

PHR 기반 개인맞춤형 건강관리 서비스 플랫폼 기술 고도화 및  
실증서비스 개발연계지원

**iOS SDK Reference**

**V 1.3**

## 승인이력

수행 역할	소속	이름	승인일	서명	비고
작성자	DKI	김세규	2019.07.29		
검토자	DKI	김현연			
총괄 PM	DKI	김현연			

## 개정 이력

개정일자	문서번호	개정 내역	작성자	확인자
2019-07-29	1.0	- 최초 작성	김세규	
2020-01-22	1.1	- 라이프로그 등록 형식에 맞춰 수정, 라이프로그 추가	김세규	
2020-02-13	1.2	- Open API 추가	김세규	
2021-03-26	1.3	- PHTHealthkitManager 생성 시 타임존 추가	김은정	

## 문서 목차

<b>1. SDK 구성</b>	<b>1</b>
<b>2. OAUTH 2.0 개요</b>	<b>2</b>
1.1 개요	2
1.2 ROLE	2
1.3 GRANT TYPE	3
1.4 브라우저를 사용하는 NATIVE APPS의 인증 흐름도	4
<b>3. PHRMANAGER</b>	<b>5</b>
3.1. REDWOODHEALTH-SDK.PHRMANAGER	5
CLASS 구성	5
3.1.1. Class <i>PHRManager</i>	5
3.1.1.1. Method shared	6
3.1.1.2. Method authorize	6
3.1.1.3. Method accessToken	7
3.1.1.4. Method accessTokenExpiry	8
3.1.1.5. Method refreshToken	9
3.1.1.6. Method isAvailableClient	9
3.1.1.7. Method logOut	10
3.1.1.6. <i>PHRManager</i> 사용예시	11
<b>4. FHIR RESOURCE CLASS와 BASE STRUCT</b>	<b>11</b>
4.1. REDWOODHEALTH-SDK.RESOURCE	12
CLASS 구성	12
4.2. REDWOODHEALTH-SDK.BASESTRUCT	13
STRUCT 및 CLASS 구성	13
4.3. RESOURCE API	15
4.3.1. Resource 생성	16
4.3.1.1. Method create	16
4.3.1.2. Method createAndReturn	16
4.3.2. Resource 조회	17
4.3.2.1. Method read	17
4.3.3. Resource 수정	19
4.3.3.1. Method update	19
4.3.4. Resource 삭제	20

4.3.4.1. Method delete.....	20
4.3.5. Resource 검색.....	21
4.3.5.1. Method search.....	21
<b>5. OPEN API.....</b>	<b>22</b>
5.1. REDWOODHEALTH-SDK.PHROPENAPIMANAGER.....	22
CLASS 구성.....	22
5.1.1. Class PHROpenAPIManager.....	22
5.1.1.1. Method shared.....	24
5.1.1.2. Method registerUserExist.....	24
5.1.1.3. Method registerIdentify.....	25
5.1.1.4. Method getStplat.....	26
5.1.1.5. Method registerUser.....	27
5.1.1.6. Method registerUserService.....	29
5.1.1.7. Method registerSearch.....	29
5.1.1.8. Method registerSearch.....	30
5.1.1.9. Method registerUserLostpw.....	31
5.1.1.10. Method registerUserCheckAuthCode.....	32
5.1.1.11. Method registerUserResetpw.....	33
5.1.1.12. Method getUser.....	34
5.1.1.13. Method userCheckpw.....	35
5.1.1.14. Method userCheckpw.....	35
5.1.1.15. Method unregisterUser.....	37
5.1.1.16. Method sharedUser.....	37
5.1.1.17. Method sharedUserList.....	38
5.1.1.18. Method removeSharedUser.....	39
5.1.1.19. Method modifySharedUsers.....	39
5.1.1.20. Method userPhrAcl.....	오류! 책갈피가 정의되어 있지 않습니다.
5.1.1.21. Method userPhrAcls.....	오류! 책갈피가 정의되어 있지 않습니다.
5.1.1.22. Method predictionCoronaryArtery.....	40
5.1.1.23. Method predictionDiabetes.....	43
5.1.1.24. Method predictionStroke.....	46
5.1.1.25. Method predictionBodyAge.....	50
5.1.1.26. Method predictionASCVD.....	51
5.1.1.27. Method predictionCHD.....	52
5.1.1.28. Method registerUserBlock.....	53
5.1.1.29. Method registerResourceBlock.....	54

5.1.1.30. Method updateUserBlock.....	55
5.1.1.31. Method getUserBlock .....	56
5.1.1.32. Method deleteUserBlock.....	56
5.1.1.33. Method getApp.....	57
<b>6. PHRHEALTHKITMANAGER.....</b>	<b>58</b>
6.1. REDWOODHEALTH-SDK.PHRHEALTHKITMANAGER.....	59
CLASS 구성 .....	59
6.1.1. Class <i>PHRHealthkitManager</i> .....	59
6.1.1.1. Method authorizeHealthKit.....	59
6.1.1.2. Method init.....	60
6.1.1.3. Method updateSampleToFHIR.....	61

## 1. SDK 구성

구성	설명
<b>PHR Manager</b>	REDWOOD 플랫폼과 연동을 위한 플랫폼의 Authorization Server로 부터 OAuth2 인증을 받아서 access token을 발급받는 패키지.
<b>FHIR Resource Class, FHIR Base Struct</b>	플랫폼의 FHIR 연동을 위한 Resource 클래스와 기본 클래스 패키지
<b>PHROpenAPIManager</b>	플랫폼에서 Open API 사용을 위한 패키지
<b>HealthKitManager</b>	HealthKit에서 라이프로그 수집을 위한 패키지

## 2. OAuth 2.0 개요

외부 서비스(3rd party)로 하여금 자신이 가지고 있는 자원에 대한 사용을 허용하기 위한 **범용인증 표준**이다. 여기에서 자원이라 함은 서비스가 가지고 있는 모든 것이다.

### 1.1 개요

외부 사이트와 인증기반의 데이터를 연동할 때 ID, Password를 넘기는 방법은 매우 위험합니다.

ID, Password는 그 사용자의 모든 권한을 얻는 것이기 때문에 ID 도용 위험이 큼니다.

그래서 ID, Password를 사용자 임시 인증을 위한 Token을 제공받기 위해 사용합니다. 그러나 이 방법이 각 서비스마다 제각각이어서 개발자들은 인증 연동을 각 서비스별로 따로 해야 했습니다. 그러다 보니 표준적인 방법이 필요했고, 그 표준 방법이 OAuth입니다.

### 1.2 Role

OAuth 2.0은 4가지 role로 구성됩니다.

