PHR 기반 개인맞춤형 건강관리 서비스 플랫폼 기술 고도화
및 실증서비스 개발연계지원

인터페이스 정의서(OAuth 2.0)

㈜ 디케이아이테크놀로지

문서번호: PHRP_KEIT-SD-04
Ver : 1.2

개정 이력

개정일자	문서 번호	Protocol Version	개정 내역	작성자	확인자
	1.0	1.0	1. 초안 작성		
2019.11.15	1.1	1.0	플랫폼 서버 정보 추가	김현연	김현연
2020.01.22	1.2	1.0	4.3.2. 기존 body 입력값 일부를 header로 이동	마현철	

문서 목차

1. 개요 4

1.1. 연동규격의 목적 4
1.2. 연동규격의 개정 4

2. 관련 규격 4

2.1. OAUTH 2.0 4

3. 연동 구조 5

3.1. 서비스 등록 5
3.2. 사용자 가입 6
3.3. 인증토큰 발급 7
3.4. PHR API(OPEN API, FHIR API) 호출 10
3.5. REFRESH_TOKEN을 이용한 토큰 재발급 12

4. 인터페이스 상세 14

4.1. 서비스 등록 14
4.1.1. 서비스 사용 신청 14
4.1.2. 서비스 사용 승인 14
4.1.3. 서비스 사용 승인 결과 조회 14
4.2. 사용자 가입 14
4.2.1. 회원 가입 14
4.3. 인증토큰 발급(AUTHORIZATION CODE GRANT TYPE) 15
4.3.1. 인증토큰 발급 요청 15
4.3.2. 로그인 및 권한 승인 후 Code를 이용한 인증토큰 발급 요청 16
4.4. 인증토큰 발급(IMPLICIT GRANT TYPE) 17
4.4.1. 인증토큰 발급 요청 17
4.5. PHR API(OPEN API, FHIR API) 호출 18
4.6. REFRESH_TOKEN을 이용한 토큰 재발급 19
4.6.1. 인증토큰 재발급 요청 19
4.7. 응답코드 20

1. 개요

본 연동규약은 PHR 기반 개인맞춤형 건강관리 서비스 플랫폼에 제공되는 인증 관련 OPEN API 요청 및 응답 메시지를 정의하고 있다. 인증 규격은 OAuth 2.0 규격을 준수한다.

1.1. 연동규격의 목적

본 기술 규격서는 "PHR 기반 개인맞춤형 건강관리 서비스 플랫폼"의 서비스 시스템과 플랫폼의 인증 관련 API 연동 규격을 정의한다.

1.2. 연동규격의 개정

본 규격서는 필요하다고 판단되는 경우 개정이 가능하며, 그 절차는 "PHR 기반 개인맞춤형 건강관리 서비스 플랫폼 기술 고도화 및 실증서비스 개발연계지원"의 규정에 준한다.

2. 관련 규격

2.1. OAuth 2.0

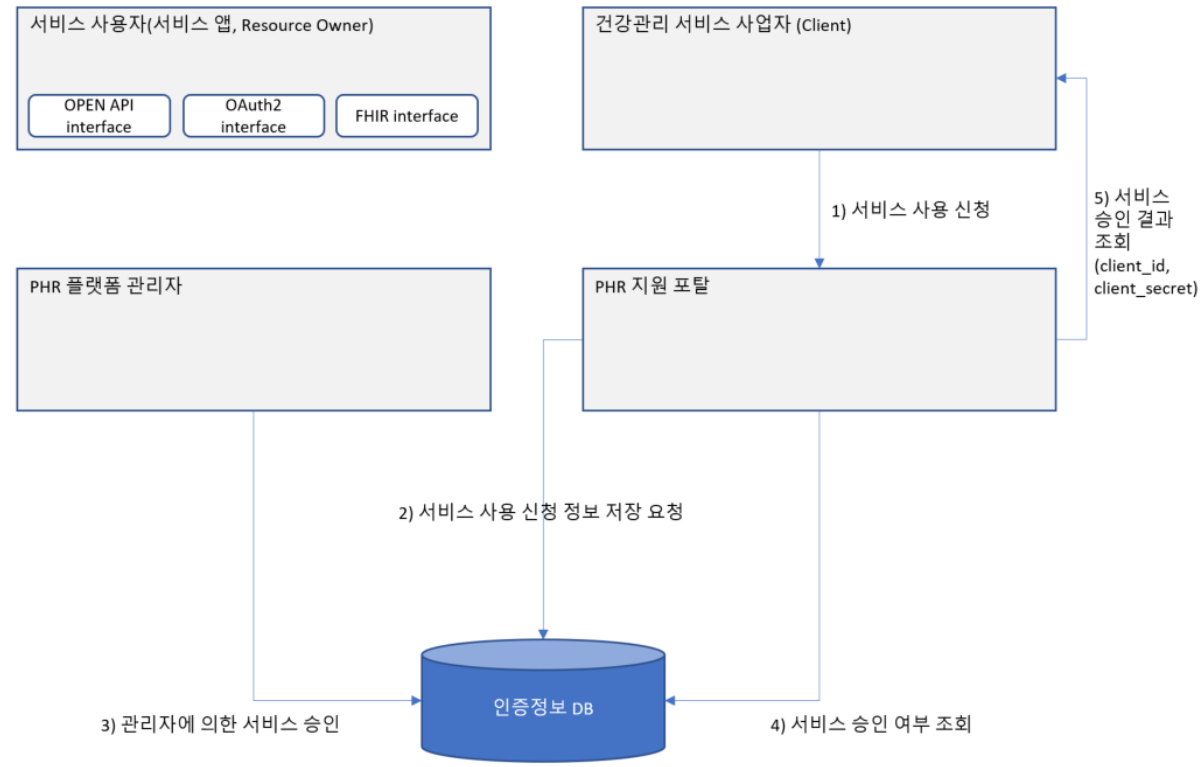
- http://oauth.net/2
- OAuth 2.0 Core
  - OAuth 2.0 Framework - RFC 6749
  - Bearer Token Usage - RFC 6750
  - Threat Model and Security Considerations - RFC 6819
- Protocols Built on OAuth 2.0
  - Open ID Connect
  - Blue Button
  - Green Button
  - UMA
  - GSMA OneAPI
- OAuth 2.0 Extensions
  - JSON Web Token - RFC 7519
  - OAuth Assertions Framework - RFC 7521
  - SAML2 Bearer Assertion - RFC 7522, for integrating with existing identity systems
  - JWT Bearer Assertion - RFC 7523, for integrating with existing identity systems

