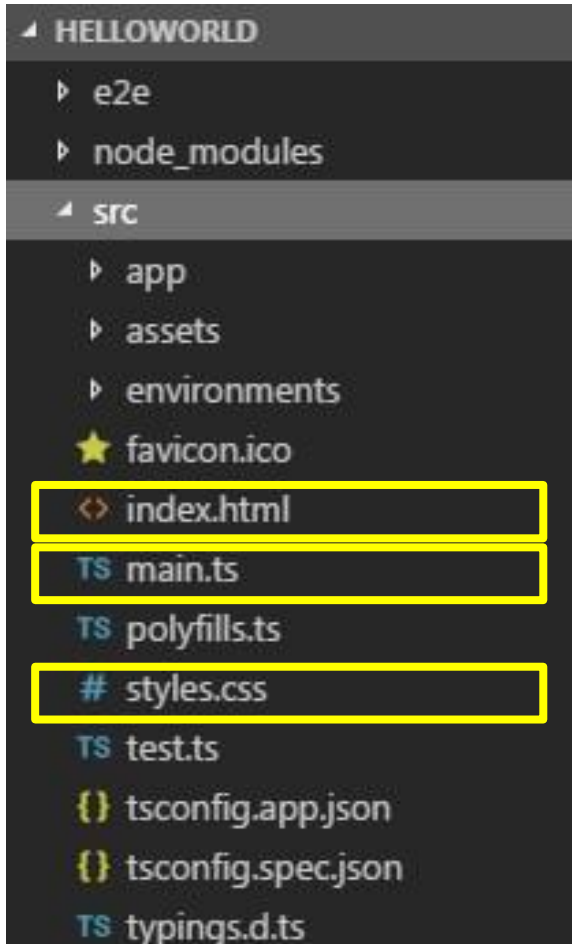
Basic Structure of ANGULAR Application

Basic Structure of ANGULAR Application



.angular.json , package.json and tsconfig.json are the responsible files on the project configuration, it's dependency management and it's external packages.

Basic Structure of ANGULAR Application

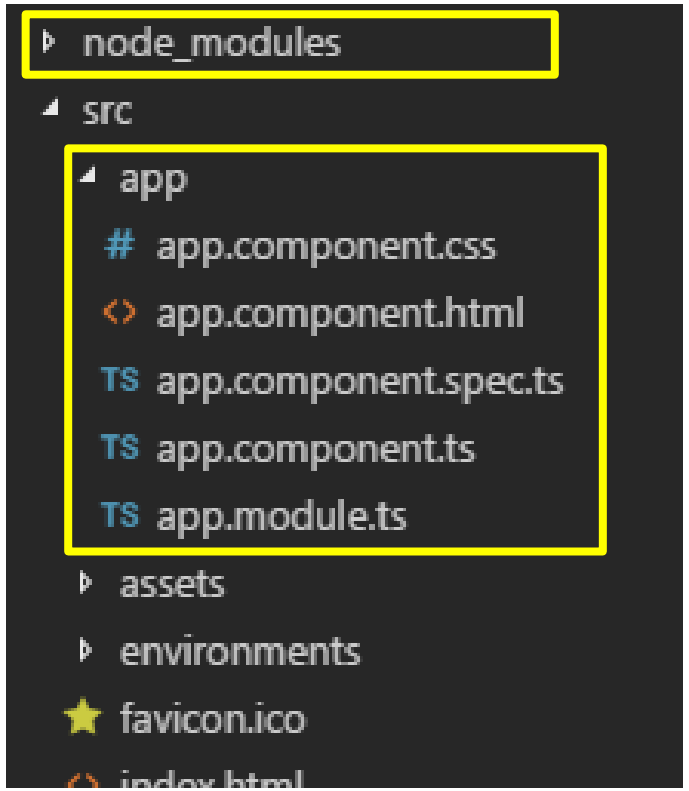


main.ts and **index.html** are the **entry point** to the application,

Since we are developing a **SPA** **index.html** is the **only HTML page** in the whole project,

style.css is the style sheet for the **global app design**.