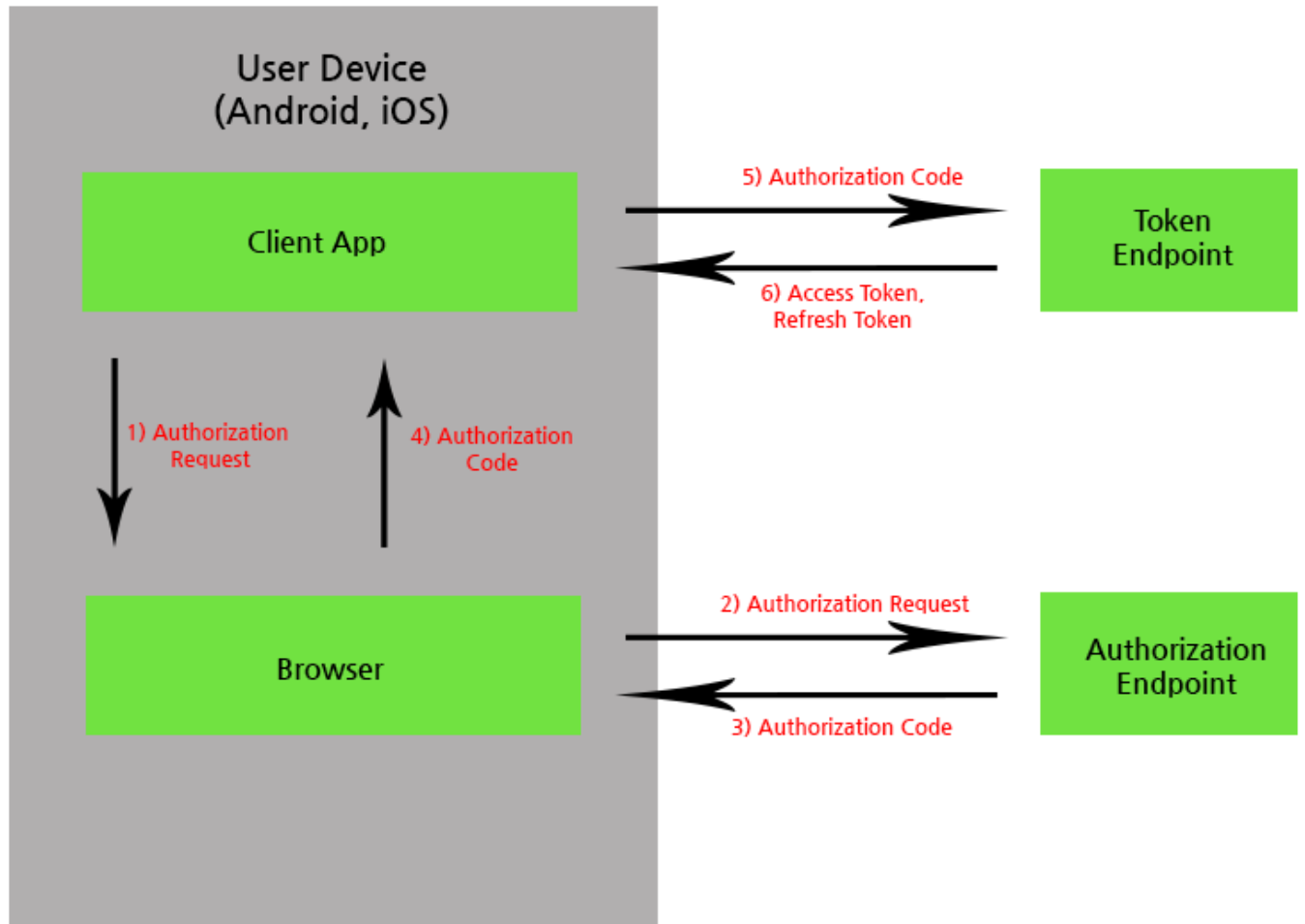
역할	설명
resource owner	1. 인증서버에 계정을 가지고 있는 개인 2. 보호된 자원에 대한 액세스 권한을 부여 할 수있는 엔티티. 리소스 소유자가 사람인 경우 리소스 소유자는 최종 사용자를 의미한다.
resource server	보호된 자원을 호스팅하는 서버를 의미하며, 액세스 토큰을 사용한 요청에 대해서만 응답합니다.
client	1. OAuth 를 사용해서 resource owner 를 대신해 resource server 에 접근하는 웹사이트나 애플리케이션. 2. 보호된 자원에 접근 권한을 가지고 있는 자원 소유자를 대신해서 보호된 리소스 요청을 하는 애플리케이션을 의미한다.
authorization server	자원 소유자의 자격을 확인하고나서 액세스 토큰을 클라이언트에게 발행하는 서버를 의미한다.



### 1.3 Grant type

Grant type	설명
Authorization Code Grant	<p>웹 서버에서 API 를 호출하는 등의 시나리오에서 Confidential Client 가 사용하는 방식이다. Public Client 인 안드로이드와 iOS 의 Native App 에서도 사용하는 방식이다.</p> <p>인증 과정에서 client_secret 이 필요하다. 로그인시에 페이지 URL 에 response_type=code 이라고 넘긴다.</p>
Implicit Grant	<p>token 과 scope 에 대한 스펙 등은 다르지만 OAuth 1.0a 과 가장 비슷한 인증방식이다.</p> <p>Public Client 인 브라우저 기반의 어플리케이션(Javascript application)이나 모바일 어플리케이션에서 이 방식을 사용하면 된다.</p> <p>Client 증명서를 사용할 필요가 없으며 실제로 OAuth 2.0 에서 가장 많이 사용되는 방식이다. 로그인시에 페이지 URL 에 response_type=token 이라고 넘긴다.</p>
Password Credentials Grant	<p>이 방식은 2-legged 방식의 인증이다.</p> <p>Client 에 아이디/패스워드를 저장해 놓고 아이디/패스워드로 직접 access token 을 받아오는 방식이다.</p> <p>Client 를 믿을 수 없을 때에는 사용하기에 위험하기 때문에 API 서비스의 공식 어플리케이션이나 믿을 수 있는 Client 에 한해서만 사용하는 것을 추천한다. 로그인시에 API 에 POST 로 grant_type=password 라고 넘긴다.</p>
Client Credentials Grant	<p>어플리케이션 이 Confidential Client 일 때 id 와 secret 을 가지고 인증하는 방식이다.</p> <p>로그인시에 API 에 POST 로 grant_type=client_credentials 라고 넘긴다.</p>

## 1.4 브라우저를 사용하는 Native Apps의 인증 흐름도



### 3. PHRManager

PHR 플랫폼과 연동을 위한 PHR 플랫폼의 Authorization Server로 부터 OAuth2 인증을 받아서 access token을 발급받는 패키지

#### 3.1. RedwoodHealth-SDK.PHRManager

PHR 플랫폼과 연동을 위한 PHR 플랫폼의 Authorization Server로 부터 OAuth2 인증을 받아서 access token을 발급받는 패키지.

Class 구성

클래스	설명
PHRManager	PHR 플랫폼의 리소스에 접근하기 위한 인증처리과정을 관리하고 제어하는 클래스.

##### 3.1.1. Class PHRManager

클래스	메소드	설명
PHRManager	shared	singleton 객체를 생성한다.
	authorize	인증서버에 로그인해서 access token을 발급 받는다.
	accessToken	인증서버에서 발급받은 access token
	accessTokenExpiry	인증서버에서 발급받은 access token의 만료일
	refreshToken	인증서버에서 발급받은 일 refresh token
	isAvailableClient	서비스 클라이언트 생성 여부를 확인한다.

클래스	메소드	설명
	logout	인증서버의 연결을 종료한다.

### 3.1.1.1. Method shared

singleton 객체를 생성하기 위한 메소드.

#### ■ Prototype

```
public class var shared: PHRManager
```

#### ■ Parameters

이름	타입	필수	설명
없음			

#### ■ Return

출력 변수	타입	설명
shared	PHRManager	PHRManager 인스턴스.

### 3.1.1.2. Method authorize

인증서버에 로그인해서 access token을 발급 받는다

#### ■ Prototype

```
public func authorize(fhirConfig: PHRFhirConfig, callback: @escaping ((_ error: Error?) -> Void))
```

## ■ Parameters

PHRFhirConfig 구조체를 선언하여 파라미터로 사용한다.

이름	타입	필수	설명
fhirBaseUrl	String	Y	플랫폼 FHIR 서버 URL
authorizeUri	String	Y	플랫폼 인증 서버 URL
clientId	String	Y	플랫폼에서 발급받은 클라이언트 아이디
clientSecret	String	Y	플랫폼에서 발급받은 클라이언트 시크릿
redirect	String	Y	플랫폼에서 발급받은 콜백 URL
scope	String	Y	권한 영역
title	String	-	로그인 화면 WebView의 제목

## ■ Return

출력 변수	타입	설명
error	FHIRError?	에러 정보. null일 경우 성공.

### 3.1.1.3. Method accessToken

인증서버에 로그인 성공시 발급받은 access token.

