

Ionic Study

Day 11

오늘 할 것들

- 화면 네비게이션 알아보기
 - 종료된 화면으로부터 파라미터 전달
 - 메뉴 레이아웃 알아보기

파라미터 전달

- pop > push
 - 사용자의 입력이 끝나면 이전 화면으로 이동.
 - 입력된 값을 이전 화면으로 보내야 할 때...
 - 콜백함수를 이용하여 결과를 전달.

파라미터 전달

- promise 패턴
 - 자바스크립트는 기본적으로 비동기 호출
 - 함수의 실행 결과가 순차적이지 못함.
 - 결과를 예측하기 어려움
 - return 값으로 Promise 객체를 사용하여 함수의 수행 결과를 반환.
 - Promise 내에는 resolve와 reject를 정의

파라미터 전달

- `git clone https://github.com/KimTibber/ionic-sample-password`
- `cd ionic-sample-password`
- `npm install`
- `ionic serve`

파라미터 전달

- src/pages/home/home.html

```
9      <ion-content padding>
10      |   <button ion-button (click)="getPassword()">Get Password</button>
11      |   <div>
12      |     {{password}}
13      |   </div>
14      </ion-content>
15
```

파라미터 전달

- src/pages/home/home.ts

```
17  callbackFunction = (_params) => {
18      return new Promise((resolve, reject) => {
19          console.log('callbackFunction:' + _params);
20          for (var i=0; i<_params.length; i++) {
21              this.password += _params[i].toString();
22          }
23          resolve();
24      });
25  }
26
27  getPassword() {
28      this.navCtrl.push(PasswordPage, {callback: this.callbackFunction});
29  }
30
```

파라미터 전달

- src/pages/password/password.ts

```
64  cancel() {  
65      console.log('cancel');  
66      this.navCtrl.pop();  
67  }  
68  
69  back() {  
70      this.navCtrl.pop();  
71  }  
72  
73  confirm() {  
74      this.callback(this.passwordInput).then(() => {  
75          this.navCtrl.pop();  
76      });  
77  }  
78
```


파라미터 전달

- src/app/app.module.ts

```
16     imports: [  
17         BrowserModule,  
18         IonicModule.forRoot(MyApp),  
19         PasswordPageModule  
20     ],
```

메뉴 레이아웃

- 현 페이지에서 이동할 수 있는 페이지들을 메뉴를 통해 사용자에게 보여줄 수 있음.
 - 햄버거라는 특수한 버튼을 통해 메뉴 노출

메뉴 레이아웃

- ionic start menu blank
 - 메뉴 프로젝트 생성
- ionic g page settingPage
 - settingPage 페이지 생성 및 프로젝트에 추가

메뉴 레이아웃

- ion-nav에 템플릿 참조 변수가 정의되어 있어야 함.
- 해당 변수의 값을 ion-menu의 content 속성으로 전달.

메뉴 레이아웃

- src/app/app.html

```
<ion-menu [content]="content">
  <ion-content>
    <button ion-button ion-left full menuClose [hidden]="checkPageHidden('SettingPage')" (click)
      ="openSettingPage()">
      <ion-icon name="settings"></ion-icon>설정
    </button>
    <button ion-button ion-left full menuClose [hidden]="checkPageHidden('HomePage')" (click)
      ="openHomePage()">
      <ion-icon name="home"></ion-icon>홈
    </button>
    <button ion-button ion-left full menuClose (click)="exit()">
      <ion-icon name="log-out"></ion-icon>종료
    </button>
  </ion-content>
</ion-menu>
<ion-nav #content [root]="rootPage"></ion-nav>
```

메뉴 레이아웃

- src/app/app.component.ts

```
import { Component } from '@angular/core';
import { Platform, App } from 'ionic-angular';
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';

import { HomePage } from '../pages/home/home';
import { SettingPage } from '../pages/setting/setting';
import { NavController } from 'ionic-angular';

@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  rootPage:any = HomePage;
  currentPage='HomePage';

  constructor(private platform: Platform, private app:App, statusBar: StatusBar, splashScreen: SplashScreen) {
    {
      platform.ready().then(() => {
        // Okay, so the platform is ready and our plugins are available.
        // Here you can do any higher level native things you might need.
        statusBar.styleDefault();
        splashScreen.hide();
      });
    }
  }

  checkPageHidden(page) {
    if (this.currentPage == page) return true;
    return false;
  }
}
```

메뉴 레이아웃

- src/app/app.component.ts

```
31  openSettingPage() {
32      console.log('openSettingPage');
33      this.app.getRootNavs()[0].setRoot(SettingPage);
34      setTimeout(() => {
35          this.currentPage = 'SettingPage';
36      }, 500);
37  }
38
39  openHomePage() {
40      console.log('openHomePage');
41      this.app.getRootNavs()[0].setRoot(HomePage);
42      setTimeout(() => {
43          this.currentPage = 'HomePage';
44      }, 500);
45  }
46
47  exit() {
48      console.log('exit');
49      this.platform.exitApp();
50  }
51  }
```

메뉴 레이아웃

- src/pages/home/home.html

```
1  <ion-header>
2    <ion-navbar>
3      <button ion-button menuToggle>
4        <ion-icon name=ios-menu></ion-icon>
5      </button>
6      <ion-title>
7        HomePage
8      </ion-title>
9    </ion-navbar>
10 </ion-header>
```


메뉴 레이아웃

- src/pages/setting/setting.html

```
<ion-header>  
  <ion-navbar>  
    <button ion-button menuToggle>  
      <ion-icon name=ios-menu></ion-icon>  
    </button>  
    <ion-title>settingPage</ion-title>  
  </ion-navbar>  
</ion-header>
```

메뉴 레이아웃

- src/app/app.module.ts

```
imports: [  
  BrowserModule,  
  IonicModule.forRoot(MyApp, {  
    menuType: 'push',  
    platforms: {  
      ios: {  
        menuType: 'overlay'  
      }  
    }  
  }  
),  
  SettingPageModule  
,  
],  
bootstrap: [IonicApp]
```

입력시 자동으로 import 추가됨

오늘은 여기까지~

See you next day!