

PHR 기반 개인맞춤형 건강관리 서비스 플랫폼 기술 고도화 및
실증서비스 개발연계지원

iOS 앱 개발 가이드

(주) 디케이아이테크놀로지

문서번호: PHRP_KEIT-DE-01

Ver : 1.1

개정 이력

개정일자	문서번호	개정 내역	작성자	확인자
2019-07-10	1.0	최초 작성	김세규	
2020-01-21	1.1	데모 앱 업데이트 사항 반영		

문서 목차

1. 앱 개발 환경 구성.....	1
1.1. 개발 환경 요구 사항.....	1
1.2. XCODE 개발 환경 설정	1
1.2.1. <i>framework</i> 추가하기.....	1
1.2.2. <i>Embedded Binaries</i> 에 <i>framework</i> 추가하기.....	2
1.2.3. 개발 환경 설정 완료.....	2
2. OAUTH 2.0 샘플소스.....	3
2.1. CLIENT ID 설정	3
2.2. 인증요청 소스코드	3
2.2.1. <i>RootViewController.swift</i>	3
2.2.2. 인증서버가 제공하는 로그인 페이지.....	4
2.3. 실행결과.....	5
3. REDWOOKHEALTH SDK 샘플소스.....	6
3.1. RESOURCE 조회	6
3.1.1. <i>PatientInfoViewController.swift</i>	6

3.2. 실행결과.....	7
3.3. RESOURCE 생성 / 수정	8
3.3.1. <i>PatientCreateViewController.swift</i>	8
3.4. RESOURCE 삭제	10
3.4.1. <i>PatientInfoViewController.swift</i>	10
3.5. 실행결과.....	11
3.5.1.....	11

1. 앱 개발 환경 구성

모바일용 SDK를 사용하여 앱을 개발하기 위한 개발 환경 요구 사항과 개발 환경 설정에 대해 설명합니다.

1.1. 개발 환경 요구 사항

모바일용 SDK를 사용하여 앱을 개발하기 위한 환경은 아래와 같습니다.

- Xcode 개발 도구
- Deployment Target : iOS 8.0 이상
- 개발언어 Swift 5.0 이하
- RedwoodHealth_SDK.framework : 제공되는 framework 라이브러리

1.2. Xcode 개발 환경 설정

제공되는 RedwoodHealth_SDK.framework 라이브러리를 추가하여 프로젝트 환경을 구성하는 방법입니다.

1.2.1. framework 추가하기

제공되는 RedwoodHealth_SDK.framework 라이브러리를 Finder에서 Project Navigator에 drag & drop하여 추가합니다.