## ■ Prototype

```
public var accessToken: String?
```

### ■ Parameters(JSON)

이름	타입	필수	설명
없음			

### ■ Return

출력 변수	타입	설명
accessToken	String	access token

#### 3.1.1.4. Method accessTokenExpiry

인증서버에 로그인 성공시 발급받은 access token 만료일.

### ■ Prototype

```
public var accessTokenExpiry: Date?
```

### ■ Parameters(JSON)

이름	타입	필수	설명
없음			

### ■ Return

출력 변수	타입	설명
accessTokenExpiry	String	access token 만료일

### 3.1.1.5. Method refreshToken

인증서버에 로그인 성공시 발급받은 refresh token.

#### ■ Prototype

```
public var refreshToken: String?
```

#### ■ Parameters(JSON)

이름	타입	필수	설명
없음			

#### ■ Return

출력 변수	타입	설명
refreshToken	String	refresh token

### 3.1.1.6. Method isAvailableClient

서비스 클라이언트 생성 유무 확인.

#### ■ Prototype

```
public var isAvailableClient: Bool
```

#### ■ Parameters

이름	타입	필수	설명
----	----	----	----

없음

### Return

출력 변수	타입	설명
isAvailableClient	Bool	서비스 클라이언트 생성 유무.

### 3.1.1.7. Method logOut

PHR 플랫폼 서버와 연결을 종료한다.

### Prototype

```
public func logOut()
```

### Parameters

이름	타입	필수	설명
없음			

### Return

출력 변수	타입	설명
없음		



### 3.1.1.6. PHRManager 사용예시

RootViewController.swift

```
@IBAction func login(_ sender: Any?) {

    ...(중략)...

    // -----
    // 앱의 어디에서가 한 번은 설정해 주어야한다.
    // 그리고 새로운 서비스를 사용하고자 할 때도 또 다시 설정해 주어야한다. (서버설정)
    // -----

    // OAuth2 인증을 위한 설정
    var fhirConfig = PHRFhirConfig(baseUrl: "http://fhirtb.redwoodhealth.kr/fhir") // baseUrl
    fhirConfig.authorizeUri = "http://oauth.tb.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"

    // OAuth2 인증 요청
    PHRManager.shared.authorize(fhirConfig: fhirConfig) { error in
        // 실패시 에러 팝업
        if let error = error {
            ...(중략)...
        } else {
            // 인증성공
            ...(중략)...
        }
    }
}
```

## 4. FHIR Resource Class와 Base Struct

플랫폼의 FHIR(STU3) 규격의 프로파일에 대한 Resource 클래스와 기본 구조체 패키지.

## 4.1. RedwoodHealth-SDK.Resource

FHIR(STU3) 규격의 Resource 클래스 패키지. 자세한 내용은 아래 링크 내용을 참조한다.

FHIR(STU3) 규격 문서 : <http://hl7.org/fhir/STU3/resourcelist.html>

Resource 구조 설명 : [https://simplifier.net/PHRtrial/~resources?fhirVersion=STU3&sortBy=RankScore\\_desc](https://simplifier.net/PHRtrial/~resources?fhirVersion=STU3&sortBy=RankScore_desc)

- Resource 상속 구조

PHRDomainResource : 도메인 Resource 프로파일 생성

└ PHRResource : 기본 Resource 프로파일 생성

└ FHIRAbstractResource

└ FHIRAbstractBase

### Class 구성

클래스	설명
PHRAllergyIntolerance	알레르기 또는 과민증(일반적으로: 물질에 대한 부작용 위험)
PHRAppointment	특정 날짜/시간 동안 환자(patient), 시술자(practitioner), 관련자(related persons) 및 장치 간의 의료 이벤트 예약(appointment). 이로 인해 하나 이상의 인카운터(encounter)가 발생할 수 있음.
PHRCapabilityStatement	시스템 기능에 대한 설명
PHRCodeSystem	하나 이상의 코드 시스템 (code system)에서 가져온 코드 세트(set)
PHRCondition	진단에 대한 상세한 정보
PHRDiagnosticReport	진단 보고서 - 요청정보, 결과, 이미지, 해석 및 형식화된 보고서의 조합
PHRImagingStudy	단일 연구에서 생성된 이미지 (하나 이상의 참조 이미지 시리즈)
PHRImmunization	예방접종 정보

클래스	설명
PHRMedication	의약품의 정의
PHRMedicationRequest	환자 혹은 그룹을 위한 의약품 주문
PHRObservation	측정 및 간단한 검사 결과
PHROrganization	공통의 목적을 갖는 그룹이나 조직
PHRPatient	의료 서비스를 제공 받는 개인과 동물의 정보
PHRPractitioner	의료 또는 관련 서비스 제공의 공식적 책임이있는 사람
PHRProcedure	환자에게 수행 중이거나 수행된 의료 시술
PHRValueSet	코드 시스템(code system)에서 가져온 코드 세트

## 4.2. RedwoodHealth-SDK.BaseStruct

FHIR Resource 기본 타입 구조체 및 클래스 패키지

Struct 및 Class 구성

구조체 및 클래스	설명
FHIRBool	Boolean 값을 사용하는 구조체
FHIRDecimal	Decimal 값을 사용하는 구조체
FHIRInteger	32bit Integer 값을 사용하는 구조체

구조체 및 클래스	설명
FHIRString	문자열을 사용하는 구조체
FHIRURL	URL을 사용하는 구조체
PHRAddress	주소 정보 클래스
PHRAge	나이 정보 클래스
PHRAnnotation	메모, 작성자, 시기 등을 포함한 주석 정보 클래스
PHRAttachment	다른 형식으로 정의된 데이터를 첨부 파일로 포함하거나 참조하기 위한 클래스
PHRBackboneElement	PHRElement 클래스를 상속받아 내부적으로 사용하는 기본 정보를 정의한 클래스
PHRCodeableConcept	용어 또는 텍스트를 정의하는 클래스
PHRCoding	전문 용어 체계에 의해 정의된 코드에 대한 참조 클래스
PHRContactDetail	개인 또는 조직의 연락처 정보를 저장하는 클래스
PHRContactPoint	의료 기술과 관련된 개인 또는 조직의 세부적인 연락처(전화, 팩스, 전자 메일 등) 정보를 저장하는 클래스
PHRCount	갯수를 정의한 클래스
PHRDistance	길이를 정의한 클래스
PHRDomainResource	PHRResource 상속받아 내러티브, 확장 등을 정의하는 클래스
PHRDosage	환자가 복용한 약, 복용 방법이 정의된 클래스
PHRDuration	시간 길이가 정의된 클래스
PHRElement	element에 대한 기본 정의 클래스
PHRExtension	resource의 확장과 관련된 클래스, 모든 resource에 포함되어 있음

구조체 및 클래스	설명
PHRHumanName	사람 이름에 대해 정의된 클래스
PHRIdentifier	식별자 클래스
PHRMeta	resource의 메타데이터가 정의된 클래스
PHRMoney	금액이 정의된 클래스
PHRNarrative	설명 텍스트(이미지를 포함)가 정의되어 있는 클래스
PHRPeriod	시작 및 종료 날짜가 정의된 클래스
PHRQuantity	측정 양에 대한 정보가 정의된 클래스
PHRRange	상/하 값을 저장할 수 있도록 정의된 클래스
PHRRatio	분모, 분자가 정의된 클래스
PHRReference	참조할 다른 resource에 대해 정의된 클래스
PHRResource	resource에 대한 기본 정의 클래스
PHRSampledData	장비에 의해 측정된 값을 정의한 클래스
PHRSignature	디지털 서명(XML DigSig, JWT, 서명의 그래픽 이미지 등)을 정의한 클래스
PHRTiming	여러 번 발생할 수 있는 이벤트를 지정하는 타이밍 일정(예: 약 투약)을 정의하는 클래스
PHRUsageContext	적합성 또는 지식 자원에 대한 사용 컨텍스트를 정의한 클래스

### 4.3. Resource API

FHIR Resource의 API를 이용해 생성, 조회, 수정 및 삭제 기능을 실행한다. API 사용전 반드시 OAuth 인증

을 완료해야 정상적으로 동작한다.

