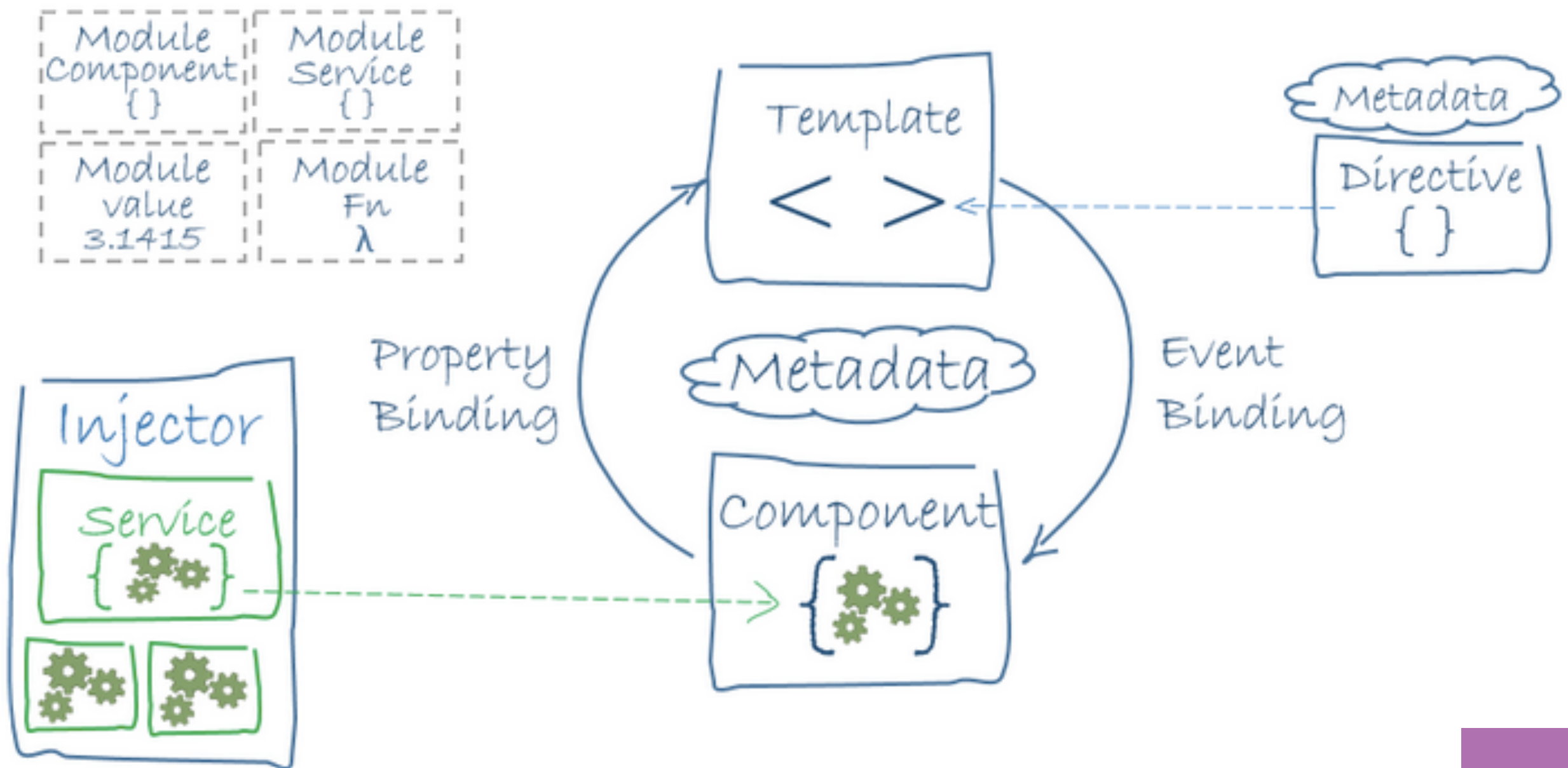


6강. ionic2의 해부

강의 순서

- ionic2 (angular2)의 요소
- ionic2 프로젝트의 구조 해부
- ionic2의 기능 해부

ionic2(angular2)의 요소



- Module
- Component
- Template
- Data Binding
- Service (Provider)

Module

프로그램의 기능을 독립적인 부품으로 분리한 것을 모듈이라고 하며, 모듈화 프로그래밍이란 이러한 분리를 강조하여 유지 보수와 타 프로그램에서의 코드 재사용을 손쉽게 하는 소프트웨어 설계 기법을 말한다. 모듈은 일반적으로 서브루틴과 데이터 구조의 집합체로서, 그 자체로서 컴파일 가능한 단위이며, 재사용 가능하고 동시에 여러 다른 모듈의 개발에 사용될 수 있다.