3. 연동 구조

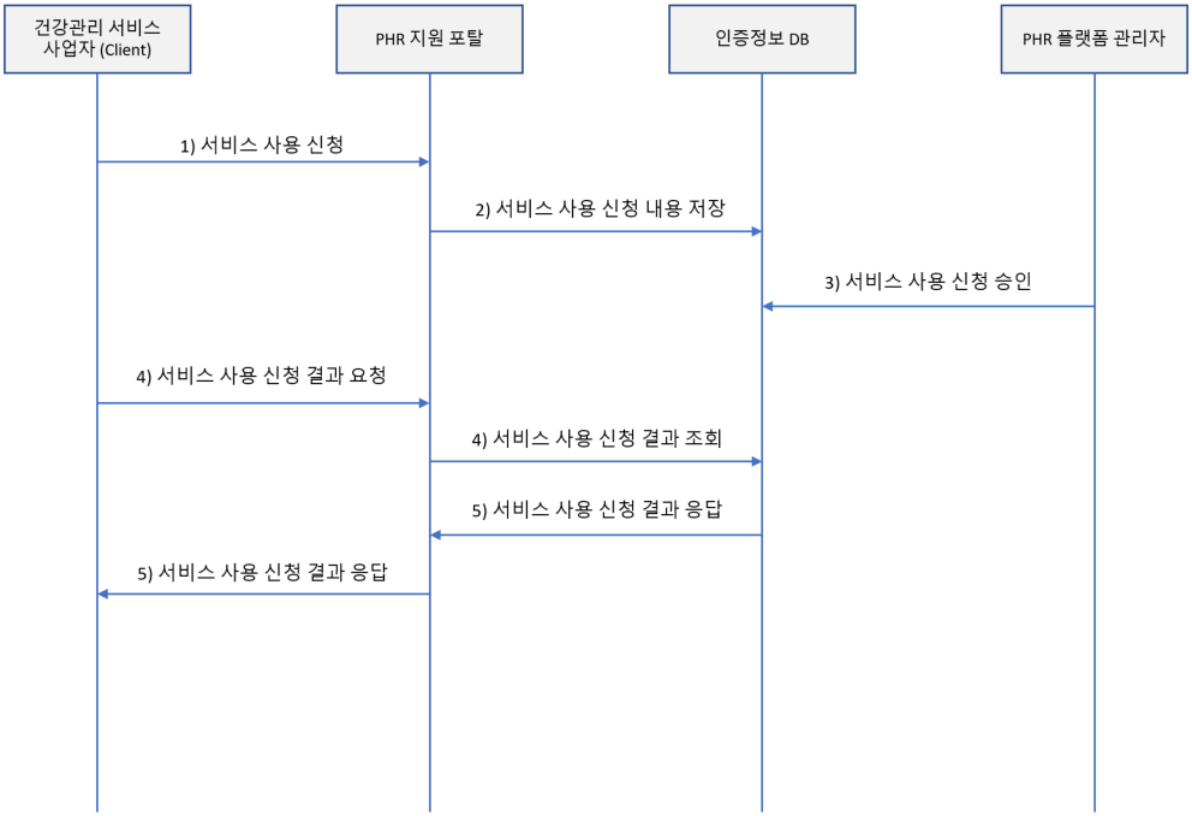
아래는 PHR 플랫폼과 서비스 시스템 간의 인증 API 연동 구조를 도식과 함께 정의하고 있다.

3.1. 서비스 등록

인증 프로세스를 수행하기 전, 각 건강관리 서비스 사업자는 PHR 지원 포탈을 통해 서비스 사용 신청을 수행해야 한다. 서비스 신청 후 관리자의 승인이 이루어지면, 인증 과정 수행 시 필요한 client_id 및 client_secret을 발급받을 수 있다.

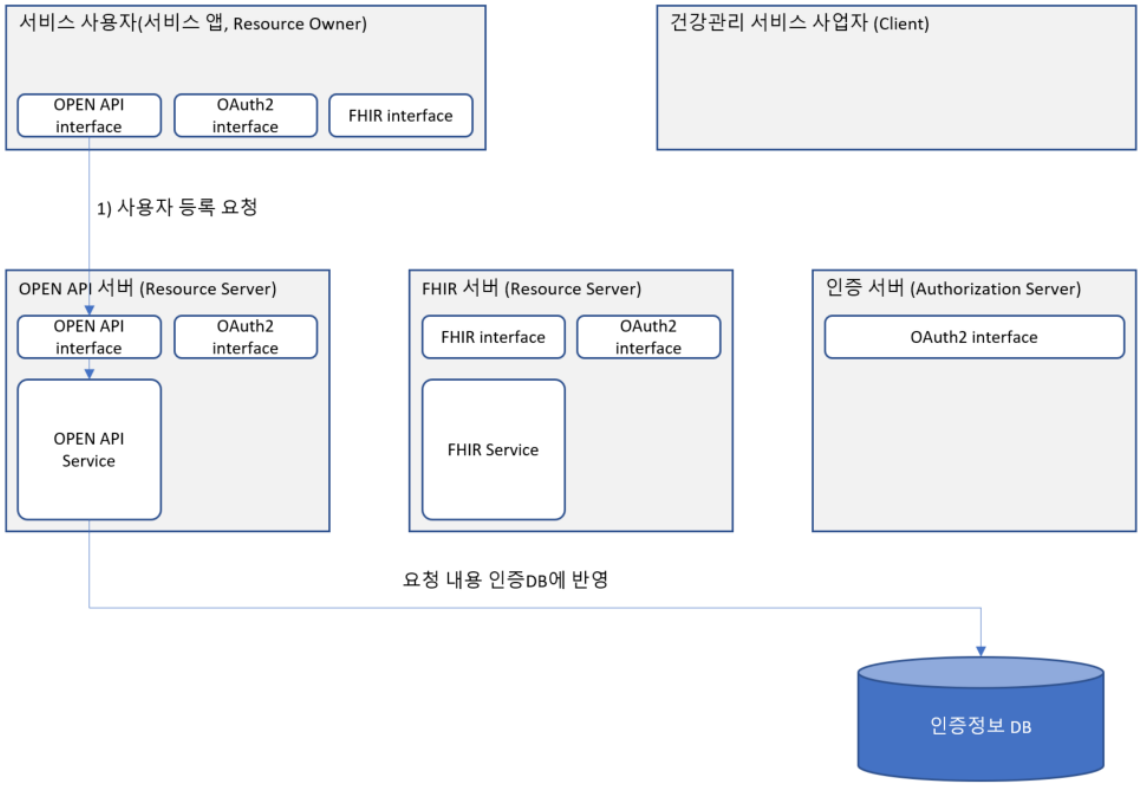


순서	설명
1	건강관리 서비스 사업자는 PHR 지원 포탈에 접속하여 서비스 사용 신청을 한다.
2	PHR 지원 포탈을 통해 서비스 사용 신청 정보가 DB에 저장된다.
3	PHR 플랫폼 관리자는 해당 서비스 정보를 확인한 후 승인 처리한다.
4	이후 건강관리 서비스 사업자가 PHR 지원 포탈을 통해 서비스 승인 결과를 조회할 수 있다.
5	서비스 승인 시 client_id 및 client_secret이 발급된다.

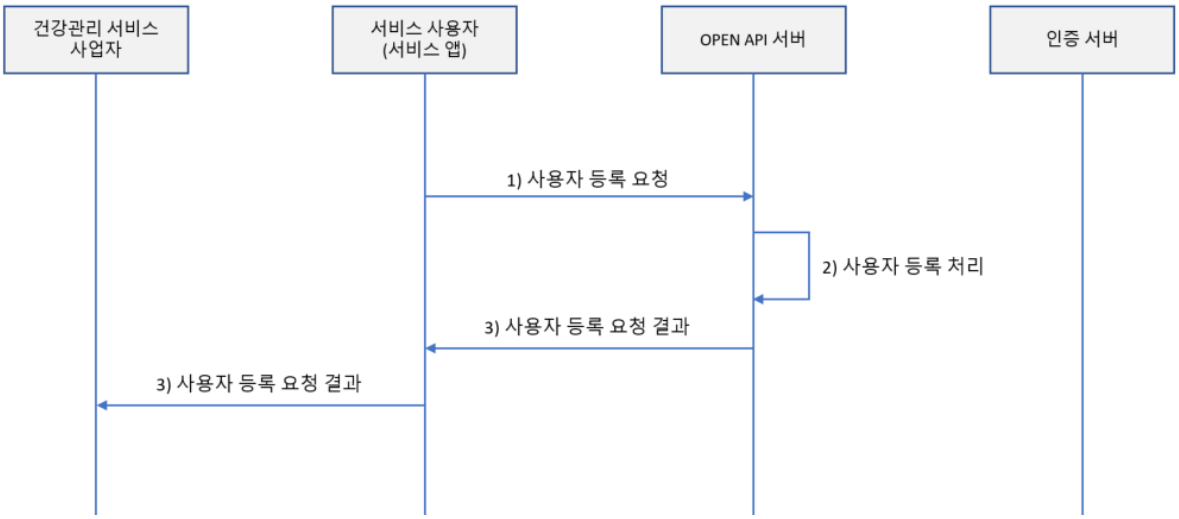


3.2. 사용자 가입

승인된 서비스를 이용할 사용자 정보를 등록한다. 서비스 앱 등을 통해 사용자가 필요한 값을 입력한 후 가입을 완료한다.