Basic Structure of ANGULAR Application

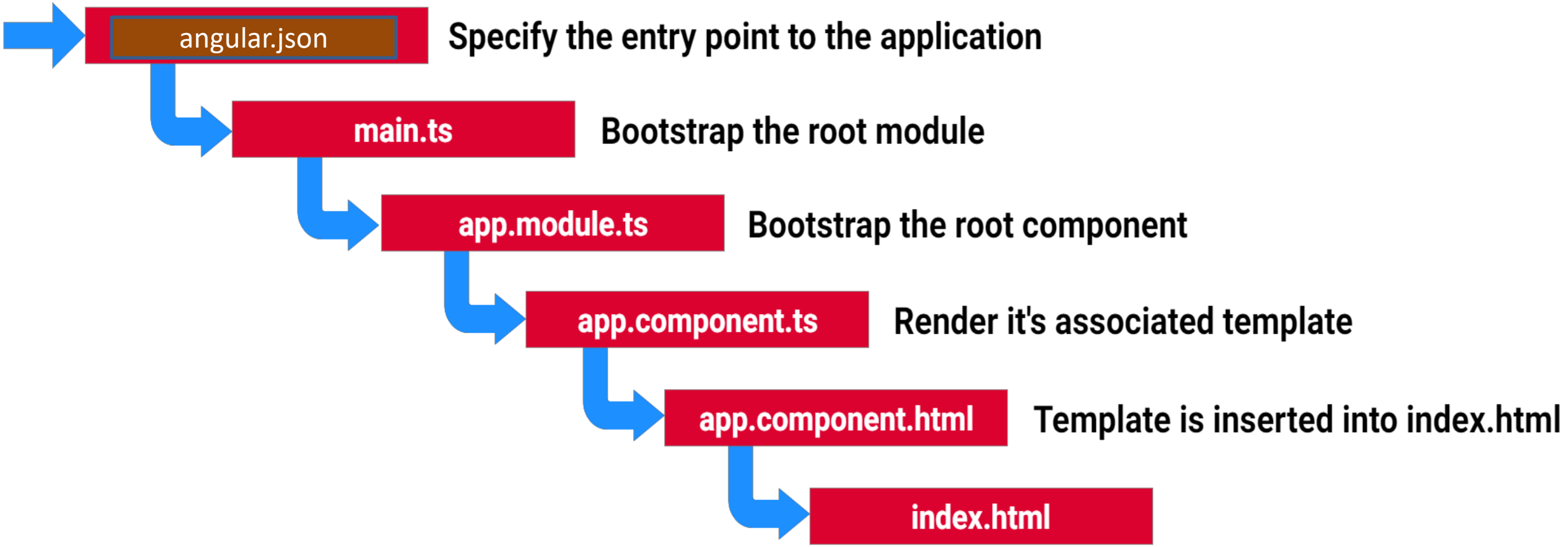


node_modules folder contains the packages installed by the npm tool,

app folder contains all the work, components, modules...

Bootstrapping the app

Bootstrapping the app



Building Blocks

Building Blocks

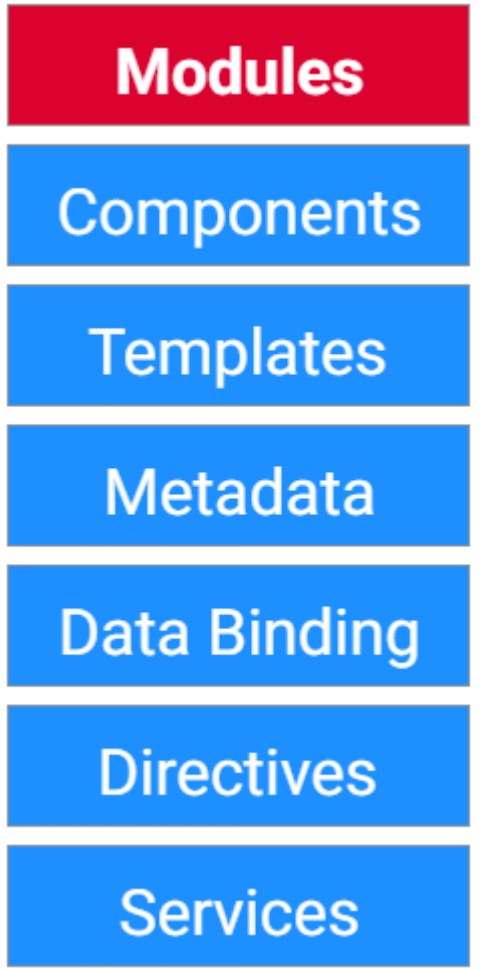
Every Angular app has at least one Module, the root module named AppModule,

An Angular app is a Modular app,

A module is a TypeScript class with the Decorator @NgModule,

Each module in the application has it's own Components, Directives, Services... They should be declared in @NgModule decorator,

Modules can cooperate to achieve some app functionalities.