### 4.3.1. Resource 생성

#### 4.3.1.1. Method create

리소스 객체의 정보를 이용해 서버에 리소스 생성을 요청한다.

##### ■ Prototype

```
public func create(callback: @escaping FHIRCallback)
```

##### ■ Parameters

이름	타입	필수	설명
없음			

##### ■ Return

출력 변수	타입	설명
error	FHIRCallback?	에러 정보. null일 경우 성공.

#### 4.3.1.2. Method createAndReturn

리소스 객체의 정보를 이용해 서버에 리소스 생성을 요청하고, 서버에 생성된 정보를 객체에 다시 저장한다.

##### ■ Prototype

```
func createAndReturn(callback: @escaping FHIRCallback)
```

##### ■ Parameters

이름	타입	필수	설명
없음			

## ■ Return

출력 변수	타입	설명
error	FHIRError?	에러 정보. null일 경우 성공.

## ■ 사용 예시

```

PatientCreateViewController.swift

func createPatient() {
    let patient = PHRPatient()
    ...(중략)...

    // 환자정보 생성
    patient.createAndReturn { error in
        ...(중략)...
    }
}

```

## 4.3.2. Resource 조회

### 4.3.2.1. Method read

리소스 정보를 요청한다.

## ■ Prototype

```

class func read(_ id: String, options: FHIRRequestOption = [],
               callback: @escaping FHIRResourceErrorCallback)

```

## ■ Parameters

이름	타입	필수	설명
id	String	Y	리소스 고유 ID
options	FHIRRequestOption	-	조회 옵션

## ■ Return

출력 변수	타입	설명
error	FHIRError?	에러 정보. null일 경우 성공.

## ■ 사용 예시

```

PatientInfoViewController.swift

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

    ...(중략)...

    // 환장정보 요청
    PHRPatient.read("1234") { (resource, error) in

        let patient = resource as? PHRPatient

        ...(중략)...
    }
}

```



### 4.3.3. Resource 수정

#### 4.3.3.1. Method update

수정된 리소스 객체를 서버에 반영하도록 수정을 요청한다. 리소스 객체는 서버에서 조회한 객체여야 한다.

##### ■ Prototype

```
func update(callback: @escaping FHIRCallback)
```

##### ■ Parameters

이름	타입	필수	설명
없음			

##### ■ Return

출력 변수	타입	설명
error	FHIRCallback?	에러 정보. null일 경우 성공.

##### ■ 사용 예시

```
PatientCreateViewController.swift

func updatePatient(_ patient: PHRPatient) {
    ...(중략)...

    patient.update { error in
        ...(중략)...
    }
}
```

### 4.3.4. Resource 삭제

#### 4.3.4.1. Method delete

리소스 삭제를 요청한다. 리소스 객체는 서버에서 조회한 객체여야 한다.

##### ■ Prototype

```
func delete(callback: @escaping FHIRErrorCallback)
```

##### ■ Parameters

이름	타입	필수	설명
			없음

##### ■ Return

출력 변수	타입	설명
error	FHIRError?	에러 정보. null일 경우 성공.

##### ■ 사용 예시

```
PatientInfoViewController.swift

func deletePatient() {
    guard let patient = patient else { return }

    patient.delete { error in
        ...(중략)...
    }
}
```

### 4.3.5. Resource 검색

#### 4.3.5.1. Method search

리소스 검색을 요청한다. 각 리소스별로 검색 파라미터가 다르다. 자세한 내용은 FHIR(STU) 사이트 (<http://www.hl7.org/fhir/STU3/search.html>, <http://www.hl7.org/fhir/STU3/searchparameter-registry.html>)를 참조한다.

#### ■ Prototype

```
public class func search(_ query: Any) -> FHIRSearch
```

#### ■ Parameters

이름	타입	필수	설명
query	Any	Y	검색 요청할 쿼리

#### ■ Return

출력 변수	타입	설명
error	FHIRError?	에러 정보. null일 경우 성공.

#### ■ 사용 예시

```
DiagnosticReportViewController.swift

search.perform() { bundle, error in
    if bundle.entry != nil {
        // PHRDiagnosticReport으로 타입캐스팅후 디스플레이
        let diagnosticReportList = bundle.entry
    }
}
```

## 5. Open API

플랫폼에서 Open API 사용을 위한 패키지. 일부 API 사용전 반드시 OAuth2 인증을 완료해야 정상적으로 동작한다.

### 5.1. RedwoodHealth-SDK.PHROpenAPIManager

Class 구성

클래스	설명
PHROpenAPIManager	플랫폼의 Open API 사용을 위한 클래스.

#### 5.1.1. Class PHROpenAPIManager

클래스	메소드	설명
PHRHealthkitManager	shared	singleton 객체를 생성한다.
	baseURL	Open API 기본 서버 주소
	registerUserExist	회원 아이디 중복 확인
	registerIdentify	핸드폰번호 인증(SMS)
	getStplat	플랫폼 약관 목록 조회
	registerUser	신규 회원 가입
	registerUserService	기존 회원 서비스 가입
	registerSearch	이메일/핸드폰번호로 아이디 찾기
	registerSearch	질문/답변으로 아이디 찾기
	registerUserLostpw	이메일/핸드폰번호로 비밀번호 찾기

클래스	메소드	설명
	registerUserCheckAuthCode	인증코드 확인
	registerUserResetpw	업데이트 비밀번호
	getUser	회원 정보 조회
	userCheckpw	비밀번호 확인
	updateUser	회원 정보 수정
	unregisterUser	회원 탈퇴
	sharedUser	PHR 공유자 추가
	sharedUserList	PHR 공유자 조회
	removeSharedUser	PHR 공유자 삭제
	modifySharedUsers	PHR 공유자 변경
	userPhrAcl	PHR정보제공 동의/비동의
	userPhrAcls	PHR정보제공 리스트 조회
	coronaryArteryPrediction	관상동맥질환 예측 요청
	diabetesPrediction	당뇨병 질환 예측 요청
	strokePrediction	뇌졸중 질환 예측 요청
	predictionBodyAge	신체나이 예측
	predictionASCVD	동맥경화성 심혈관질환(ASCVD) 예측
	predictionCHD	관상동맥질환(CHD) 예측
	registerUserBlock	회원 정보 Block 생성
	registerResourceBlock	Resource Block 추가
	updateUserBlock	회원 정보 Block 수정
	getUserBlock	회원 정보 Block 조회

클래스	메소드	설명
	deleteUserBlock	회원 정보 Block 삭제
	getApp	모바일 앱 최신 버전정보 조회

#### 5.1.1.1. Method shared

singleton 객체를 생성하기 위한 메소드.

##### ■ Prototype

```
public class var shared: PHROpenAPIManager
```

##### ■ Parameters

이름	타입	필수	설명
없음			

##### ■ Return

출력 변수	타입	설명
shared	PHROpenAPIManager	PHROpenAPIManager 인스턴스.

