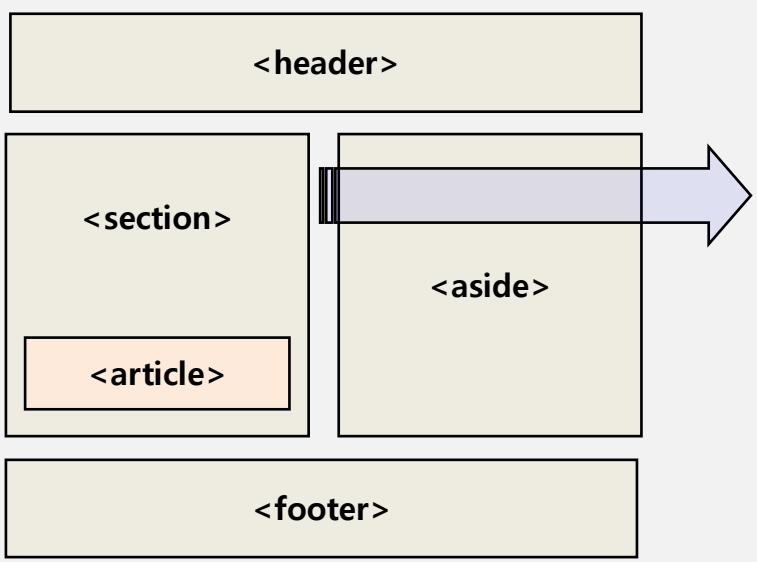
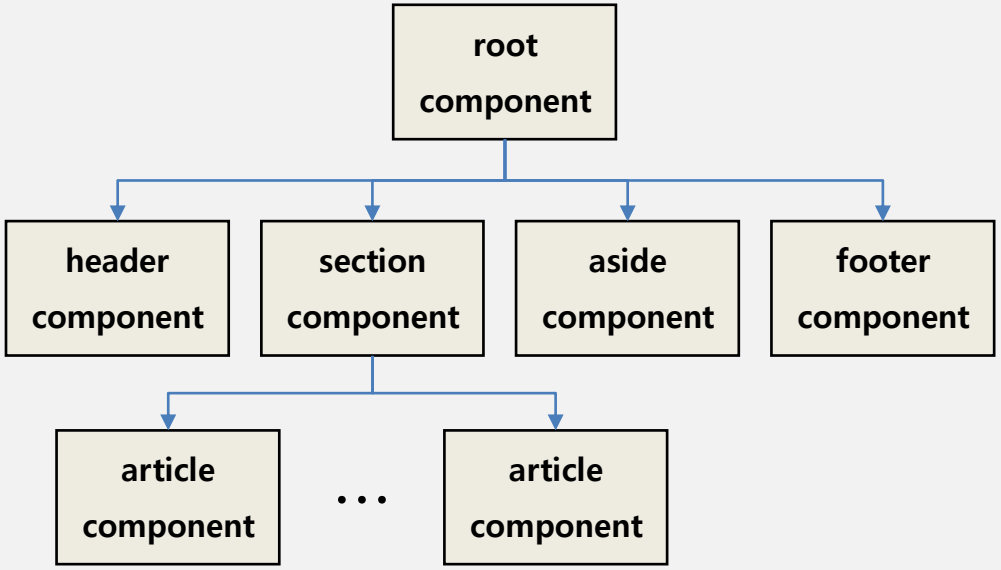
Angular with Java