[네이버 지식백과] (두산백과)

Module



관제탑!

- Angular Modules help organize an application into cohesive blocks of functionality.
- An Angular Module is a class adorned with the `@NgModule` decorator function. `@NgModule` takes a metadata object that tells Angular how to compile and run module code. It identifies the module's own components, directives and pipes, making some of them public so external components can use them.

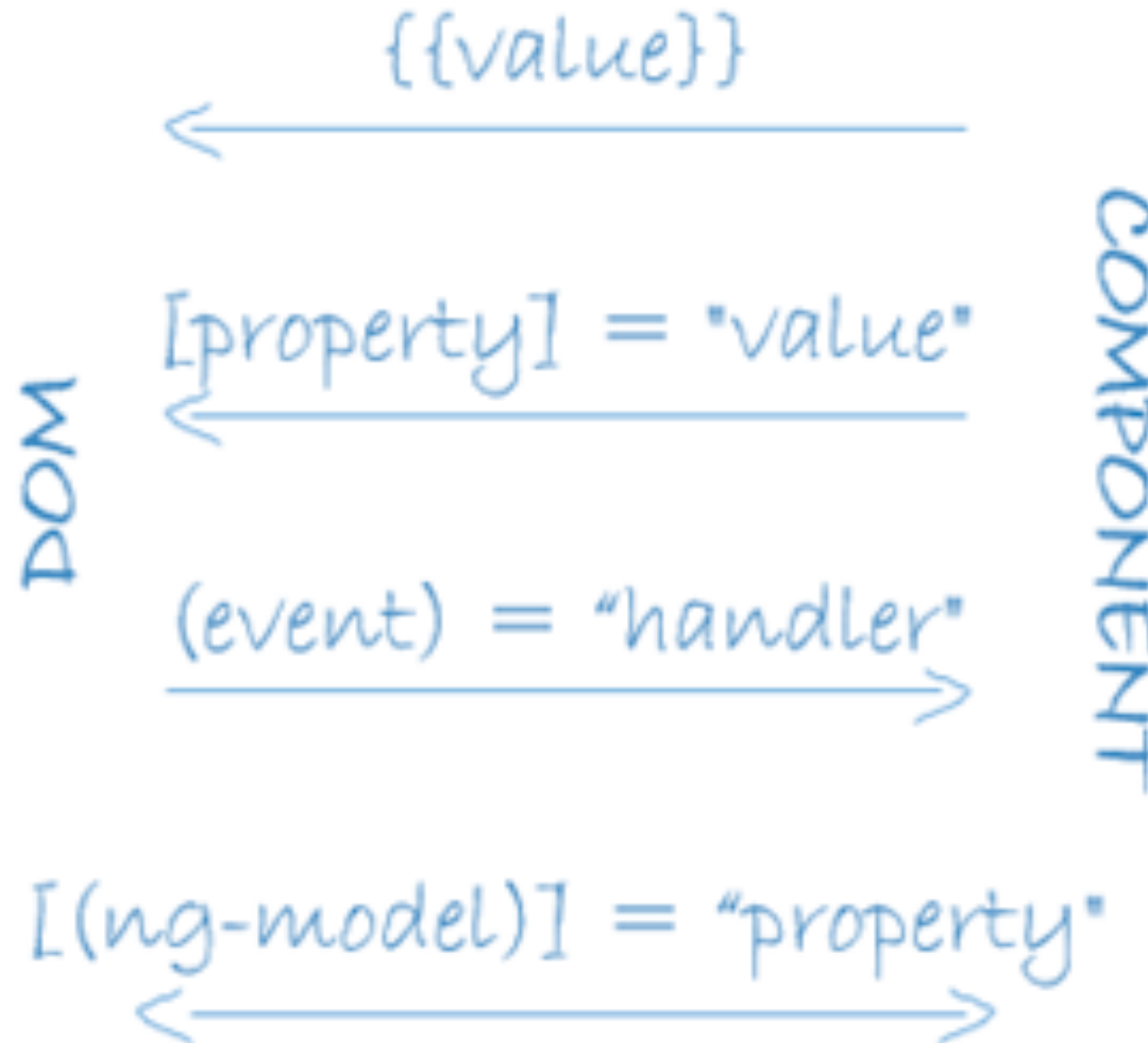
Component

- 화면(View)를 제어하는 관리자
- Component class의 property 및 method를 활용하여 화면을 제어함.
- 화면이 생성되고 지워지는 각 세부단계를 제어할 수 있음.

Template

- 스크린 화면을 표현하는 HTML 파일
- Component와 유기적으로 상호작용을 하여 사용자 친화적인 역동적인 화면을 가
능케 한다. (Data Binding을 이용하여)

Data Binding





```
<ion-header>
  <ion-navbar mddColor>
    <button ion-button icon-only
menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>{{titleText}}</ion-
title>
    <ion-buttons end>
      <button ion-button icon-only
(click)="addComment()"><ion-icon
name="add"></ion-icon></button>
    </ion-buttons>
  </ion-navbar>
</ion-header>
```

```
<ion-content>
  <ion-list>
    <ion-item text-wrap>
      <h2>포도</h2>
      <p>
        나이키의 어원은 그리스의 승리의 여신 니케
(Nίκη, Nike)에서 유래되었다고 한다.
      </p>
    </ion-item>
    ...
  </ion-list>
</ion-content>
```

모두닥

```

<ion-header>
  <ion-navbar mddColor>