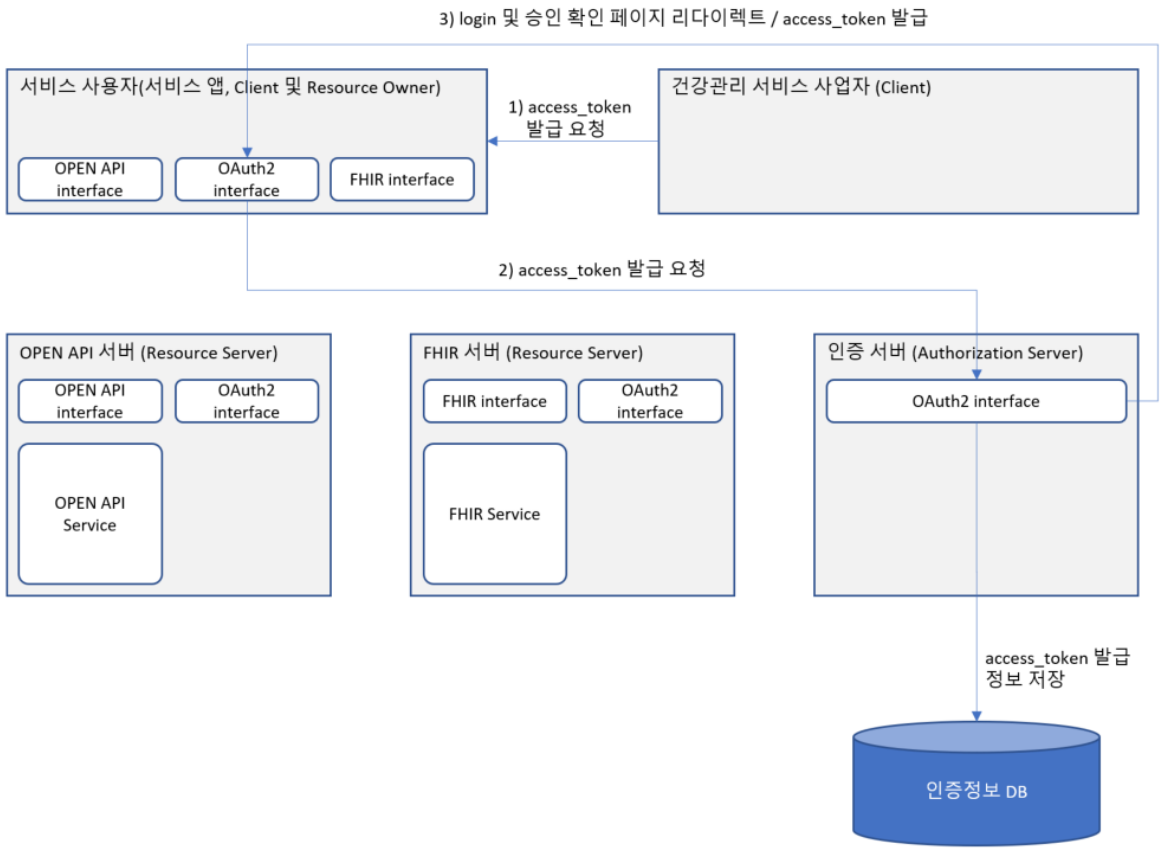
순서	설명
1	서비스 앱 등을 통해 서비스 사용자는 사용자 등록 요청을 수행한다.
2	등록 내용을 인증 DB에 반영한다.



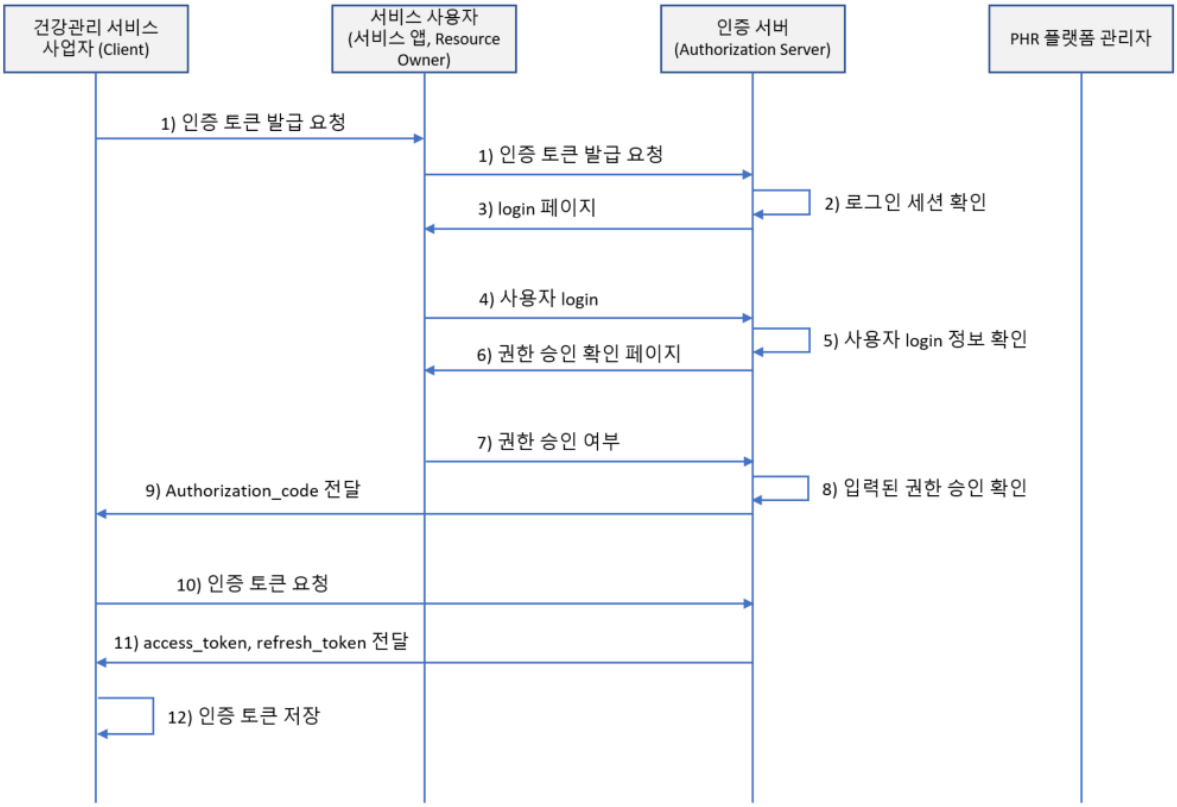
3.3. 인증토큰 발급

PHR 플랫폼의 OPEN API 또는 FHIR API 호출 시 사용할 인증 토큰을 발급받는다. access_token 및

refresh_token이 발급되며, 유효기간이 지난 경우 다시 토큰 발급 프로세스를 수행하여 토큰을 획득한다. PHR 플랫폼에서는 OAuth 2.0 규격에서 정의하는 토큰 발급 방식 중 '**Authorization Code Grant**' 방식과 '**Implicit Grant**' 방식을 통한 토큰 발급을 권장한다.