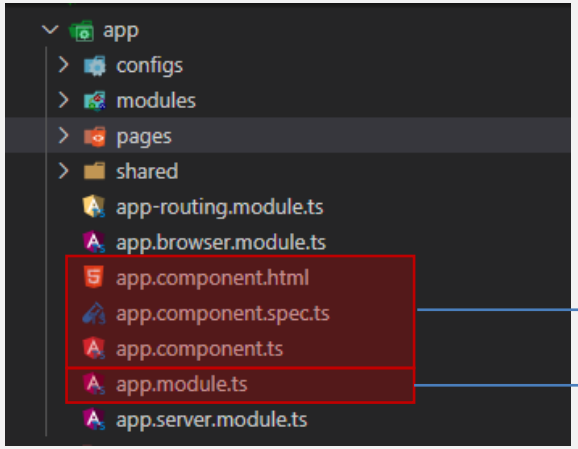
Component 구성도



Block Structure



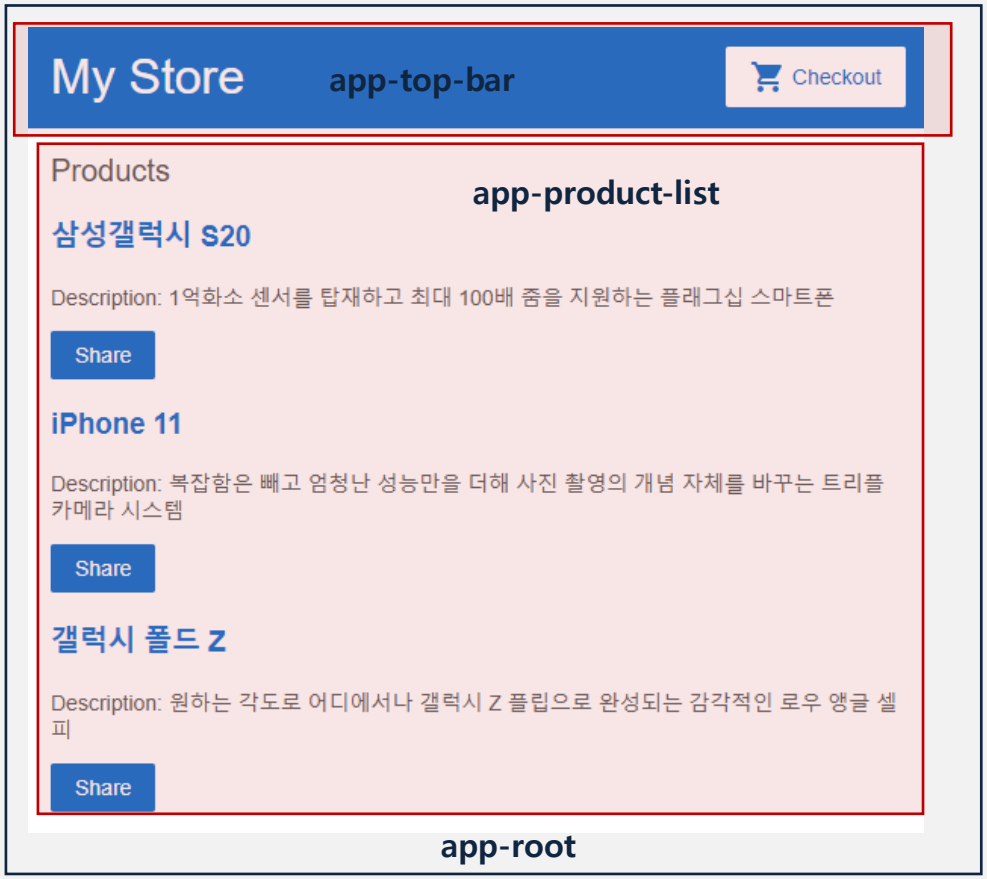
Component Tree



Root Component

Root Module

Component 구성도



Component 구성

```
You, a few seconds ago | 1 author (You)
@Component({
selector: 'app-product-alerts',
templateUrl: './product-alerts.component.html',
styleUrls: ['./product-alerts.component.css']
})
export class ProductAlertsComponent implements OnInit {

    @Input() product: any;        You, 3 days ago • commit
    @Output() notify = new EventEmitter();

    constructor() {
        // undefined
        console.log(this.product);
    }

    ngOnInit() {
        console.log(this.product);
    }

}
```

@Component Decorator는 Component의 Angular 메타 데이터를 제공합니다.

CSS 선택 자 이름인 product-list는 상위 Component의 템플릿 내에서 이 Component를 식별하는 element tag와 일치합니다.

컴포넌트 클래스는 항상 다른 곳에서 import하기 때문에 언제나 export해야 합니다.

constructor는 자바스크립트 엔진에 의해 초기화 되는데 타입스크립트에서는 Angular에 의존성이 어느 프로퍼티에 적용되는지 직접 지정 안하고도 사용할 수 있다.

constructor를 호출하는 주체가 Angular가 아닌 자바스크립트 엔진이라는 점이 중요하다.

ngOnInit은 순수하게 Angular가 컴포넌트 초기화를 완료했다는 점을 전달하기 위해 존재한다.

이 단계는 컴포넌트에 프로퍼티를 지정하고 첫 변경 감지가 되는 범위까지 포함되어 있다. @Input() 데코레이터를 사용하는 경우를 예로 들 수 있다.

@Input() 프로퍼티는 ngOnInit 내에서 접근 가능하지만 constructor에서는 undefined를 반환하는 방식으로 디자인되어 있다.