    <button ion-button icon-only
menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>{{titleText}}</ion-
title>
    <ion-buttons end>
      <button ion-button icon-only
(click)="addComment()"><ion-icon
name="add"></ion-icon></button>
    </ion-buttons>
  </ion-navbar>
</ion-header>

<ion-content>
  <ion-list>
    <ion-item text-wrap>
      <h2>포도</h2>
      <p>
        나이키의 어원은 그리스의 승리의 여신 니케
(Nίκη, Nike)에서 유래되었다고 한다.
      </p>
    </ion-item>
    ...
  </ion-list>
</ion-content>

```

```

import { Component } from '@angular/core';
import { NavController, NavParams } from
'ionic-angular';

```

```

import { CommentPage } from '../..//pages/
comment/comment';

```

```

@Component({
  selector: 'page-playground',
  templateUrl: 'playground.html'
})

```

```

export class PlaygroundPage {

```

```

  titleText: string = '놀이터';

```

property
(속성)

```

  constructor(public navCtrl:
NavController, public NavParams:
NavParams) {}

```

```

  ionViewDidLoad() {

```

method
(함수)

```

    addComment() {
      this.navCtrl.push(CommentPage);
    }

```

```

  }

```

모두닥

Service(Provider)

- Service is a broad category encompassing any value, function, or feature that your application needs.
- 앱 전반에 걸쳐 사용되는 값 혹은 함수를 service 로 만들어서 유지 보수를 용이하게 하는 목적.

Variables

url = "http://modoodoc-booking.com"