#### 5.1.1.2. Method registerUserExist

회원 아이디 중복 확인

##### ■ Prototype

```
public func registerUserExist(clientId: String, clientSecret: String, userId: String, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

### ■ Parameters

이름	타입	필수	설명
clientId	String	Y	플랫폼에서 발급받은 클라이언트 아이디
clientSecret	String	Y	플랫폼에서 발급받은 클라이언트 시크릿
userId	String	Y	사용자 아이디

### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	요청한 사용자 아이디
exist	Bool	아이디 중복여부

#### 5.1.1.3. Method registerIdentify

핸드폰 번호 인증(SMS)

### ■ Prototype

```
public func registerIdentify(clientId: String, clientSecret: String, mobilePhone: String, completion:
@escaping (Error?, Dictionary<String, Any>?) -> Void)
```

### ■ Parameters

이름	타입	필수	설명
clientId	String	Y	플랫폼에서 발급받은 클라이언트 아이디
clientSecret	String	Y	플랫폼에서 발급받은 클라이언트 시크릿
mobilePhone	String	Y	사용자 휴대폰 번호

### Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
smsSendResult	String	SMS 발송 결과 (success/failure)
authCode	String	인증번호
userId	String	플랫폼 사용자 아이디 (미가입자 일 경우 "")
isPlatformUser	Bool	플랫폼 가입 여부
IsServiceUser	Bool	서비스 가입 여부

#### 5.1.1.4. Method getStplat

플랫폼 약관 목록 조회

### Prototype

```
public func getStplat(clientId: String, clientSecret: String, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

### Parameters

이름	타입	필수	설명
clientId	String	Y	플랫폼에서 발급받은 클라이언트 아이디
clientSecret	String	Y	플랫폼에서 발급받은 클라이언트 시크릿

### Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
stplat	Array	약관 목록
- stplatSn	Int	약관 일련번호
- stplatNm	String	약관 명



- stplatCn	String	약관 내용
- essntlAt	String	필수 여부. 회원 가입 시 반드시 동의되어야 함
- creatDt	String	생성 일시(yyyy-MM-dd)

### 5.1.1.5. Method registerUser

신규 회원 가입

#### ■ Prototype

```
public func registerUser(clientId: String, clientSecret: String, userId: String, userName: String,
    password: String, birthday: String? = nil, weddingAnniversary: String? = nil, gender: String,
    email: String, mobilePhone: String, cablePhone: String? = nil, zip: String? = nil, address: String? = nil,
    detailAddress: String? = nil, idLostQuestion: String, idLostAnswer: String,
    platformStplat: [Dictionary<String, Any>],
    completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

#### ■ Parameters

이름	타입	필수	설명
clientId	String	Y	플랫폼에서 발급받은 클라이언트 아이디
clientSecret	String	Y	플랫폼에서 발급받은 클라이언트 시크릿
userId	String	Y	서비스 사용자 아이디
userName	String	Y	서비스 사용자 명
password	String	Y	서비스 사용자 패스워드(sha256 인코딩)
birthday	String	Y	생년월일 ( YYYY-MM-DD)
weddingAnniversary	String	Y	결혼기념일 ( YYYY-MM-DD)
gender	String	Y	성별 ( M, F )
email	String	Y	사용자 email 주소

mobilePhone	String	Y	사용자 핸드폰
cablePhone	String	-	사용자 전화번호
zip	String	-	사용자 우편번호
address	String	-	사용자 주소 (시, 군, 구, 동 까지)
detailAddress	String	-	사용자 상세 주소
idLostQuestion	String	Y	아이디 찾기 질문
idLostAnswer	String	Y	아이디 찾기 답변
platformStplat	Array	Y	플랫폼 이용약관 동의 내역
- stplatSn	Int	Y	플랫폼 이용약관 일련번호
- agreeYn	String	Y	동의 여부 ('Y' 또는 'N')

### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
result	String	결과(success/failure)
serviceld	String	클라이언트 아이디
userId	String	서비스 사용자 아이디
userName	String	서비스 사용자 명
birthday	String	생년월일 ( YYYY-MM-DD )
weddingAnniversary	String	결혼기념일 ( YYYY-MM-DD )
gender	String	성별 ( M, F )
email	String	사용자 email 주소
mobilePhone	String	사용자 핸드폰
cablePhone	String	사용자 전화번호
zip	String	사용자 우편번호
address	String	사용자 주소 (시, 군, 구, 동 까지)
detailAddress	String	사용자 상세 주소
idLostQuestion	String	아이디 찾기 질문
idLostAnswer	String	아이디 찾기 답변

### 5.1.1.6. Method registerUserService

기존 회원 서비스 가입

#### ■ Prototype

```
public func registerUserService(clientId: String, clientSecret: String, userId: String, completion:
@escaping (Error?, Dictionary<String, Any>?) -> Void)
```

#### ■ Parameters

이름	타입	필수	설명
clientId	String	Y	플랫폼에서 발급받은 클라이언트 아이디
clientSecret	String	Y	플랫폼에서 발급받은 클라이언트 시크릿
userId	String	Y	서비스 사용자 아이디

#### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
serviceld	String	클라이언트 아이디
userId	String	서비스 사용자 아이디

### 5.1.1.7. Method registerSearch

이메일/핸드폰번호로 아이디 찾기

#### ■ Prototype

```
public func registerSearch(clientId: String, clientSecret: String, userName: String, type: String, email:
```

```
String? = nil, mobilePhone: String? = nil, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

## ■ Parameters

이름	타입	필수	설명
clientId	String	Y	플랫폼에서 발급받은 클라이언트 아이디
clientSecret	String	Y	플랫폼에서 발급받은 클라이언트 시크릿
userName	String	Y	요청한 사용자 이름
type	String	Y	검색타입 (값은 "email" 또는 "mobile" 선택)
email	String	-	사용자 이메일 (이메일, 핸드폰 중 하나는 필수)
mobilePhone	String	-	사용자 핸드폰 (이메일, 핸드폰 중 하나는 필수)

## ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
userName	String	사용자 이름
result	String	검색결과(success/failure)

### 5.1.1.8. Method registerSearch

질문/답변으로 아이디 찾기

## ■ Prototype

```
public func registerSearch(clientId: String, clientSecret: String, userName: String, idLostQuestion: String, idLostAnswer: String?, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

### ■ Parameters

이름	타입	필수	설명
clientId	String	Y	플랫폼에서 발급받은 클라이언트 아이디
clientSecret	String	Y	플랫폼에서 발급받은 클라이언트 시크릿
userName	String	Y	요청한 사용자 이름
idLostQuestion	String	Y	아이디 분실 질문
idLostAnswer	String	Y	아이디 분실 답변

### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
userName	String	사용자 이름
result	String	검색결과(success/failure)

#### 5.1.1.9. Method registerUserLostpw

이메일/핸드폰번호로 비밀번호 찾기

### ■ Prototype

```
public func registerUserLostpw(clientId: String, clientSecret: String, userId: String, type: String, email: String?, mobilePhone: String?, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

### ■ Parameters

이름	타입	필수	설명
----	----	----	----

clientId	String	Y	플랫폼에서 발급받은 클라이언트 아이디
clientSecret	String	Y	플랫폼에서 발급받은 클라이언트 시크릿
userId	String	Y	요청한 사용자 이름
type	String	Y	검색타입 (값은 "email" 또는 "mobile" 선택)
email	String	-	사용자 이메일 (이메일, 핸드폰 중 하나는 필수)
mobilePhone	String	-	사용자 핸드폰 (이메일, 핸드폰 중 하나는 필수)

### Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
result	String	검색결과(success/failure)

#### 5.1.1.10. Method registerUserCheckAuthCode

인증코드 확인

### Prototype

```
public func registerUserCheckAuthCode(clientId: String, clientSecret: String, userId: String, authCode: String, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

### Parameters

이름	타입	필수	설명
clientId	String	Y	플랫폼에서 발급받은 클라이언트 아이디
clientSecret	String	Y	플랫폼에서 발급받은 클라이언트 시크릿

userId	String	Y	요청한 사용자 아이디
authCode	String	Y	이메일 또는 SMS로 발송된 인증코드

### Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
result	String	처리 결과(success/failure)

#### 5.1.1.11. Method registerUserResetpw

업데이트 패스워드

### Prototype

```
public func registerUserResetpw(clientId: String, clientSecret: String, userId: String, authCode: String, password: String, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

### Parameters

이름	타입	필수	설명
clientId	String	Y	플랫폼에서 발급받은 클라이언트 아이디
clientSecret	String	Y	플랫폼에서 발급받은 클라이언트 시크릿
authCode	String	Y	이메일 또는 SMS로 발송된 인증코드
password	String	Y	신규 패스워드(sha256 인코딩)