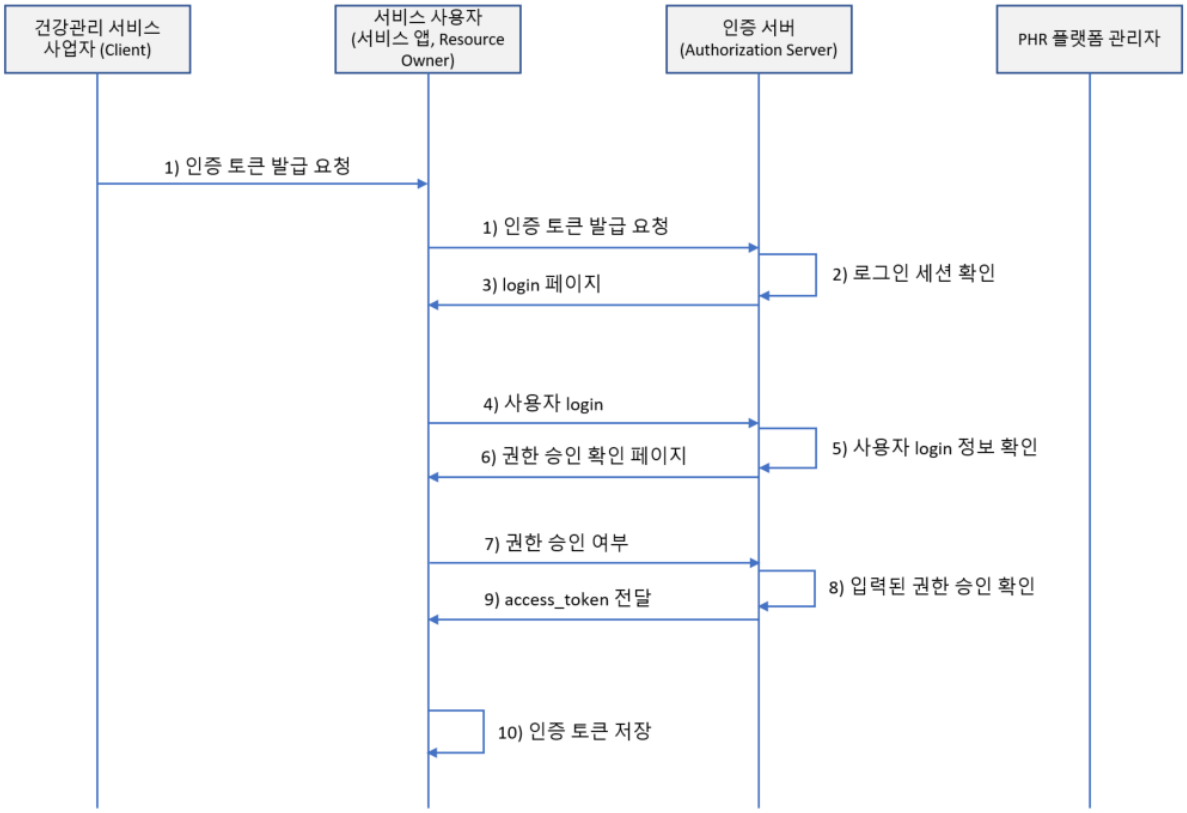


순서	설명
1	서비스 사업자의 어플리케이션에서 필요에 따라 인증토큰 발급 요청을 시작한다.
2	서비스 앱에서 OAuth2 interface를 통해 인증 토큰 발급 요청을 수행한다.
3	1) 인증 서버에 현재 로그인 된 세션이 존재하지 않을 경우 login 화면으로 리다이렉트되며, 사용자의 로그인 후 승인 확인 페이지가 노출된다. 2) 인증 서버에 로그인 된 세션이 존재할 경우 인증 토큰 정보가 리턴된다.



인증 토큰 발급 (Authorization Code Grant Type)

순서	설명
1	서비스 사업자의 어플리케이션에서 필요에 따라 인증토큰 발급 요청을 시작한다.
2	인증 서버에서는 현재 사용자 계정으로 로그인이 되었는지 확인한다.
3	로그인 되지 않았을 경우 서비스 사용자에게 login 페이지를 리턴한다.
4	사용자가 로그인을 수행한다.
5	인증 서버에서 로그인 정보가 유효한지 확인한다.
7	로그인에 성공할 경우, 사용자에게 해당 API 호출을 위한 권한을 고지하고, 이에 대한 승인을 받는 페이지를 리턴한다.
8	사용자가 권한을 승인 여부를 확인한다.
9	권한을 승인할 경우, 서비스 등록 시 입력된 redirect_uri로 authorization_code값이 리턴된다.
10	전달받은 authorization_code를 포함하여 인증 토큰 발급 요청을 수행한다.
11	access_token 및 refresh_token이 리턴된다.
12	서비스 서버에서는 해당 access_token과 refresh_token값을 저장한 후 인증에 사용한다.