CommentPage

url = "http://modoodoc.com"
this.variables.url

PlaygroundPage

url = "http://modoodoc.com"
this.variables.url

RegisterPage

url = "http://modoodoc.com"
this.variables.url

LoginPage

url = "http://modoodoc.com"
this.variables.url

ionic2 프로젝트의 구조 해부

해부학



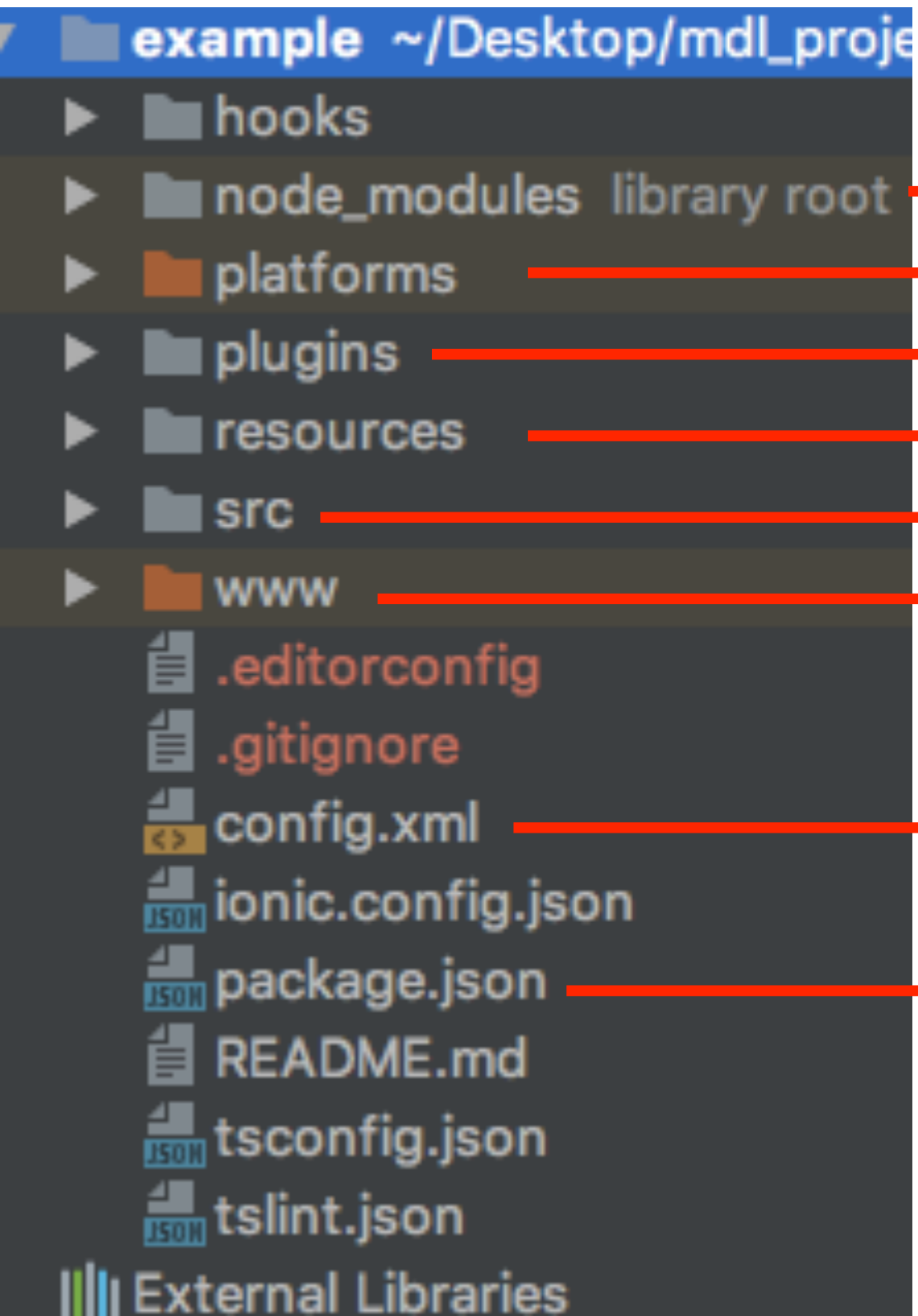
프로젝트 구조의 해부

생리학



Module 의 기능

폴더 구조의 해부



angular2, ionic2, 기타 라이브러리 모음

android, ios 플랫폼 소스코드

cordova 플러그인들

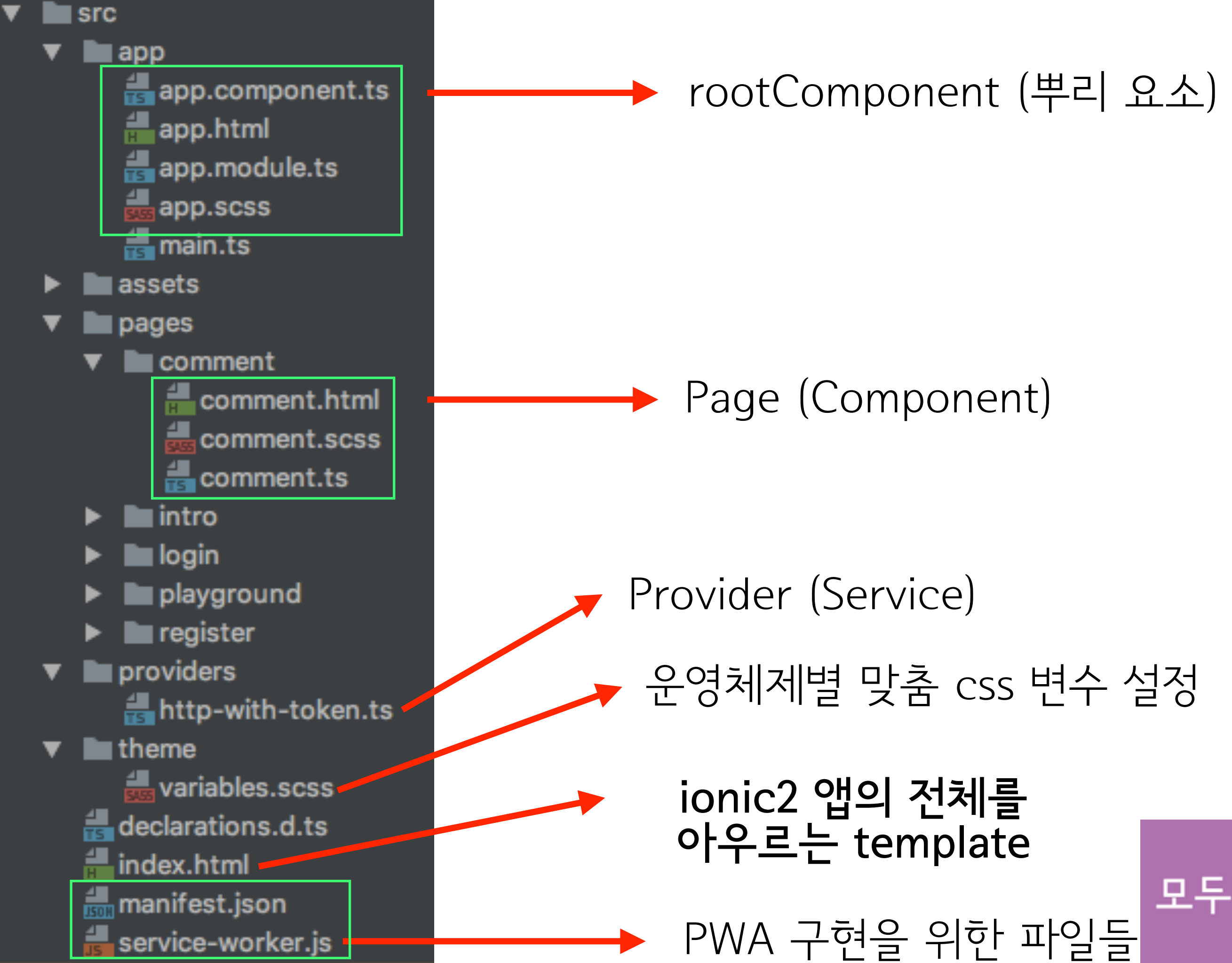
icon, splashscreen 이미지 파일

ionic2 프로젝트의 소스코드

ionic2 프로젝트의 컴파일된 코드들

