설치

1. Node.js깔기.
   1. npm package manager

http://cli.angular.io

node -v

npm -v

angular cli 설치

|  |
| --- |
| npm install -g @angular/cli  ng -help |

angular cli

|  |
| --- |
| ng new ProjectName -d |
| 프로젝트 생성 |

|  |
| --- |
| ng new ProjectName --skip-install |
| 프로젝트 생성 이지만 노드모듈폴더는 설치안함. |

라우터 포함 생성하기

|  |
| --- |
| ng new AngularNote --routing |

npm으로 노드 패키지모듈을 떙겨오자

|  |
| --- |
| npm i |
| 현재폴더의 package.json 으로 땡겨온다 |

시작하기

|  |
| --- |
| ng serve -o  ng serve --open |
| 서버 시작 localhost:4200  -o는 바로 브라우저 열림 |

빌드하기

|  |
| --- |
| ng build |
| ng build --prod --output-path ../../ |
| ng build -op ./webapp |
| ng build -op ./webapp -w |

모듈은 여러개의 컴포넌트를 묶어서 관리해주는것

|  |  |
| --- | --- |
| main.ts | 프로그램에서 메인매서드같은것 |
|  |  |
|  |  |

--dev

디버그 모드

--prod

제품모드

ngStyle어트리뷰트 지시자

|  |
| --- |
| <p [ngStyle=”{color:’red’,backgroundCOlor:’yellow’}”>  내용  </p> |
| 컨포넌트에서 호출  <p [ngStyle=”{color:’red’,backgroundCOlor: getBgColor()}”>  export class NgStyleComponent imlements OnInit{  ...  //코드 비하인드에서 배경색 반환  getBgColor(): string{  return “yellow”;  //return (new Date).getSeconds() %2==0 “yellow”:”green”;  }  } |

모듈 생성

|  |
| --- |
| ng g m point -d |
| 모듈생성 |
|  |
| ng g m point –routing |
| 라우팅 까지 |
|  |
| ng g c point -m point/.point.module –spec false |
| 컴포넌트 생성하고 모듈에 적용까지 되라. |

서비스

|  |
| --- |
| ng g service point/point-data –spec false -m point/point.module -d |

클래스 생성

|  |
| --- |
| ng g class point/model |

인터페이스

|  |
| --- |
| export i nterface aaa{} |

라우팅쪽에 컴포넌트 등록

|  |
| --- |
| import {PointComponent} from “.pont.component;  cost routes :Routes =[  {  path :’point’  component : PointComponent,  data :{ title:”title”}  }  ] |

스트링 변수이용

|  |
| --- |
| this.http.get(‘${this.uri}/fff’) |

캐스팅

|  |
| --- |
| return <IPoint>response.json(); |

switch

|  |
| --- |
| this.ds.getPoint(111).subscribe(  (model:IPoint)=>{  },  (err)=>{...  },  ()=>{  //완료  }) |

subscribe

db->json 데이터 읽기

컴포넌트 사용 (구성요소)

|  |
| --- |
| ng generate component **name** –dry-run  ng g c name -d  같은 명령어 |
| name으로 컴포넌트 생성 될거라는 예시보여줌 -dry-run 이나 -d 빼면 생성됨 |
| --spec false -d  하면 스팩 파일은 빼고 처리됨 |
| ng g c header –spec false -d |
|  |
| ng g c footer –inline-template –spec false -d  ng g c footer –it –spec false -d |
| 컴포넌트 ts쪽에 인라인 템플릿쪽으로 html내용이 들어간다 (html생성안됨) |
| ng g c footer -it –inline-style –spec false -d  ng g c footer -it –is –spec false -d |
| 인라인 스타을로 만들어짐. |
|  |

가져다 쓰기

|  |
| --- |
| <app-componentname></app-componentname> |
| 셀렉터 이름을 태그로 넣으면된다. |

일반 경로로 처리할것인지 해쉬 값으로 처리할것인지 path를

|  |
| --- |
| **import** {Routes, RouterModule} **from** '@angular/router'; **import** {MainLayoutComponent} **from** "./shared/layout/app-layouts/main-layout.component"; **import** {AuthLayoutComponent} **from** "./shared/layout/app-layouts/auth-layout.component"; **import** {ModuleWithProviders} **from** "@angular/core";  **export const** routes: Routes = [  {  path: '',  component: MainLayoutComponent,  children: [  {  path: '', redirectTo: 'home', pathMatch: 'full'  },  {  path: 'home',  loadChildren: 'app/+home/home.module#HomeModule'  },  ]  },  ];  **export const** routing: ModuleWithProviders = RouterModule.*forRoot*(routes, {**useHash: true**}); |

pipe 만들기

|  |
| --- |
| ng g p +home\khh |
| **import** { Pipe, PipeTransform } **from** '@angular/core';  @Pipe({  name: 'khh' }) **export class** KhhPipe **implements** PipeTransform {   transform(value: **any**, args?: **any**): **any** {  **debugger**;  **return** value+"\_1111";  }  } |

흐름

|  |
| --- |
| index.html  </head> <body>  **<app-root>Loading...</app-root>**  <script type="text/javascript" src="inline.bundle.js"></script><script type="text/javascript" src="polyfills.bundle.js"></script><script type="text/javascript" src="styles.bundle.js"></script><script type="text/javascript" src="vendor.bundle.js"></script><script type="text/javascript" src="main.bundle.js"></script></body> </html> |
| main.ts  **import** './lib'  **import** { platformBrowserDynamic } **from** '@angular/platform-browser-dynamic'; **import** { enableProdMode } **from** '@angular/core'; **import** { environment } **from** './environments/environment'; **import** { AppModule } **from** './app/app.module';  **if** (environment.production) {  enableProdMode(); }  platformBrowserDynamic().bootstrapModule(AppModule); |
|  |