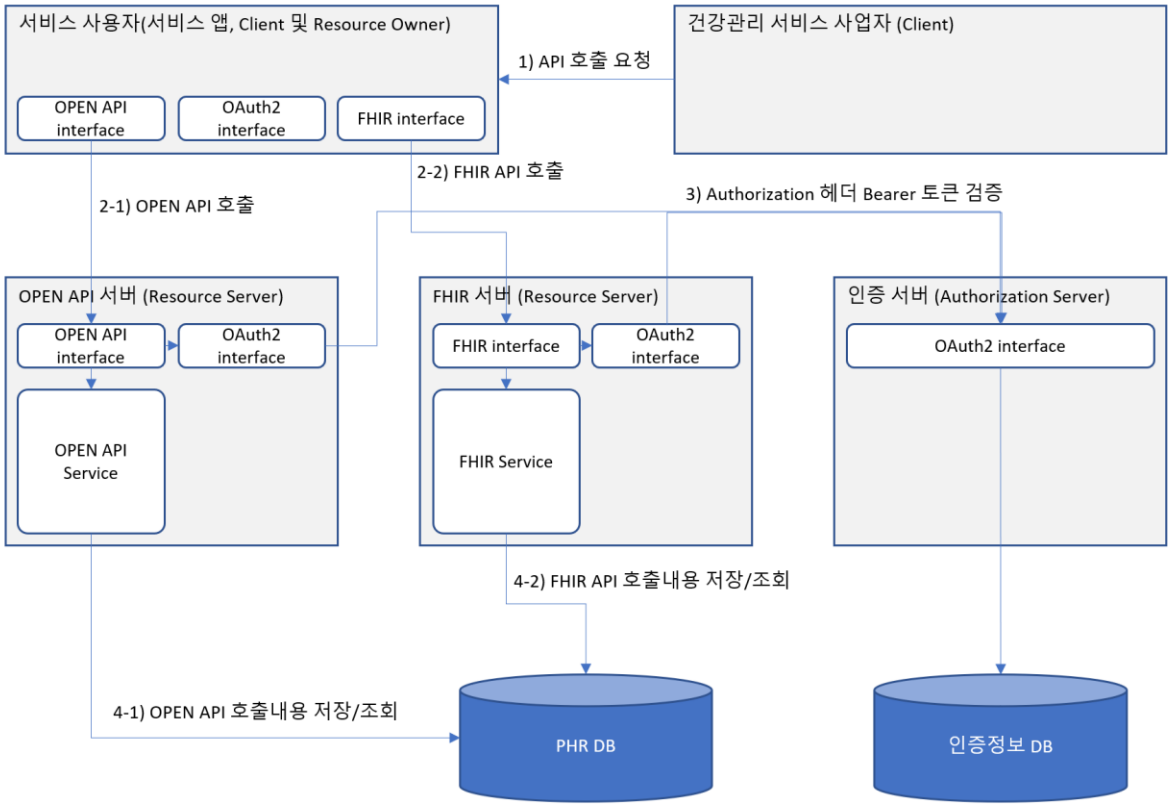


인증 토큰 발급 (Implicit Grant Type)

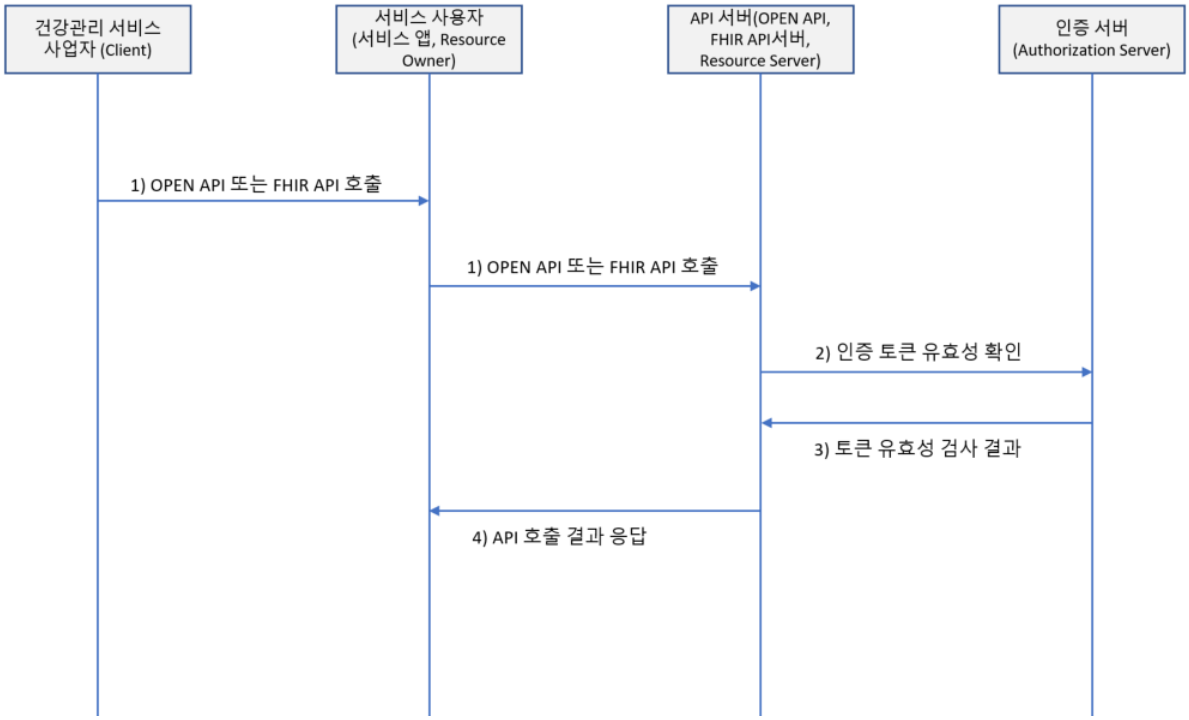
순서	설명
1	서비스 사업자의 어플리케이션에서 필요에 따라 인증토큰 발급 요청을 시작한다.
2	인증 서버에서는 현재 사용자 계정으로 로그인이 되었는지 확인한다.
3	로그인 되지 않았을 경우 서비스 사용자에게 login 페이지를 리턴한다.
4	사용자가 로그인을 수행한다.
5	인증 서버에서 로그인 정보가 유효한지 확인한다.
7	로그인에 성공할 경우, 사용자에게 해당 API 호출을 위한 권한을 고지하고, 이에 대한 승인을 받는 페이지를 리턴한다.
8	사용자가 권한을 승인 여부를 확인한다.
9	권한을 승인할 경우, 서비스 등록 시 입력된 redirect_uri로 access_token값이 전달 된다.
10	서비스 앱에서는 발급받은 토큰을 추후 API호출 시 인증에 사용한다.

3.4. PHR API(OPEN API, FHIR API) 호출

3.3. 인증토큰 발급에서 발급받은 인증 토큰을 HTTP header에 입력하여 PHR API(OPEN API, FHIR API)를 호출한다. 각 OPEN API서버 및 FHIR 서버에서는 전달받은 인증 토큰의 유효성을 인증 서버를 통해 확인하며, 유효한 경우 해당 API의 호출 결과를 처리, 응답한다.

