

```
=====
Angular Integration with Spring Boot
=====
```

- => Angular is a client side framework
- => Angular framework developed by Google company
- => Angular developed using TypeScript
- => Angular is free & open source
- => Angular supports multiple browsers
- => Angular is mainly used for SPA (single page app)

Note: Angular JS & Angular framework both are not same.

- => Angular JS developed using Java Script. (Angular 1.x)

=> Google identified some performance issues in Angular JS 1.x version then they re-developed angular is Typescript which is called as Angular framework.

Note: From 2.x version onwards it is called as Angular Framework.

```
=====
Angular Setup
=====
```

Step-1 : Download and Install Node

URL : <https://nodejs.org/en/>

Note: After installation, verify node version

```
$ node -v
```

Step-2 : Install Type Script

```
$ npm install -g typescript
$ tsc -v
```

Step-3 : Install Angular CLI

```
$ npm install @angular/cli -g
$ ng v
```

Step-4 : Download and Install VS Code IDE

URL : <https://code.visualstudio.com/download>

Step-5: Create Angular Application

```
$ ng new app1
$ cd app1
$ ng serve --open
```

Note: Angular apps will use Live server for execution and it will run on 4200 port number.

```
=====
=> Angular application nothing but collection of components
```

Ex: menu-component, login-component, register-component, dashboard-component....

=> In angular application by default "app-component" will be created. It is called as Parent Component.

Note: Every component contains 3 files

- 1) component.ts (logic to handle req & response)
- 2) component.html (presentation file - template)
- 3) component.css (styles for specific component)

=> Every Component will have a selector to access that component.

Ex : <app-root></app-root>

Note: App-Component is the entry point for angular application.

=> index.html page is called as welcome page in angular application.

=> When we run angular application, index.html page will be loaded and it will invoke app-component hence we will get response from app.component.html page.

Note: To apply styles for app-component template we have app-component.css file.

Angular Building Blocks

- 1) Components
- 2) Templates
- 3) Data Binding
- 4) Directives
- 5) Pipes
- 6) Services

=> Component represents small portion of web page

Ex : header component, footer component, menu component, login component, reg component..

=> Template represents view page (html)

Note: Every component will have its own template (html)

=> Data Binding is used to transfer the data between component and template

Component <-----> Template

=> Directives are used to manipulate DOM elements in the template

Ex : ngIf, ngFor, ngModel...

=> Pipes are used to transform the data

Ex : uppercase, lowercase, INR, USD ...

=> Services are used to develop business logic (api calls)

Summary

- 1) What is Angular
- 2) Angular Setup
- 3) Ng First App Development

4) Angular Build blocks

```
=====
Spring Boot with Angular Integration
=====
```

Spring Boot : Used for backend development

Angular : Used for frontend development

Fullstack App : Spring Boot + Angular

```
=====
Spring Boot - Rest API
=====
```

```
@SpringBootApplication
@RestController
@CrossOrigin
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

    @GetMapping("/welcome")
    public String getWelcomeMsg() {
        return "Welcome To Ashok IT";
    }
}
```

```
===== Angular Application Development =====
```

##Step-1 : Create Angular application

```
$ ng new app2
```

##Step-2 : Run angular application

```
$ ng serve --open
```

##Step-3 : Configure httpclient details in "app.config.ts" file

```
-----
export const appConfig: ApplicationConfig = {
  providers: [provideRouter(routes), provideHttpClient()]
};
```

##Step-4: Create Service class and invoke backend rest api

```
$ ng generate service app
```

```
-----
@Injectable({
  providedIn: 'root'
})
export class AppService {

  private welcomeUrl: string;

  constructor(private http: HttpClient) {
    this.welcomeUrl = 'http://localhost:8080/welcome';
  }

  public getWelcomeMsg(){
    return this.http.get(this.welcomeUrl, {responseType:'text'});
  }
}
```

}

##Step-5 : Inject service class in app-component

```
import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';
import { RouterOutlet } from '@angular/router';
import { HttpClient, HttpClientModule } from '@angular/common/http';
import { AppService } from './app.service';

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [CommonModule, RouterOutlet, HttpClientModule],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'app1';

  msg:string="";

  constructor(private appService: AppService){}

  getWelcomeMsg(){
    this.appService.getWelcomeMsg().subscribe(resp => {
      this.msg = resp;
    })
  }
}

##Step-6 : Write Presentation logic in app-template


---


<button (click)="getWelcomeMsg()">Get Message</button>

<hr/>

{{msg}}
```