메뉴 등록

GitHub에서 material dashboard 오픈소스 가져옴. https://github.com/creativetimofficial/material-dashboard-angular2.git

npm install 후 ng serve --open 실행

app.module.ts

import { AdminCodeComponent } from './admin/code/admin-code.component'; // 코드관리 컴포넌트 추가

/src/app/components/sidebar/sidebar.component.ts

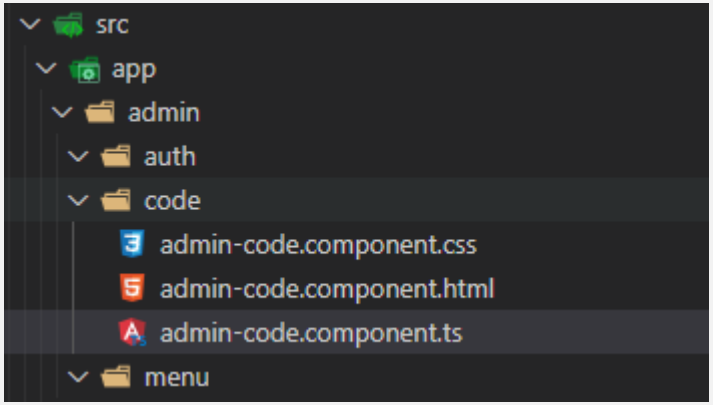
```
export const ROUTES: RouteInfo[] = [
    { path: '/dashboard', title: 'Dashboard',  icon: 'dashboard', class: '' },
    { path: '/code', title: '코드관리',  icon: 'code', class: '' },
    { path: '/user-profile', title: 'User Profile',  icon: 'person', class: '' },
    { path: '/table-list', title: 'Table List',  icon: 'content_paste', class: '' },
    { path: '/typography', title: 'Typography',  icon: 'library_books', class: '' },
    { path: '/icons', title: 'Icons',  icon: 'bubble_chart', class: '' },
    { path: '/maps', title: 'Maps',  icon: 'location_on', class: '' },
    { path: '/notifications', title: 'Notifications',  icon: 'notifications', class: '' },
    { path: '/upgrade', title: 'Upgrade to PRO',  icon: 'unarchive', class: 'active-pro' },
];
```

src/app/layouts/admin-layout/admin-layout.routing.ts

import { AdminCodeComponent } from '../../admin/code/admin-code.component';

```
    { path: 'dashboard',      component: DashboardComponent },
    { path: 'code',           component: AdminCodeComponent },
    { path: 'user-profile',   component: UserProfileComponent },
    { path: 'table-list',     component: TableListComponent },
    { path: 'typography',     component: TypographyComponent },
    { path: 'icons',          component: IconsComponent },
    { path: 'maps',           component: MapsComponent },
    { path: 'notifications',  component: NotificationsComponent },
    { path: 'upgrade',        component: UpgradeComponent },
];
```

메뉴 등록



코드관리 모듈을 위한 컴포넌트 추가

메뉴 등록

src/app/app.module.ts 에서 declarations 선언을 하지 않음

```
  declarations: [
    AppComponent,
    AdminLayoutComponent,
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

src/app/layouts/admin-layout/admin-layout.module.ts에서 사용자 모듈을 declarations 선언
list 화면에서 *ngFor 를 사용하기 위함

```
],
declarations: [
  DashboardComponent,
  UserProfileComponent,
  TableListComponent,
  TypographyComponent,
  IconsComponent,
  MapsComponent,
  NotificationsComponent,
  UpgradeComponent,
  AdminAuthComponent,
  AdminCodeComponent,
  AdminMenuComponent,
]
```

API 서비스를 위한 설정

CORSFilter.java

```
public class CORSFilter implements Filter {
.
.
 ((HttpServletResponse) servletResponse).addHeader("Access-Control-Allow-Origin", "*");
        ((HttpServletResponse) servletResponse).addHeader("Access-Control-Allow-Methods","GET, OPTIONS, HEAD, PUT, POST, DELETE");
        ((HttpServletResponse) servletResponse).setHeader("Access-Control-Max-Age", "3600");
        ((HttpServletResponse) servletResponse).setHeader("Access-Control-Allow-Headers", "x-requested-with, Authorization, origin, content-type, accept, access-control-request-method,Access-Control-Request-Headers");
.
.
}
```

WEB.xml

```
<filter>
                <filter-name>CorsFilter</filter-name>
                <filter-class>com.web.api.CORSFilter</filter-class>
</filter>
<filter-mapping>
                <filter-name>CorsFilter</filter-name>
                <url-pattern>*</url-pattern>
</filter-mapping>
```