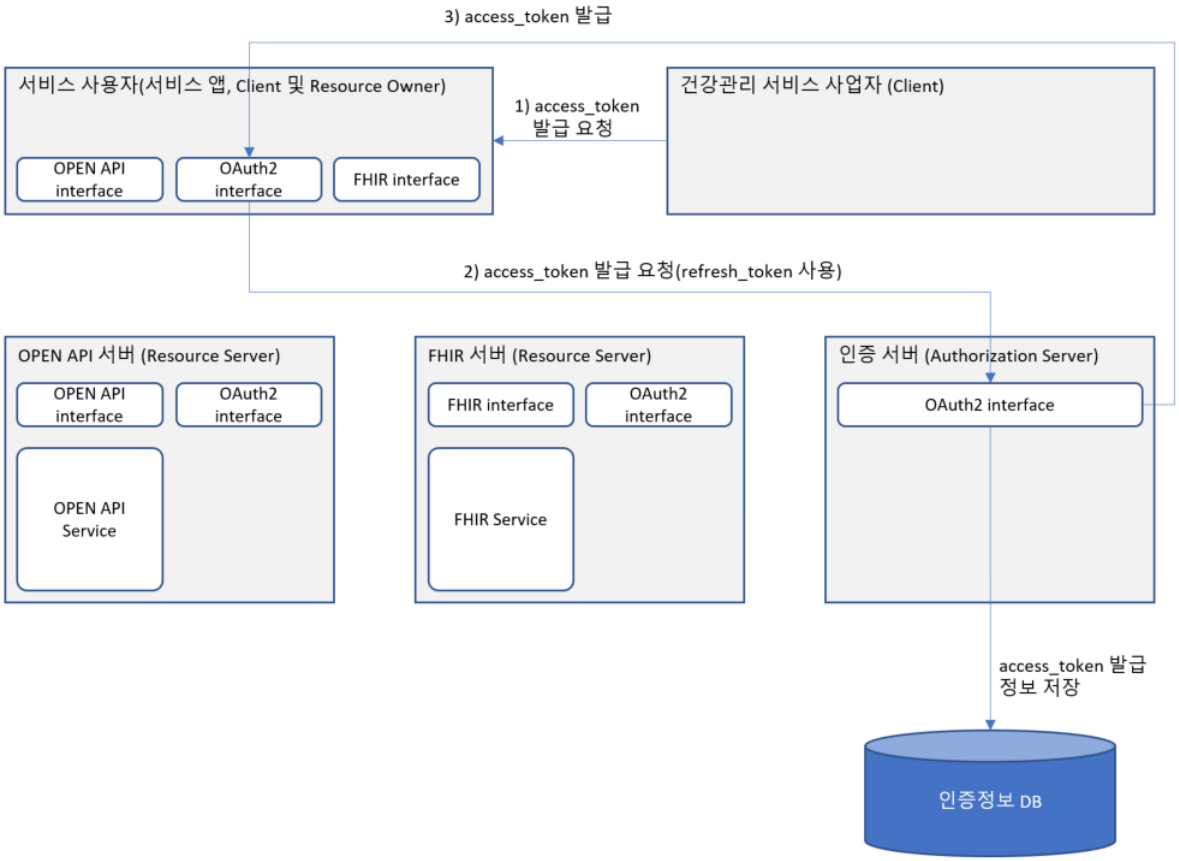


순서	설명
1	서비스 사업자의 어플리케이션에서 필요에 따라 API 호출 요청을 시작한다.
2	인증 토큰이 획득된 상태일 경우, 각 기능 상 필요로 하는 API를 호출한다. 1) OPEN API호출 시 HTTP Header 'Authorization'값에 인증 토큰을 입력한다. 2) FHIR API호출 시 HTTP Header 'Authorization'값에 인증 토큰을 입력한다.
3	각 서버에서는 해당 인증 토큰의 유효성을 인증 서버를 통해 확인한다.
4	인증 토큰이 유효한 경우 각 API 호출 결과를 수행한 후 응답한다.

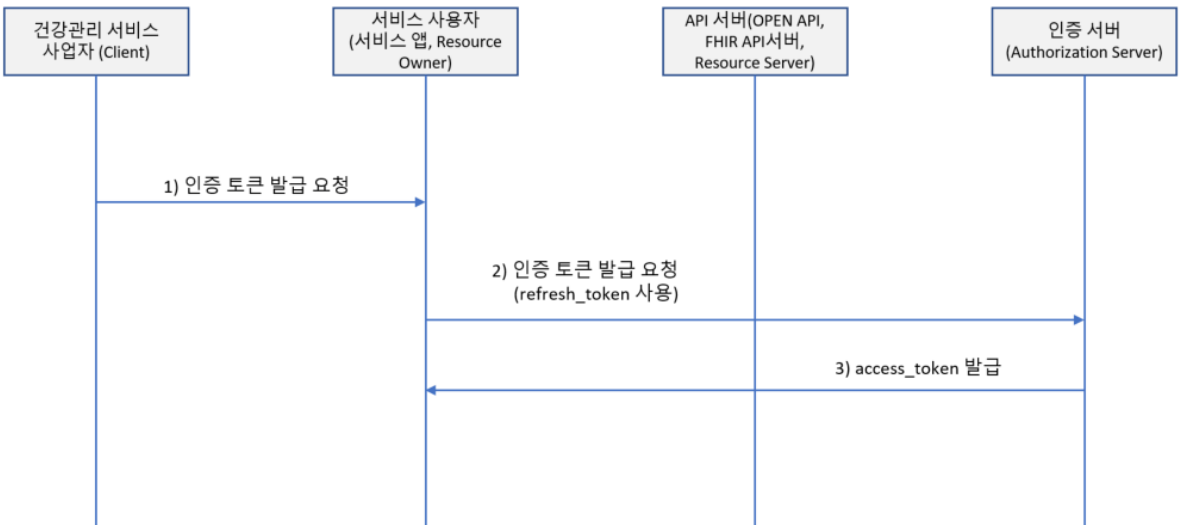


3.5. refresh_token을 이용한 토큰 재발급

인증 토큰 발급 시 'Authorization Code Grant' 방식을 통해 토큰을 발급받을 경우, access_token과 함께 refresh_token이 발급된다. refresh_token을 소유하고 있을 경우, 해당 토큰을 이용하여 access_token의 유효기간이 만료되었을 때 재발급을 받을 수 있다. 'Implicit Grant' 방식을 통해 획득한 access_token이 만료될 경우에는 인증토큰 발급 과정을 다시 수행하여 access_token을 발급받는다.



순서	설명
1	서비스 사업자의 어플리케이션에서 필요에 따라 인증토큰 발급 요청을 시작한다.
2	서비스 앱에서 OAuth2 interface를 통해 인증 토큰 발급 요청을 수행한다. 이때 refresh_token을 이용하여 토큰 발급을 요청한다.
3	refresh_token이 유효할 경우 인증 토큰이 발급된다.



3.6. 플랫폼 서버 정보

구분	서버	URL	비고
테스트 배드	OAuth	http://oauthtb.redwoodhealth.kr/	
	Open API	http://apitb.redwoodhealth.kr	
	FHIR	http://fhirtb.redwoodhealth.kr	
	블록체인	http://bctb.redwoodhealth.kr	
	지원포탈	http://wwwtb.redwoodhealth.kr	
	운영관리	http://onmtb.redwoodhealth.kr	
	Push	http://pushtb.redwoodhealth.kr	
상용 서비스	OAuth	http://oauth.redwoodhealth.kr	
	Open API	https://api.redwoodhealth.kr	
	FHIR	https://fhir.redwoodhealth.kr	
	블록체인	https://bc.redwoodhealth.kr	
	지원포탈	https://www.redwoodhealth.kr	
	운영관리	https://onm.redwoodhealth.kr	
	Push	http://push.redwoodhealth.kr	

4. 인터페이스 상세

4.1. 서비스 등록

4.1.1. 서비스 사용 신청

서비스 사용 신청은 PHR 지원 포탈 UI를 통해 이루어진다.

4.1.2. 서비스 사용 승인

서비스 사용 승인은 PHR 관리 UI를 통해 플랫폼 관리자가 수행한다.

4.1.3. 서비스 사용 승인 결과 조회

서비스 사용 신청 결과 조회는 PHR 지원 포탈 UI를 통해 이루어지며, 발급된 client_id와 client_secret을 확인할 수 있다.

4.2. 사용자 가입

4.2.1. 회원 가입

회원 가입은 OPEN API 서버를 통해 이루어지며, 자세한 사항은 PHRP_KEIT-SD-04-(인터페이스정의-OpenAPI) 문서를 참조한다.

4.3. 인증토큰 발급(Authorization Code Grant Type)

인증토큰 발급 시 사용되는 OAuth 2.0 API의 상세 내용을 기술한다. 아래 기술된 API는 OAuth 2.0 인증 타입 중 **Authorization Code Grant Type**을 이용한 경우이다.

4.3.1. 인증토큰 발급 요청

⊙ 요청 주소

Method	요청 주소
GET	{PHR 인증서버}/oauth/authorize

⊙ 요청 변수 Parameters

파라미터	타입	필수여부	설명
client_id	string	Y	승인 완료된 서비스 아이디
redirect_uri	string	Y	서비스 사용 신청 시 입력한 redirect_uri
response_type	string	Y	"code" 입력(Authorization Code Grant Type 표시)
scope	string	Y	"phr.read phr.write" 입력(공백문자로 구분)
state	string	Y	클라이언트에서 서버의 응답 유효성을 확인하기 위한 값. code 응답 시 입력한 값이 함께 전달된다.