### Return(JSON<String, Any>)

출력 변수	타입	설명
-------	----	----

error	Error	HTTP 통신 에러
userId	String	사용자 아이디
result	String	처리 결과(success/failure)

### 5.1.1.12. Method getUser

회원 정보 조회

#### ■ Prototype

```
public func getUser(completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

#### ■ Parameters

이름	타입	필수	설명
없음			

#### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
username	String	처리 결과(success/failure)
birthday	String	생년월일 ( YYYY-MM-DD)
weddingAnniversary	String	결혼기념일 ( YYYY-MM-DD)
gender	String	성별 ( M, F )
email	String	사용자 email 주소
mobilePhone	String	사용자 핸드폰
cablePhone	String	사용자 전화번호
zip	String	사용자 우편번호
address	String	사용자 주소 (시, 군, 구, 동 까지)
detailAddress	String	사용자 상세 주소
idLostQuestion	String	아이디 찾기 질문



idLostAnswer	String	아이디 찾기 답변
--------------	--------	-----------

### 5.1.1.13. Method userCheckpw

패스워드 확인

#### ■ Prototype

```
public func userCheckpw(password: String, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

#### ■ Parameters

이름	타입	필수	설명
password	String	Y	서비스 사용자 패스워드(sha256 암호화)

#### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
result	String	처리 결과(success/failure)

### 5.1.1.14. Method updateUser

회원 정보 수정

#### ■ Prototype

```
public func updateUser(userName: String, password: String? = nil, birthday: String? = nil, weddingAnniversary: String? = nil, gender: String, email: String, mobilePhone: String,
```

```
cablePhone: String? = nil, zip: String? = nil, address: String? = nil, detailAddress: String? = nil,
idLostQuestion: String, idLostAnswer: String,
completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

## ■ Parameters

이름	타입	필수	설명
userName	String	Y	사용자 명
password	String	-	사용자 패스워드(sha256 암호화)
birthday	String	-	생년월일 ( YYYY-MM-DD )
weddingAnniversary	String	-	결혼기념일 ( YYYY-MM-DD )
gender	String	Y	성별 ( M, F )
email	String	Y	사용자 email 주소
mobilePhone	String	Y	사용자 핸드폰
cablePhone	String	-	사용자 전화번호
zip	String	-	사용자 우편번호
Address	String	-	사용자 주소 (시, 군, 구, 동 까지)
detailAddress	String	-	사용자 상세 주소
idLostQuestion	String	Y	아이디 찾기 질문
idLostAnswer	String	Y	아이디 찾기 답변

## ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
result	String	처리 결과(success/failure)

### 5.1.1.15. Method unregisterUser

회원 탈퇴

#### ■ Prototype

```
public func unregisterUser(serviceld: String, reason: String, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

#### ■ Parameters

이름	타입	필수	설명
serviceld	String	Y	플랫폼에서 발급받은 클라이언트 아이디
reason	String	Y	해지 사유

#### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
result	String	처리 결과(success/failure)

### 5.1.1.16. Method sharedUser

PHR 공유자 추가

#### ■ Prototype

```
public func sharedUser(sharedUserId: String, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

## ■ Parameters

이름	타입	필수	설명
sharedUserId	String	Y	추가할 공유 사용자 아이디

## ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
result	String	처리 결과(success/failure)

### 5.1.1.17. Method sharedUserList

PHR 공유자 조회

## ■ Prototype

```
public func sharedUserList(completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

## ■ Parameters

이름	타입	필수	설명
없음			

## ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
result	String	처리 결과(success/failure)
sharedUserList	Array	공유 사용자 아이디 목록
- sharedUserId	String	공유 사용자 아이디

- regDate	String	등록 일시 ( YYYY-MM-DD hh:mm:ss )
-----------	--------	-------------------------------

#### 5.1.1.18. Method removeSharedUser

PHR 공유자 삭제

##### ■ Prototype

```
public func removeSharedUser(sharedUserId: String, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

##### ■ Parameters

이름	타입	필수	설명
sharedUserId	String	Y	삭제할 공유 사용자 아이디

##### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
result	String	처리 결과(success/failure)

#### 5.1.1.19. Method modifySharedUsers

PHR 공유자 변경

##### ■ Prototype

```
public func modifySharedUsers(sharedUserIds: [String],
```

```
completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

## ■ Parameters

이름	타입	필수	설명
sharedUserIds	Array	Y	공유 사용자 아이디 목록

## ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
result	String	처리 결과(success/failure)

### 5.1.1.20. Method predictionCoronaryArtery

관상동맥질환 예측 요청

## ■ Prototype

```
public func predictionCoronaryArtery(param1: Double, param2: Double, param3: Double,
    param4: Double, param5: Double, param6: Double, param7: Double, param8: Double, param9: Double,
    param10: Double, param11: Double, param12: Double, param13: Double, param14: Double,
    param15: Double, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

## ■ Parameters

이름	타입	필수	설명
params(1)	Double	Y	성별
params(2)	Double	Y	나이

params(3)	Double	Y	관상동맥질환 가족력 관상동맥질환에 진단을 받은 부모, 형제, 자매 0 : 아니요 또는 모름, 1 : 예
params(4)	Double	Y	당뇨병 진단 여부 의사에게 당뇨병 진단을 받았는지 여부 0 : 아니요 또는 모름, 1 : 예
params(5)	Double	Y	체질량지수
params(6)	Double	Y	허리둘레 1. 76미만(~29inch) 2. 76이상 84미만(30~32inch) 3. 84이상 90미만(33~34inch) 4. 90이상(35inch~)
params(7)	Double	Y	수축기 혈압 1. 120미만(정상) 2. 120이상 140미만(고혈압전단계) 3. 140이상 160미만(고혈압 1기) 4. 160이상(고혈압 2기)
params(8)	Double	Y	이완기 혈압 1. 80미만(정상) 2. 80이상 90미만(고혈압전단계) 3. 90이상 100미만(고혈압 1기) 4. 100이상(고혈압 2기)
params(9)	Double	Y	총콜레스테롤 1. 200미만(정상) 2. 200이상 240미만(고지혈증 전단계) 3. 240이상 280미만(고지혈증 1기) 4. 280이상(고지혈증 2기)
params(10)	Double	Y	중성지방 1. 150미만(정상)

			2. 150이상 200미만(경계) 3. 200이상 400미만(높음) 4. 400이상(매우 높음)
params(11)	Double	Y	흡연 여부 0. 모름 1. 흡연 2. 금연 3. 비흡연
params(12)	Double	Y	1년간 음주빈도 0. 술을 마셔 본 적 없음 1. 최근 1년간 전혀 마시지 않았다 2. 월 1회미만 3. 월 1회정도 4. 월 2-4회 5. 주 2-3회정도 6. 주 4회이상
params(13)	Double	Y	평소 스트레스 인지 정도 0. 모름 1. 대단히 많이 느낀다 2. 많이 느끼는 편이다 3. 조금 느끼는 편이다 4. 거의 느끼지 않는다
params(14)	Double	Y	1주일간 걷기 일수 0. 모름 1. 전혀 하지 않음 2. 1일 3. 2일 4. 3일 5. 4일 6. 5일



			7. 6일 8. 7일(매일)
params(15)	Double	Y	1주일간 근력운동 일수 0. 모름 1. 최근 7일 동안 없다 2. 1일 3. 2일 4. 3일 5. 4일 6. 5일이상

### Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
modelId	String	관상동맥질환 예측모델 아이디
rate	Double	관상동맥질환 발병 확률

#### 5.1.1.21. Method predictionDiabetes

당뇨병 질환 예측 요청

### Prototype