모바일앱의 설정 파일

라이브러리 파일들의 명세서



```
src/index.html
```

```
<body>
```

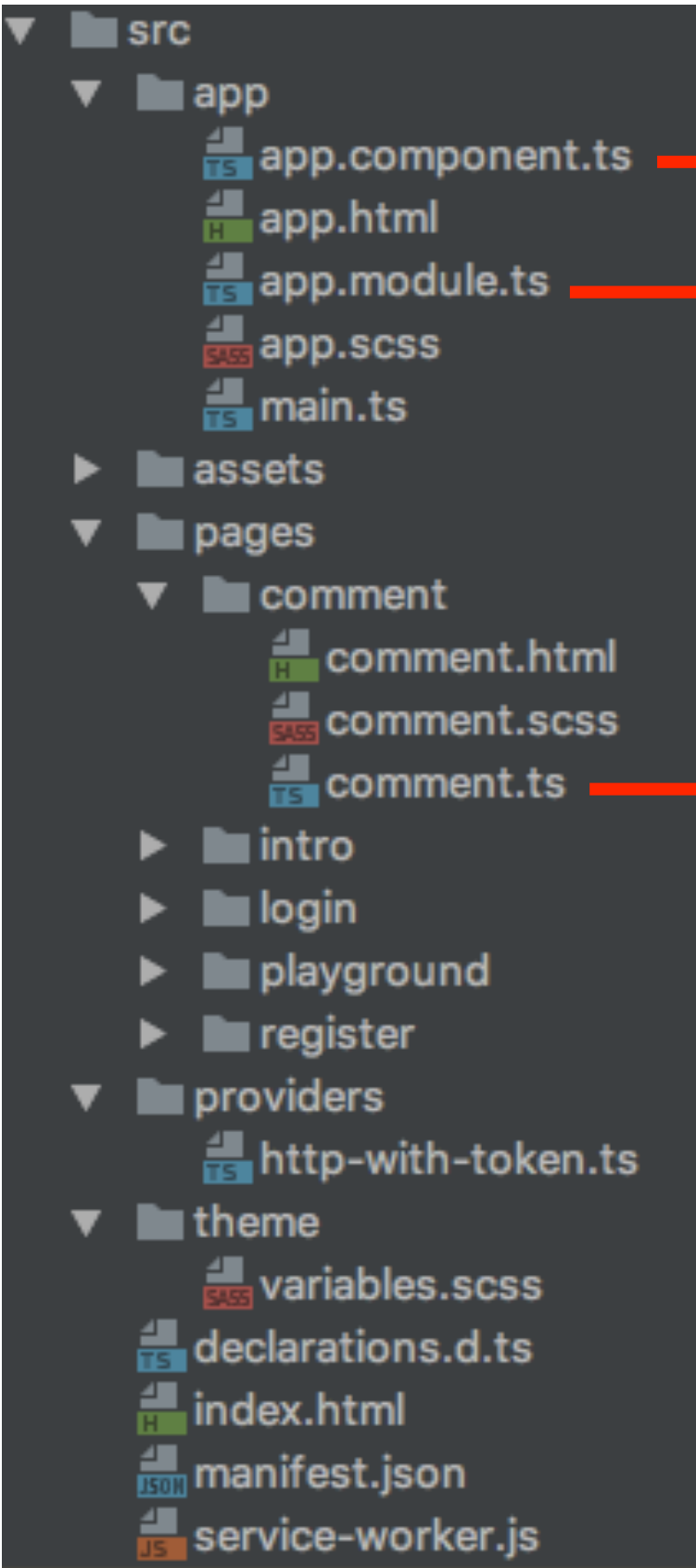
```
  <ion-app></ion-app>
```

```
  <script src="cordova.js"></script>
```

```
  <script src="build/main.js"></script>
```

```
</body>
```

ionic2의 기능 해부



MyApp (Root Component)

AppModule (Root Module)

CommentPage (Component)

AppModule

“비행기의 조종실”

MyApp

Page

“상품의 포장지”

root

Page

“얼굴”

src/app/app.module.ts

```
@NgModule({  
  declarations: [MyApp,HelloIonicPage, ItemDetailsPage, ListPage],  
  imports: [IonicModule.forRoot(MyApp)],  
  bootstrap: [IonicApp],  
  entryComponents: [MyApp,HelloIonicPage,ItemDetailsPage,ListPage],  
  providers: []  
})  
  
export class AppModule {}
```


src/app/app.component.ts

```
import {HelloIonicPage} from '../pages/hello-ionic/hello-ionic';
```

```
@Component({  
  templateUrl: 'app.html'  
})
```

```
export class MyApp {
```

```
  rootPage: any = HelloIonicPage;
```

```
  constructor(  
    public platform: Platform,  
    public menu: MenuController  
  ) {  
    ...  
  }  
}
```