⊙ 요청 예시

```
GET /oauth/authorize
?scope=phr.read%20phr.write
&redirect_uri=http%3A%2F%2F127.0.0.1%3A7000%2Fphrtest%2FreceiveCode.html
&response_type=code&client_id=my_client_id
&state=1234 HTTP/1.1
```

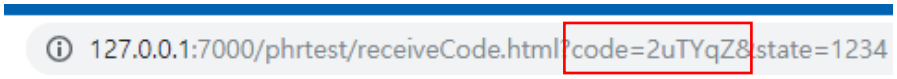
⊙ 응답 예시

```
HTTP/1.1 302 found
Location: http://localhost:8080/login
```

➔ 인증서버에 로그인 세션이 존재하지 않을 경우 사용자에게 login및 권한 승인을 요청하는 페이지가 전달된다.

➔ 유효한 세션이 존재할 경우 입력한 redirect_uri로 authorization_code값이 전달된다.

예)



해당 코드를 이용하여 access_token 및 refresh_token을 요청할 수 있게 된다.

4.3.2. 로그인 및 권한 승인 후 Code를 이용한 인증토큰 발급 요청

⊙ 요청 주소

Method	요청 주소
POST	{PHR 인증서버}/oauth/token

⊙ HTTP Headers

파라미터	타입	필수여부	설명
Content-Type	string	Y	application/x-www-form-urlencoded
Authorization	string	Y	client_id(서비스 아이디)와 client_secret(클라이언트 시크릿) 값을 ':'으로 연결하여 base64 인코딩한 basic 인증값

⊙ 요청 변수 Parameters

파라미터	타입	필수여부	설명
code	string	Y	4.3.1. 인증토큰 발급 요청 시 발급받은 code값
redirect_uri	string	Y	서비스 승인 요청 시 입력한 redirect_uri
grant_type	string	Y	"authorization_code" 입력

⊙ 요청 예시

```
POST /oauth/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Authorization: basic myb64encodedbasicauthentication==

code=lfn00r
&redirect_uri= http%3A%2F%2F127.0.0.1%3A7000%2Fphrtest%2FreceiveCode.html
&grant_type=authorization_code
```

⊙ 응답 예시

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
    "access_token":"f8e44056-f8b9-4958-b53c-a54f2c636357",
    "token_type":"bearer",
    "refresh_token":"e9ccf1fa-3867-4b17-8dfd-2c947b0fa2e7",
    "expires_in":35141,
    "scope":"phr.read phr.write"
}
```

4.4. 인증토큰 발급(Implicit Grant Type)

인증토큰 발급 시 사용되는 OAuth 2.0 API의 상세 내용을 기술한다. 아래 기술된 API는 OAuth 2.0 인증 타입 중 **Implicit Grant Type**을 이용한 경우이다.

4.4.1. 인증토큰 발급 요청

⊙ 요청 주소

Method	요청 주소
GET	{PHR 인증서버}/oauth/authorize

⊙ 요청 변수 Parameters

파라미터	타입	필수여부	설명
client_id	string	Y	승인 완료된 서비스 아이디
redirect_uri	string	Y	서비스 사용 신청 시 입력한 redirect_uri
response_type	string	Y	"token" 입력(Implicit Grant Type 표시)
scope	string	Y	"phr.read phr.write" 입력(공백문자로 구분)

⊙ 요청 예시

```
GET /oauth/authorize
?scope=phr.read%20phr.write
&redirect_uri=http%3A%2F%2F127.0.0.1%3A7000%2Fphrtest%2FreceiveCode.html
&response_type=token&client_id=my_client_id HTTP/1.1
```

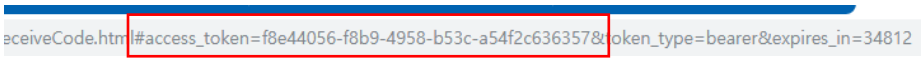
⊙ 응답 예시

```
HTTP/1.1 302 found
Location: http://localhost:8080/login
```

➔ 인증서버에 로그인 세션이 존재하지 않을 경우 사용자에게 login및 권한 승인을 요청하는 페이지가 전달된다.

➔ 유효한 세션이 존재할 경우 입력한 redirect_uri로 access_token값이 전달된다.

예)



redirect_uri 뒤에 연결된 '#' 이하로 전달된 access_token을 이용하여 api를 호출할 수 있다.

4.5. PHR API(OPEN API, FHIR API) 호출

앞선 절차에서 획득한 access_token을 이용하여 PHR 플랫폼의 OPEN API 및 FHIR API를 호출할 수 있다. HTTP Header 'Authorization' 항목에 'Bearer {발급된 access_tone}'값을 입력하여 API를 호출한다.

⊙ 요청 예시

```
GET /fhir/Patient/1102/_history/2 HTTP/1.1
Authorization: Bearer f8e44056-f8b9-4958-b53c-a54f2c636357
Accept: application/json
```

⊙ 응답 예시

```
HTTP/1.1 200 OK
Content-Type: application/fhir+json;charset=UTF-8
Content-Location: http://localhost:8080/fhir/Patient/1102/_history/2

{
  "resourceType": "Patient",
  "id": "1102",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2019-04-23T13:47:10.000+09:00"
```

```
  },
  "text": {
    "status": "generated",
    "div": "<div xmlns=₩"http://www.w3.org/1999/xhtml₩"><div class=₩"hapiHeaderText₩">Mr. Kris <b>MOHR </b></div><table class=₩"hapiPropertyTable₩"><tbody><tr><td>Identifier</td><td>00ac7f61-47bb-4423-8d1a-acc07b206e34</td></tr><tr><td>Address</td><td><span>14169 Una Lock </span><br/><span>Suite 360 </span><br/><span>Springfield </span><span>MA </span><span>US </span></td></tr><tr><td>Date of birth</td><td><span>19 8월 1989</span></td></tr></tbody></table></div>"
  },
  "identifier": [
    {
      "system": "https://github.com/synthetichealth/synthea",
      "value": "00ac7f61-47bb-4423-8d1a-acc07b206e34"
    },
    ...(중략)...
}
```


4.6. refresh_token을 이용한 토큰 재발급

Authorization Code Grant Type을 통해 access_token을 발급받은 경우 refresh_token을 함께 발급받게 된다. 아래 예시는 access_token이 만료된 경우 refresh_token을 이용하여 재발급 받는 API를 기술한다.

4.6.1. 인증토큰 재발급 요청

⊙ 요청 주소

Method	요청 주소
POST	{PHR 인증서버}/oauth/token

⊙ HTTP Headers

파라미터	타입	필수여부	설명
Content-Type	string	Y	application/x-www-form-urlencoded
Authorization	string	Y	승인완료 시 발급된 client_id와 client_secret을 이용 Basic [client_id]:[client_secret] (base64 인코딩)

⊙ 요청 변수 Parameters

파라미터	타입	필수여부	설명
grant_type	string	Y	"refresh_token" 입력
refresh_token	string	Y	발급받은 refresh_token값 입력

⊙ 요청 예시

```
POST /oauth/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Authorization: Basic bXlfY2xpZW50X2lkOm15X2NsaWVudF9zZWNyZXQ=

grant_type=refresh_token&
refresh_token=4be13e9b-f2bb-4828-98b9-d5c306ff7742
```

⊙ 응답 예시

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
    "access_token":"91d5d010-321b-479f-bef9-eee3c2034505",
    "token_type":"bearer",
    "refresh_token":"4be13e9b-f2bb-4828-98b9-d5c306ff7742",
    "expires_in":35999,
    "scope":"phr.read phr.write"
}
```

4.7. 응답코드

구분	응답 코드	응답코드명	설명
2xx (Success)	200	OK	서버가 요청을 제대로 처리했다는 뜻이다. 이는 주로 서버가 요청한 페이지를 제공했다는 의미한다.
	201	Created	성공적으로 요청되었으며 서버가 새 리소스를 작성했다.
	202	Accepted	서버가 요청을 접수했지만 아직 처리하지 않았다.
	203	Non-Authoritative Information	서버가 요청을 성공적으로 처리했지만 다른 소스에서 수신된 정보를 제공하고 있다.
	204	No Content	서버가 요청을 성공적으로 처리했지만 콘텐츠를 제공하지 않는다.