```
public func predictionCoronaryArtery(param1: Double, param2: Double, param3: Double,
  param4: Double, param5: Double, param6: Double, param7: Double, param8: Double, param9: Double,
  param10: Double, param11: Double, param12: Double, param13: Double, param14: Double,
  param15: Double, param16: Double, completion: @escaping (Error?, Dictionary<String, Any>?) ->
  Void)
```

## Parameters

이름	타입	필수	설명
params(1)	Double	Y	성별
params(2)	Double	Y	나이
params(3)	Double	Y	당뇨병 가족력 당뇨병 진단을 받은 부모, 형제, 자매 0 : 아니요 또는 모름, 1 : 예
params(4)	Double	Y	신장(키)
params(5)	Double	Y	체중(몸무게)
params(6)	Double	Y	체질량지수
params(7)	Double	Y	허리둘레 1. 76미만(~29inch) 2. 76이상 84미만(30~32inch) 3. 84이상 90미만(33~34inch) 4. 90이상(35inch~)
params(8)	Double	Y	수축기 혈압 1. 120미만(정상) 2. 120이상 140미만(고혈압전단계) 3. 140이상 160미만(고혈압 1기) 4. 160이상(고혈압 2기)
params(9)	Double	Y	이완기 혈압 1. 80미만(정상) 2. 80이상 90미만(고혈압전단계) 3. 90이상 100미만(고혈압 1기) 4. 100이상(고혈압 2기)
params(10)	Double	Y	총콜레스테롤 1. 200미만(정상)

			2. 200이상 240미만(고지혈증 전단계) 3. 240이상 280미만(고지혈증 1기) 4. 280이상(고지혈증 2기)
params(11)	Double	Y	중성지방 1. 150미만(정상) 2. 150이상 200미만(경계) 3. 200이상 400미만(높음) 4. 400이상(매우 높음)
params(12)	Double	Y	흡연 여부 0. 모름 1. 흡연 2. 금연 3. 비흡연
params(13)	Double	Y	1년간 음주빈도 0. 술을 마셔 본 적 없음 1. 최근 1년간 전혀 마시지 않았다 2. 월 1회미만 3. 월 1회정도 4. 월 2-4회 5. 주 2-3회정도 6. 주 4회이상
params(14)	Double	Y	평소 스트레스 인지 정도 0. 모름 1. 대단히 많이 느낀다 2. 많이 느끼는 편이다 3. 조금 느끼는 편이다 4. 거의 느끼지 않는다
params(15)	Double	Y	1주일간 걷기 일수 0. 모름

			<ol style="list-style-type: none"> <li>1. 전혀 하지 않음</li> <li>2. 1일</li> <li>3. 2일</li> <li>4. 3일</li> <li>5. 4일</li> <li>6. 5일</li> <li>7. 6일</li> <li>8. 7일(매일)</li> </ol>
params(16)	Double	Y	<p>1주일간 근력운동 일수</p> <ol style="list-style-type: none"> <li>0. 모름</li> <li>1. 최근 7일 동안 없다</li> <li>2. 1일</li> <li>3. 2일</li> <li>4. 3일</li> <li>5. 4일</li> <li>6. 5일이상</li> </ol>

#### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
modelId	String	당뇨병 질환 예측모델 아이디
rate	Double	당뇨병 질환 발병 확률

#### 5.1.1.22. Method predictionStroke

뇌졸중 질환 예측 요청

#### ■ Prototype

```
public func predictionCoronaryArtery(param1: Double, param2: Double, param3: Double,
    param4: Double, param5: Double, param6: Double, param7: Double, param8: Double, param9: Double,
    param10: Double, param11: Double, param12: Double, param13: Double, param14: Double,
    param15: Double, param16: Double, param17: Double,
    completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

## ■ Parameters

이름	타입	필수	설명
params(1)	Double	Y	성별
params(2)	Double	Y	나이
params(3)	Double	Y	뇌졸중 가족력 뇌졸중 질환에 진단을 받은 부모, 형제, 자매 0 : 아니요 또는 모름, 1 : 예
params(4)	Double	Y	당뇨병 진단 여부 의사에게 당뇨병 진단을 받았는지 여부 0 : 아니요 또는 모름, 1 : 예
params(5)	Double	Y	신장(키)
params(6)	Double	Y	체중(몸무게)
params(7)	Double	Y	체질량지수
params(8)	Double	Y	허리둘레 1. 76미만(~29inch) 2. 76이상 84미만(30~32inch) 3. 84이상 90미만(33~34inch) 4. 90이상(35inch~)
params(9)	Double	Y	수축기 혈압 1. 120미만(정상) 2. 120이상 140미만(고혈압전단계) 3. 140이상 160미만(고혈압 1기)

			4. 160이상(고혈압 2기)
params(10)	Double	Y	이완기 혈압 1. 80미만(정상) 2. 80이상 90미만(고혈압전단계) 3. 90이상 100미만(고혈압 1기) 4. 100이상(고혈압 2기)
params(11)	Double	Y	총콜레스테롤 1. 200미만(정상) 2. 200이상 240미만(고지혈증 전단계) 3. 240이상 280미만(고지혈증 1기) 4. 280이상(고지혈증 2기)
params(12)	Double	Y	중성지방 5. 150미만(정상) 6. 150이상 200미만(경계) 7. 200이상 400미만(높음) 8. 400이상(매우 높음)
params(13)	Double	Y	흡연 여부 0. 모름 1. 흡연 2. 금연 3. 비흡연
params(14)	Double	Y	1년간 음주빈도 0. 술을 마셔 본 적 없음 1. 최근 1년간 전혀 마시지 않았다 2. 월 1회미만

			3. 월 1회정도 4. 월 2-4회 5. 주 2-3회정도 6. 주 4회이상
params(15)	Double	Y	평소 스트레스 인지 정도 5. 모름 6. 대단히 많이 느낀다 7. 많이 느끼는 편이다 8. 조금 느끼는 편이다 9. 거의 느끼지 않는다
params(16)	Double	Y	1주일간 걷기 일수 9. 모름 10. 전혀 하지 않음 11. 1일 12. 2일 13. 3일 14. 4일 15. 5일 16. 6일 17. 7일(매일)
params(17)	Double	Y	1주일간 근력운동 일수 7. 모름 8. 최근 7일 동안 없다 9. 1일 10. 2일 11. 3일 12. 4일 13. 5일이상

### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
modelId	String	뇌졸중 예측모델 아이디
rate	Double	뇌졸중 질환 발병 확률

### 5.1.1.23. Method predictionBodyAge

신체나이 예측

#### ■ Prototype

```
public func predictionCoronaryArtery(param1: Double, param2: Double, param3: Double,
    param4: Double, param5: Double, param6: Double, param7: Double,
    completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

#### ■ Parameters

이름	타입	필수	설명
params(1)	Double	Y	나이
params(2)	Double	Y	키
params(3)	Double	Y	체중
params(4)	Double	Y	수축기혈압
params(5)	Double	Y	이완기혈압
params(6)	Double	Y	공복혈당
params(7)	Double	Y	혈색소(해모글로빈)

#### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
-------	----	----



error	Error	HTTP 통신 에러
modelId	String	신체나이 예측모델 아이디
rate	Double	신체나이

#### 5.1.1.24. Method predictionASCVD

동맥경화성 심혈관질환(ASCVD) 예측

##### ■ Prototype

```
public func predictionCoronaryArtery(param1: Double, param2: Double, param3: Double,
    param4: Double, param5: Double, param6: Double, param7: Double,
    completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

##### ■ Parameters

이름	타입	필수	설명
params(1)	Double	Y	성별 0. 여자 1. 남자
params(2)	Double	Y	나이
params(3)	Double	Y	총 콜레스테롤
params(4)	Double	Y	HDL 콜레스테롤
params(5)	Double	Y	수축기 혈압
params(6)	Double	Y	흡연여부 1. 흡연 2. 비흡연
params(7)	Double	Y	당뇨 유병여부 1. 당뇨

			2. 비당뇨
--	--	--	--------

### Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
modelId	String	동맥경화성 심혈관질환 예측모델 아이디
rate	Double	동맥경화성 심혈관질환 발병 확률