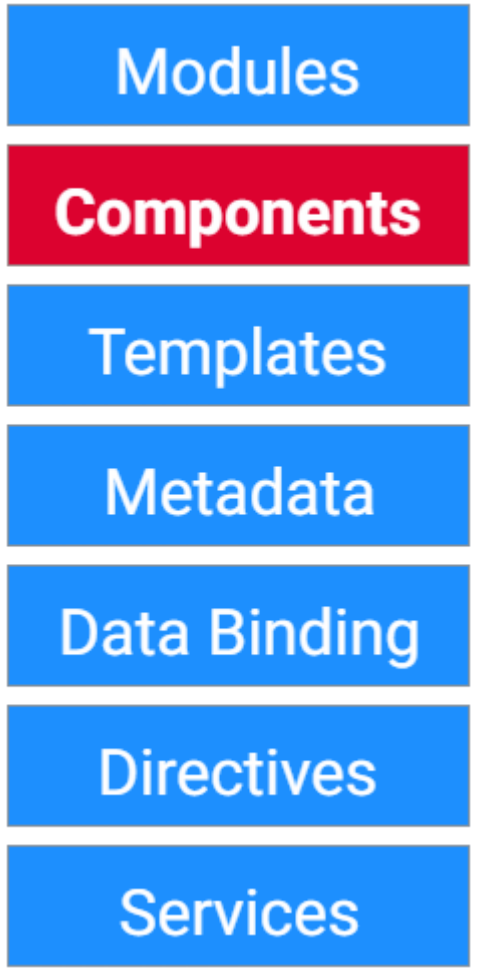
Building Blocks

A Component controls a piece of screen called view,

This view is defined by the Template associated to the Component,

A Component handles the user interaction with the view (Template), passes data and properties to the view and updates it dynamically,

A Component is a TypeScript class with the Decorator @Component.



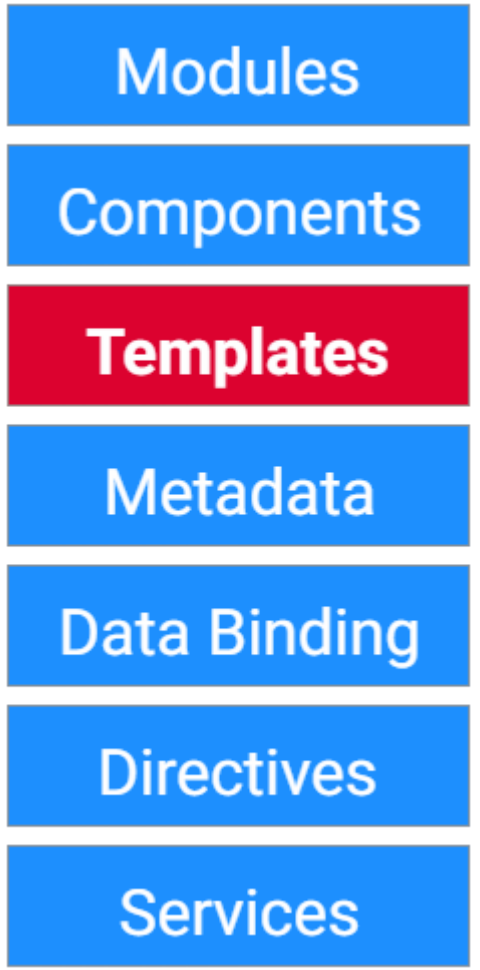
Building Blocks

A Component view is defined by its associated Template,

A Template is bunch of HTML tags,

Along side with the HTML, a Template typically contains some particular expressions and syntax which belong to Angular,

A Template may contain some Custom Tags, that represent another Component, the selectors.