side menu & tab

src/app/app.html

```
<ion-menu [content]="content">
  <ion-content>
    <ion-list>
      <ion-list no-lines>
        <ion-list-header>우리들의 놀이터</ion-list-header>
        <button ion-item (click)="openPage(rootPage)">놀이터</button>
        <button ion-item (click)="openPage(introPage)">모두닥 소개</button>
        <ion-list-header>나의 계정</ion-list-header>
        <button ion-item (click)="logout()">로그아웃</button>
      </ion-list>
    </ion-list>
  </ion-content>
</ion-menu>
<ion-nav id="nav" #content [root]="rootPage"></ion-nav>
```

src/pages/playground/playground.ts

```
import { Component } from '@angular/core';  
import { NavController, NavParams } from 'ionic-angular';
```

```
@Component({  
  selector: 'page-playground',  
  templateUrl: 'playground.html'  
})
```

```
export class PlaygroundPage {
```

```
  constructor(public navCtrl: NavController, public navParams: NavParams) {}
```

```
  ionViewDidLoad() {
```

```
}
```

```
}
```

필수적이진 않음.
페이지만의 고유의 styling을 위해 필요

연관된 html 템플릿

provider(service)를 Page에서 사용하도록 property로 설정. 로딩 과정에서 여러 속성들을 정의할 때 사용

Runs when the page has loaded.
This event only happens once per page being created.

src/pages/playground/playground.html

```
<ion-header>
  <ion-navbar>
    <button ion-button icon-only menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>놀이터</ion-title>
    <ion-buttons end>
      <button ion-button icon-only (click)="addComment()"><ion-icon name="add"></
ion-icon></button>
    </ion-buttons>
  </ion-navbar>
</ion-header>

<ion-content padding>

</ion-content>
```

src/pages/playground/playground.scss

```
page-playground {
```

```
}
```

src/app/app.scss

앱 전체에 적용되면 좋은 style들을 여기에 적는다.

src/theme/variables.scss

```
$font-path: "../assets/fonts";
```

```
@import "ionic.globals";
```

```
$colors: (  
  primary:    #387ef5,  
  secondary:  #32db64,  
  danger:     #f53d3d,  
  light:      #f4f4f4,  
  dark:       #222  
);
```

각 플랫폼에 통일적으로 적용되는 styling 변수들을 재조정할 때 사용한다.

src/providers/http-with-token.ts

```
import { Injectable } from '@angular/core';  
import { Http } from '@angular/http';  
import 'rxjs/add/operator/map';
```

```
@Injectable()
```

```
export class HttpWithToken {
```

```
  constructor(public http: Http) {  
    console.log('Hello HttpWithToken Provider');  
  }
```

```
  url = 'http://modoodoc.com';
```

```
}
```

앱 전체에 적용되면 좋을 value, function들을 여
기에 적는다.

src/providers/http-with-token.ts

```
import { Injectable } from '@angular/core';  
import { Http } from '@angular/http';  
import 'rxjs/add/operator/map';
```

```
@Injectable()
```

```
export class HttpWithToken {
```

```
  constructor(public http: Http) {  
    console.log('Hello HttpWithToken Provider');  
  }
```

```
  url = 'http://modoodoc.com';
```

```
}
```

앱 전체에 적용되면 좋을 value, function들을 여
기에 적는다.

src/pages/playground/playground.ts

```
import { Component } from '@angular/core';
import { NavController, NavParams } from 'ionic-angular';

import { CommentPage } from '../../pages/comment/comment';

import { HttpWithToken } from '../../providers/http-with-token';

@Component({
  selector: 'page-playground',
  templateUrl: 'playground.html'
})
export class PlaygroundPage {

  url: string;

  constructor(public navCtrl: NavController, public navParams: NavParams,
               public httpWithToken: HttpWithToken) {}

  ionViewDidLoad() {
    this.url = this.httpWithToken.url;
  }

  addComment() {
    this.navCtrl.push(CommentPage);
  }
}
```