framework를 추가하기전 프로젝트 구성



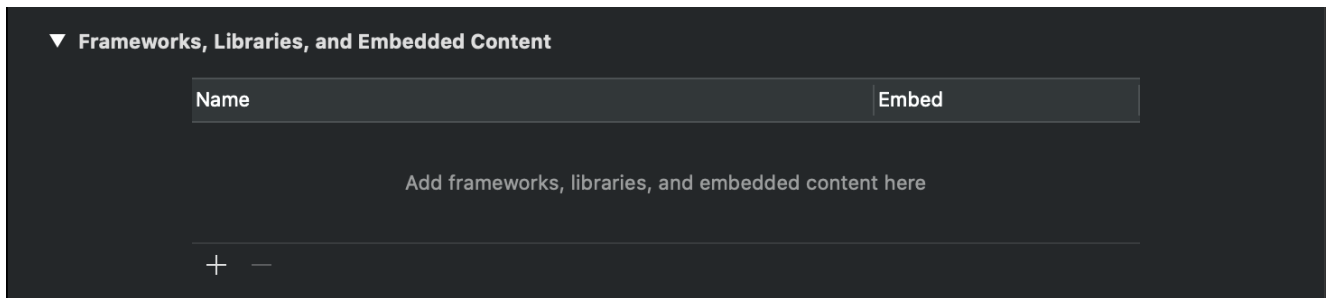
framework를 추가후 프로젝트 구성



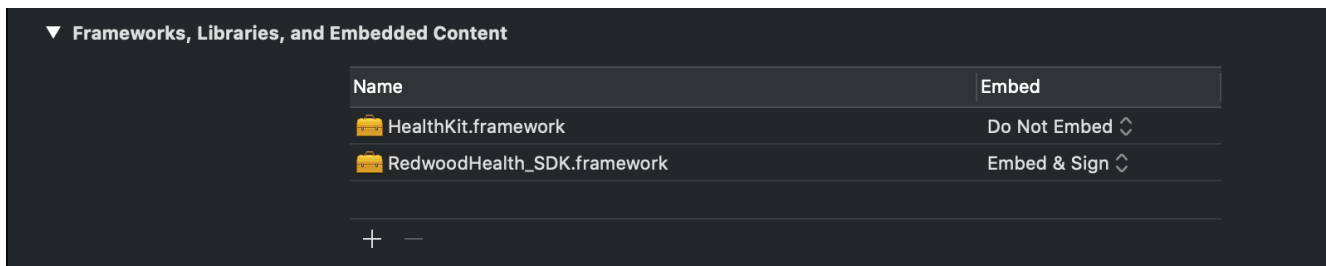
1.2.2. Embedded Binaries에 framework 추가하기

Project Navigator 에 추가한 framework 를 프로젝트 Target > General 탭의 Frameworks, Libraries, and Embedded Content 에 drag & drop 하여 추가해 줍니다. 그리고 '+' 버튼 눌러 HealthKit.framework 를 추가합니다.

framework 추가하기전 Embedded Binaries 구성.



framework 추가후 Embedded Binaries 구성.



1.2.3. 개발 환경 설정 완료

위의 과정이 마무리되면 제공되는 RedwoodHealth_SDK.framework 라이브러리를 사용할 수 있습니다.

2. OAuth 2.0 샘플소스

소스코드는 RedwoodHealth-SDK.framework 라이브러리와 함께 제공되는 테스트앱에 구현되어 있습니다.

RedwoodHealth_SDK.framework 라이브러리를 추가한 프로젝트에서 OAuth 2.0 API 을 사용하여 PHR 플랫폼의 인증서버로부터 access token 을 발급받는 샘플소스에 대한 설명입니다.

2.1. Client ID 설정

OAuth 2.0 API 를 통해서 access token 을 발급받으려면 PHR 플랫폼에 가입한 서비스정보를 설정해야 합니다.

- client ID : 서비스 가입시 입력한 client ID.
- client secret : 서비스 승인후 발급받은 client secret.
- redirect : 서비스 가입시 입력한 redirect uri.
- scope : 권한 영역.

2.2. 인증요청 소스코드

RedwoodHealth-SDK 의 PHRFhirConfig 구조체에 플랫폼에 가입한 서비스정보를 입력한 후 인증요청하는 소스코드입니다.

인증요청을 하면 Redwood 플랫폼의 인증서버에서 제공하는 로그인 페이지를 통해서 서비스에 가입한 사용자 정보를 입력하여 로그인하면 인증서버가 사용자를 확인하여 access token 을 발급하게됩니다.

2.2.1. RootViewController.swift

```
var fhirConfig = PHRFhirConfig(baseUrl: Config.fhirUrl)
fhirConfig.authorizeUri = "https://oauth.redwoodhealth.kr/oauth/authorize"
fhirConfig.clientId = "testService001"
fhirConfig.clientSecret = "781aff68-9296-403e-9884-949350ba8c8a"
fhirConfig.redirect = "http://127.0.0.1:7000/phrtest/receiveCode.html"
fhirConfig.scope = "phr.read phr.write"
fhirConfig.title = "DEMO" // webview에서 사용할 제목


// OAuth2 인증 요청
PHRManager.shared.authorize(fhirConfig: fhirConfig) { error in
    guard error == nil else {
        self.show(error: error!)
        return
    }
}
```

2.2.2. 인증서버가 제공하는 로그인 페이지

4:51

DEMO Cancel

←



REDWOOD HEALTH
DK1 / TECHNOLOGY

로그인


☐ 아이디저장

[회원가입](#) | [아이디/비밀번호 찾기](#)

4:51

DEMO Cancel

⏮



테스트서비스001
서비스가 다음 권한을 요청합니다.

개인 건강정보 조회

☒ 승인
 ☐ 거절

개인 건강정보 등록

☒ 승인
 ☐ 거절

테스트서비스001 서비스가 다음 작업을 할 수 있도록 허용하시겠습니까?

취소

허용

RedwoodHealth-SDK 에서 자동으로 WebView 를 띄워줍니다.

2.3. 실행결과

로그인 페이지에서 입력한 사용자 정보가 정상적이면 PHR 플랫폼의 인증서버가 access token 를 보내옵니다.