	205	Reset Content	서버가 요청을 성공적으로 처리했지만 콘텐츠를 표시하지 않는다. 204 응답과 달리 이 응답은 요청자가 문서 보기를 재설정할 것을 요구한다(예: 새 입력을 위한 양식 비우기).
	206	Partial Content	서버가 GET 요청의 일부만 성공적으로 처리했다
	207	Multi-Status	
	208	Already Reported	
3xx (Redirection)	300	Multiple Choices	서버가 요청에 따라 여러 조치를 선택할 수 있다. 서버가 사용자 에이전트에 따라 수행할 작업을 선택하거나, 요청자가 선택할 수 있는 작업 목록을 제공한다.
	301	Moved Permanently	요청한 페이지를 새 위치로 영구적으로 이동했다. GET 또는 HEAD 요청에 대한 응답으로 이 응답을 표시하면 요청자가 자동으로 새 위치로 전달된다.
	302	Found	현재 서버가 다른 위치의 페이지로 요청에 응답하고 있지만 요청자는 향후 요청 시 원래 위치를 계속 사용해야 한다.
	303	See Other	요청자가 다른 위치에 별도의 GET 요청을 하여 응답을 검색할 경우 서버는 이 코드를 표시한다. HEAD 요청 이외의 모든 요청을 다른 위치로 자동으로 전달한다.
	304	Not Modified	마지막 요청 이후 요청한 페이지는 수정되지 않았다. 서버가 이 응답을 표시하면 페이지의 콘텐츠를 표시하지 않는다. 요청자가 마지막으로 페이지를 요청한 후 페이지가 변경되지 않으면 이 응답(If-Modified-Since HTTP 헤더라고 함)을 표시하도록 서버를 구성해야 한다
	305	Use Proxy	요청자는 프록시를 사용하여 요청한 페이지만 액세스할 수 있다. 서버가 이 응답을 표시하면 요청자가 사용할 프록시를 가리키는 것이기도 하다.
	307	Temporary Redirect	현재 서버가 다른 위치의 페이지로 요청에 응답하고 있지만 요청자는 향후 요청 시 원래 위치를 계속 사용해야 한다.
	308	Permanent Redirect	
4xx (Client Error)	400	Bad Request	서버가 요청의 구문을 인식하지 못했다. 입력 파라메터가 잘못되었을 때 발생한다.
	401	Unauthorized	이 요청은 인증이 필요하다. 서버는 로그인이 필요

			한 페이지에 대해 이 요청을 제공할 수 있다. 상태 코드 이름이 권한 없음(Unauthorized)으로 되어 있지만 실제 뜻은 인증 안됨(Unauthenticated)에 더 가깝다
	402	Payment Required	서버가 요청을 거부하고 있다. 예를 들자면, 사용자가 리소스에 대한 필요 권한을 갖고 있지 않다. (401은 인증 실패, 403은 인가 실패라고 볼 수 있음)
	403	Forbidden	서버가 요청을 거부하고 있다. 예를 들자면, 사용자가 리소스에 대한 필요 권한을 갖고 있지 않다. (401은 인증 실패, 403은 인가 실패라고 볼 수 있음)
	404	Not Found	서버가 요청한 페이지(Resource)를 찾을 수 없다. 예를 들어 서버에 존재하지 않는 페이지에 대한 요청이 있을 경우 서버는 이 코드를 제공한다.
	405	Method Not Allowed	요청에 지정된 방법을 사용할 수 없다. 예를 들어 POST 방식으로 요청을 받는 서버에 GET 요청을 보내는 경우, 또는 읽기 전용 리소스에 PUT 요청을 보내는 경우에 이 코드를 제공한다
	406	Not Acceptable	요청한 페이지가 요청한 콘텐츠 특성으로 응답할 수 없다.
	407	Proxy Authentication Required	이 상태 코드는 401(권한 없음)과 비슷하지만 요청자가 프록시를 사용하여 인증해야 한다. 서버가 이 응답을 표시하면 요청자가 사용할 프록시를 가리키는 것이기도 한다.
	408	Request Timeout	서버의 요청 대기가 시간을 초과하였다
	409	Conflict	서버가 요청을 수행하는 중에 충돌이 발생했다. 서버는 응답할 때 충돌에 대한 정보를 포함해야 한다. 서버는 PUT 요청과 충돌하는 PUT 요청에 대한 응답으로 이 코드를 요청 간 차이점 목록과 함께 표시해야 한다
	410	Gone	서버는 요청한 리소스가 영구적으로 삭제되었을 때 이 응답을 표시한다. 404(찾을 수 없음) 코드와 비슷하며 이전에 있었지만 더 이상 존재하지 않는 리소스에 대해 404 대신 사용하기도 한다. 리소스가 영구적으로 이동된 경우 301을 사용하여 리소스의 새 위치를 지정해야 한다.
	411	Length	서버는 유효한 콘텐츠 길이 헤더 입력란 없이는 요

		Required	청을 수락하지 않는다.
	412	Precondition Failed	서버가 요청자가 요청 시 부과한 사전조건을 만족하지 않는다.
	413	Payload Too Large	요청이 너무 커서 서버가 처리할 수 없다.
	414	URI Too Long	요청 URI(일반적으로 URL)가 너무 길어 서버가 처리할 수 없다.
	415	Unsupported Media Type	요청이 요청한 페이지에서 지원하지 않는 형식으로 되어 있다.
	416	Range Not Satisfiable	요청이 페이지에서 처리할 수 없는 범위에 해당되는 경우 서버는 이 상태 코드를 표시한다.
	417	Expectation Failed	서버는 Expect 요청 헤더 입력란의 요구사항을 만족할 수 없다.
5xx (Server Error)	500	Internal Server Error	서버에 오류가 발생하여 요청을 수행할 수 없다.
	501	Not Implemented	서버에 요청을 수행할 수 있는 기능이 없다. 예를 들어 서버가 요청 메소드를 인식하지 못할 때 이 코드를 표시한다.
	502	Bad Gateway	서버가 게이트웨이나 프록시 역할을 하고 있거나 또는 업스트림 서버에서 잘못된 응답을 받았다.
	503	Service Unavailable	서버가 오버로드 되었거나 유지관리를 위해 다운되었기 때문에 현재 서버를 사용할 수 없다. 이는 대개 일시적인 상태이다.
	504	Gateway Timeout	서버가 게이트웨이나 프록시 역할을 하고 있거나 또는 업스트림 서버에서 제때 요청을 받지 못했다.
	505	HTTP Version Not Supported	서버가 요청에 사용된 HTTP 프로토콜 버전을 지원하지 않는다.