#### 5.1.1.25. Method predictionCHD

관상동맥질환(CHD) 예측

### Prototype

```
public func predictionCoronaryArtery(param1: Double, param2: Double, param3: Double,
    param4: Double, param5: Double, param6: Double, param7: Double, param8: Double,
    completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

### Parameters

이름	타입	필수	설명
params(1)	Double	Y	성별 0. 여자 1. 남자
params(2)	Double	Y	나이
params(3)	Double	Y	총 콜레스테롤
params(4)	Double	Y	HDL 콜레스테롤
params(5)	Double	Y	수축기 혈압

params(6)	Double	Y	이완기 혈압
params(7)	Double	Y	흡연여부 1. 흡연 2. 비흡연
params(8)	Double	Y	당뇨 유병여부 1. 당뇨 2. 비당뇨

### Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
modelId	String	관상동맥질환 예측모델 아이디
rate	Double	관상동맥질환 발병 확률

### 5.1.1.26. Method registerUserBlock

회원 정보 Block 생성

### Prototype

```
public func registerUserBlock(_ url: String, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

### Parameters

이름	타입	필수	설명
url	String	Y	사용자 정보(Patient) FHIR 리소스 URL

### Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
url	String	사용자 정보(Patient) FHIR 리소스 URL
status	String	상태 ("ACTIVE", "REMOVED")
docType	String	문서 타입("Patient")
resources	Array	해당 사용자의 FHIR 리소스 목록

#### 5.1.1.27. Method registerResourceBlock

Resource Block 추가

### Prototype

```
public func registerResourceBlock(type: String, url: String, lastUpdated: String? = nil,
    completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

### Parameters

이름	타입	필수	설명
type	String	Y	FHIR 리소스 타입
url	String	Y	사용자 정보(Patient) FHIR 리소스 URL
lastUpdated	String	-	해당 리소스 최종 수정 일시 ( YYYY-MM-DD hh:mm:ss ) 입력되지 않을 경우 현재 서버 시간으로 반영됨

### Return(JSON<String, Any>)

출력 변수	타입	설명
-------	----	----

error	Error	HTTP 통신 에러
userId	String	사용자 아이디
url	String	사용자 정보(Patient) FHIR 리소스 URL
status	String	상태 ("ACTIVE", "REMOVED")
docType	String	문서 타입("Patient")
resources	Array	해당 사용자의 FHIR 리소스 목록
- type	String	FHIR 리소스 타입
- url	String	해당 리소스 타입의 가장 최근에 등록된 URL
- lastUpdated	String	해당 리소스 최종 수정 일시
- count	Int	해당 타입의 리소스 등록 개수

### 5.1.1.28. Method updateUserBlock

회원 정보 Block 수정

#### ■ Prototype

```
public func updateUserBlock(_ url: String, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

#### ■ Parameters

이름	타입	필수	설명
url	String	Y	사용자 정보(Patient) FHIR 리소스 URL

#### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
url	String	사용자 정보(Patient) FHIR 리소스 URL
status	String	상태 ("ACTIVE", "REMOVED")

docType	String	문서 타입("Patient")
resources	Array	해당 사용자의 FHIR 리소스 목록

### 5.1.1.29. Method getUserBlock

회원 정보 Block 조회

#### ■ Prototype

```
public func getUserBlock(completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

#### ■ Parameters

이름	타입	필수	설명
없음			

#### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
userId	String	사용자 아이디
url	String	사용자 정보(Patient) FHIR 리소스 URL
status	String	상태 ("ACTIVE", "REMOVED")
docType	String	문서 타입("Patient")
resources	Array	해당 사용자의 FHIR 리소스 목록

### 5.1.1.30. Method deleteUserBlock

회원 정보 Block 삭제

#### ■ Prototype

```
public func updateUserBlock(_ url: String, completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

#### ■ Parameters

이름	타입	필수	설명
없음			

#### ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러

### 5.1.1.31. Method getApp

모바일 앱 최신 버전정보 조회

#### ■ Prototype

```
public func getApp(clientId: String, clientSecret: String, appSerialNumber: String,
completion: @escaping (Error?, Dictionary<String, Any>?) -> Void)
```

#### ■ Parameters

이름	타입	필수	설명
clientId	String	Y	플랫폼에서 발급받은 클라이언트 아이디
clientSecret	String	Y	플랫폼에서 발급받은 클라이언트 시크릿
appSerialNumber	String	Y	플랫폼에 등록된 PHR서비스 앱 일련번호

## ■ Return(JSON<String, Any>)

출력 변수	타입	설명
error	Error	HTTP 통신 에러
appSn	String	앱 일련번호
appld	String	앱 ID
appNm	String	앱 이름
osTypeCode	String	OS 유형코드 ('Android', 'iOS', 'Windows')
appVerVal	String	앱 버전 값
dwldUrl	String	다운로드 URL
enfrcUpdtAt	String	강제 업데이트 여부 ('Y', 'N')
creatDt	String	생성 일시
updtDt	String	수정 일시

## 6. PHRHealthkitManager

HealthKit 정보를 플랫폼 FHIR 서버에 등록하는 패키지



## 6.1. RedwoodHealth-SDK.PHRHealthkitManager

HealthKit 정보를 플랫폼 FHIR 서버에 등록하는 패키지.

Class 구성

클래스	설명
PHRHealthkitManager	HealthKit에 접근하여 라이프로그 수집과 플랫폼 FHIR에 데이터를 전송하는 클래스

### 6.1.1. Class PHRHealthkitManager

클래스	메소드	설명
PHRHealthkitManager	authorizeHealthKit	HealthKit에 접근권한을 요청한다
	init	HealthKit에서 검색할 날짜와 patient ID를 초기화 한다
	updateSampleToFHIR	HealthKit 데이터를 FHIR 서버에 등록 요청한다

#### 6.1.1.1. Method authorizeHealthKit

HealthKit에 접근권한을 요청한다.

##### ■ Prototype

```
static public func authorizeHealthKit(completion: @escaping (Bool, Error?) -> Void)
```

##### ■ Parameters

이름	타입	필수	설명
없음			

## Return

출력 변수	타입	설명
success	Bool	접근 권한 성공 여부
error	Error	접근 실패시 에러 정보

### 6.1.1.2. Method init

HealthKit 데이터를 FHIR 서버에 등록 요청한다.

## Prototype

```
public init(fhIrPatiendID patiendID: String, startDate: Date, endDate: Date, timeZone: TimeZone?)
```

## Parameters

이름	타입	필수	설명
fhIrPatiendID	String	Y	Patient ID
startDate	Date	Y	시작 날짜/시간
endDate	Date	Y	종료 날짜/시간
timeZone	TimeZone?	N	데이터 발생 시간의 타임존 설정, 미설정 시 UTC

## Return

출력 변수	타입	설명
없음		

### 6.1.1.3. Method updateSampleToFHIR

HealthKit 데이터를 FHIR 서버에 등록 요청한다.

#### ■ Prototype

```
static public func authorizeHealthKit(completion: @escaping (Bool, Error?) -> Void)
```

#### ■ Parameters

이름	타입	필수	설명
없음			

#### ■ Return

출력 변수	타입	설명
error	Error	등록 실패시 에러 정보, null 이면 성공이다.

#### ■ 사용 예시

```
PHRHealthkitManager.authorizeHealthKit { (success, error) in
    // 성공
    if success {
        let nowDate = Date()
        let startDate = Calendar.current.date(byAdding: .day, value: -1, to: nowDate)
        let timeZone = TimeZone(abbreviation: "KST")

        let manager = PHRHealthkitManager(fhirPatiendID: patientId,
                                           startDate: startDate!, endDate: nowDate,
                                           timeZone: timeZone)

        manager.updateSampleToFHIR() { (error) in
            if let err = error {
                // 에러처리
            }
        }
    }
}
```

```
        ...(중략)...
    } else {
        // 성공
        ...(중략)...
    }
}
}
```