SDK에서는 인증서버로 받은 access token 값을 내부 keychain에 저장하여 access token 이 필요한 리소스 서버에 접근할 때 사용됩니다.

3. RedwookHealth SDK 샘플소스

소스코드는 RedwoodHealth_SDK.framework 라이브러리와 함께 제공되는 테스트앱에 구현되어 있습니다. 반드시 인증 이후에 실행해야 정상적으로 동작합니다.

예제 코드는 FHIR Resource 중 Patient 기준으로 조회, 생성, 수정, 삭제 기능을 설명합니다.

Patient 이외 다른 Resource 도 동일하게 사용할 수 있습니다.

3.1. Resource 조회

Patient 정보를 조회하는 샘플코드입니다.

3.1.1. PatientInfoViewController.swift

PHRPatient 의 read() 함수를 이용하여 Patient 정보를 전달받고, 화면에 출력합니다. 파라미터로 patient ID 가 필요합니다.

```
override func viewWillAppear(_ animated: Bool) {
    super.viewWillAppear(animated)

    addActivityIndicatorView()
    setPatientData()

    if let patientID = appDelegate?.patientID {
        // 환장정보 요청
        PHRPatient.read(patientID) { (resource, error) in
            DispatchQueue.main.async {
                self.removeActivityIndicatorView()

                if let err = error {
                    self.show(error: err)
                }

                self.patient = resource as? PHRPatient
                self.setPatientData()
            }
        }
    }
}
```

```
}
}
```

3.2. 실행결과

The screenshot displays the '환자 정보 조회' (Patient Information Search) screen. At the top, there is a status bar with the time 6:32 and signal indicators. Below the status bar, a navigation bar contains a back arrow labeled 'Back', the title '환자 정보 조회', and a 'JSON' link. The main content area lists patient details in a two-column format:

사용자 명	홍 길동
전화번호	01012345678
E-Mail	email@email.com
성별	남자
생년월일	1999-10-11
집 주소	서울시 영등포구 여의나루로 72
결혼상태	미혼
보호자 성명	홍 기원
보호자 연락처	01098765432

At the bottom of the list, there are two buttons: '수정' (Modify) and '삭제' (Delete).

3.3. Resource 생성 / 수정

3.3.1. PatientCreateViewController.swift

- Patient 생성

```
func createPatient() {
    let patient = PHRPatient()

    // 이름
    let name = PHRHumanName()
    if let familyName = familyNameTextField.text, familyName.count > 0 {
        name.family = FHIRString(familyName)
        name.text = FHIRString(familyName)
    }
    if let givenName = givenNameTextField.text, givenName.count > 0 {
        name.given = [FHIRString(givenName)]

        if let familyName = name.family?.string, familyName.count > 0 {
            name.text = FHIRString(familyName + " " + givenName)
        }
    }
    if name.family != nil || name.given != nil {
        patient.name = [name]
    } else {
        patient.name = nil
    }
    ...(중략)...
    // 환자정보 생성
    patient.createAndReturn { error in
        DispatchQueue.main.async() {
            self.removeActivityIndicatorView()
        }
    }
}
```

```

        // 에러처리
        if let error = error {
            self.show(error: error)
        }
        // 성공
        else {
            ...(중략)...
        }
    }
}

```

- Patient 수정

```

func updatePatient(_ patient: PHRPatient) {

    ...(중략)...
    addActivityIndicatorView()

    patient.update { error in
        DispatchQueue.main.async() {
            self.removeActivityIndicatorView()

            // 에러처리
            if let error = error {
                self.show(error: error)
            }
            // 성공
            else {
                ...(중략)...
            }
        }
    }
}

```

3.4. Resource 삭제

3.4.1. PatientInfoViewController.swift

```
func deletePatient() {  
    guard let patient = patient else { return }  
  
    self.addActivityIndicatorView()  
    // 삭제 요청  
    patient.delete { error in  
        DispatchQueue.main.async {  
            self.removeActivityIndicatorView()  
            // 에러처리  
            if let error = error {  
                self.show(error: error)  
            }  
            // 성공  
            else {  
                self.show(message: "완료") {  
                    self.appDelegate?.patientID = nil  
                    self.navigationController?.popToRootViewController(animated: true)  
                }  
            }  
        }  
    }  
}
```

3.5. 실행결과

3.5.1.