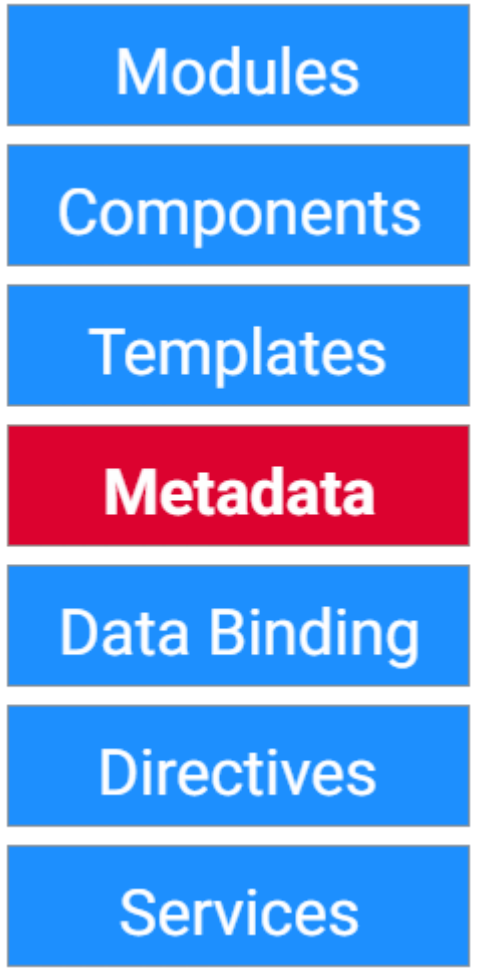
Building Blocks

A Module, a Component, a Directive or a Service are just TypeScript Classes until we tell Angular about the difference between them, and that is done by the Metadata,

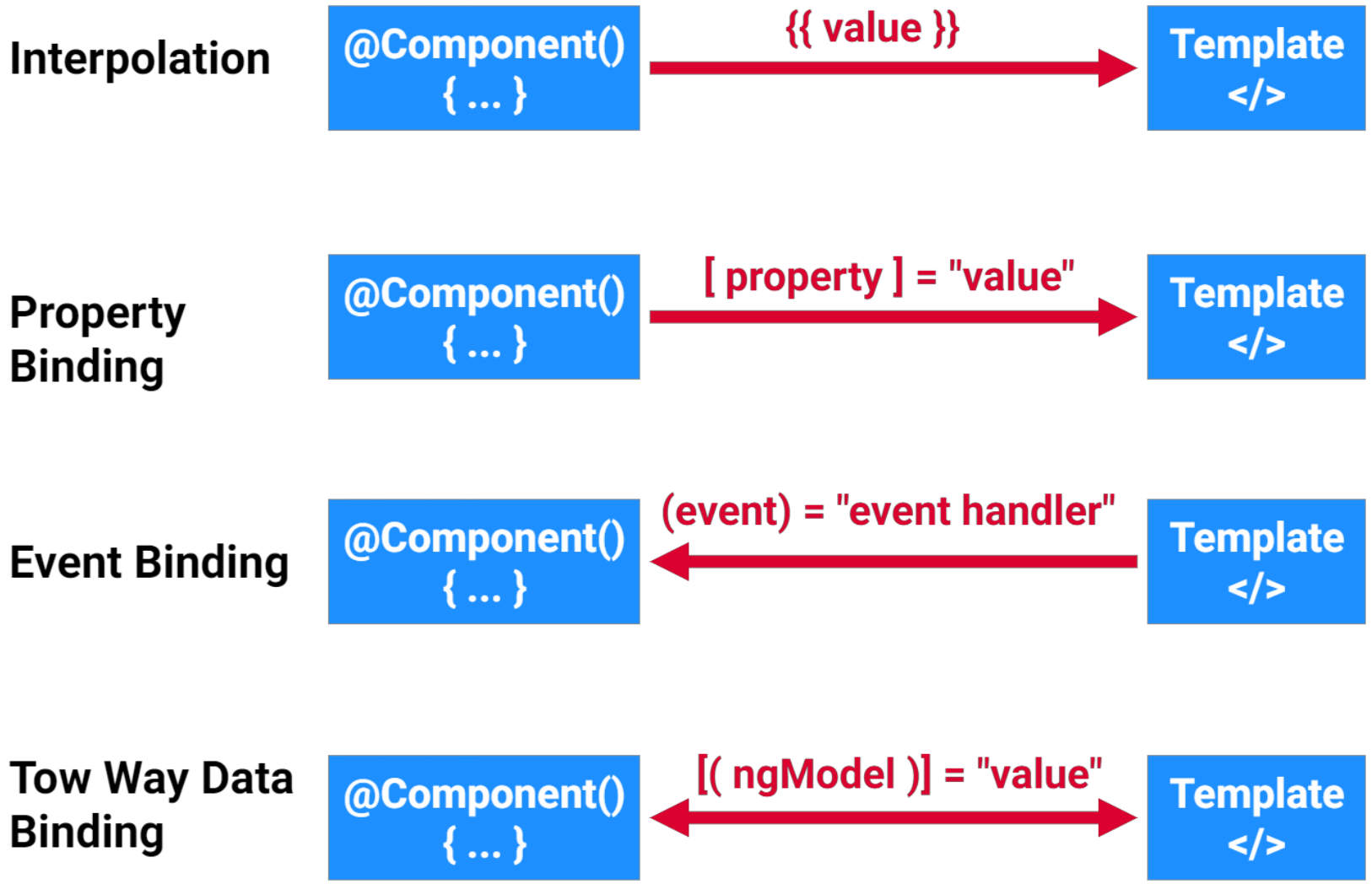
We attach Metadata in TypeScript to a class by using Decorators, @NgModule, @Component, @Injectable…

Metadata tells Angular which Template belongs to which Component and which Component belongs to which Module…

Class + @Component + Metadata = A Component



Building Blocks



- Modules
- Components
- Templates
- Metadata
- **Data Binding**
- Directives
- Services

Building Blocks

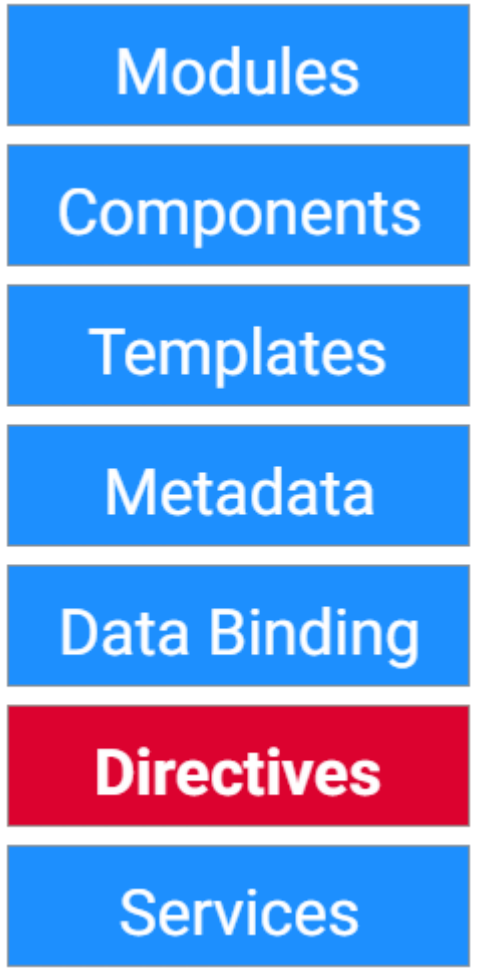
A Directive is a TypeScript class with the Decorator @Directive,

Directives appear within HTML tag as attributes,

Two kind of directives: Structural &Attribute Directives,

Structural Directives change the layout by addingor removing Template elements,
Example: *ngFor & *ngIf.

Attribute Directives change the appearance or the behavior of an existing Template element,
Example: ngModel, ngStyle & ngClass.



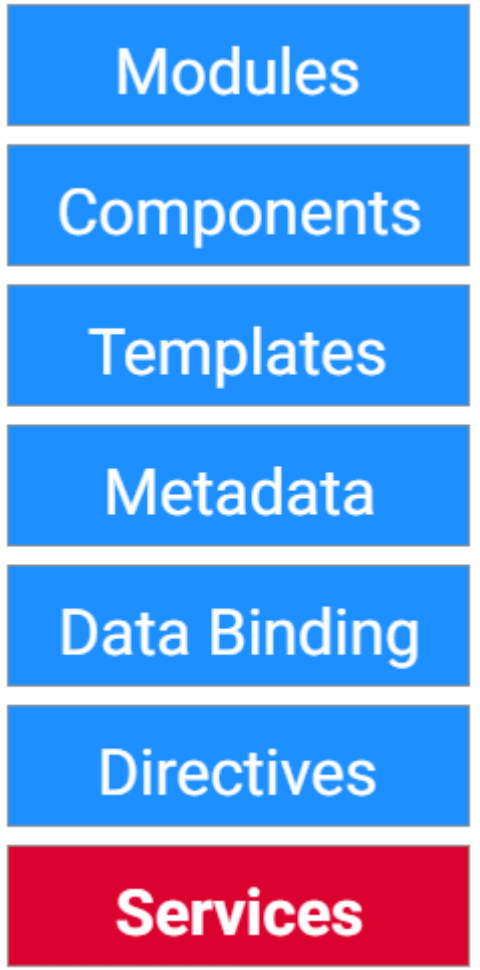
Building Blocks

A Service is a TypeScript class with the Decorator @Injectable,

A Service achieve some functionalities for the application such us fetching data from server, authentication...

This kind of functionalities should not be done in the Component,

A Component should only take care of rendering the Template (view).



Thanks for your attention !!