

PHR 기반 개인맞춤형 건강관리 서비스 플랫폼 기술 고도화  
및 실증서비스 개발연계지원

## OAuth Library

---

(주) 디케이아이테크놀로지

문서번호: PHRP\_KEIT-SD-04

Ver : 1.1

## 개정 이력

[illegible]

# 문서 목차

<b>1. 개요</b>	<b>4</b>
1.1. OAUTH LIBRARY의 목적	4
1.2. OAUTH LIBRARY의 개정	4
<b>2. OAUTH LIBRARY</b>	<b>4</b>
2.1. 사용 가이드	4
2.1.1. oauth.js 불러오기	4
2.1.2. 인증정보 설정	5
2.1.3. 인증 토큰 발급 요청	5
<b>3. OAUTH LIBRARY 요청</b>	<b>6</b>
3.1. 인증 토큰 발급(AUTHORIZATION CODE GRANT TYPE)	6
3.1.1. 인증 흐름	6
3.1.2. 인증 토큰 발급 요청	6
3.1.3. 로그인 및 권한 승인 후 Code를 이용한 인증 토큰 발급 요청	7
3.2. 인증 토큰 발급(IMPLICIT GRANT TYPE)	9
3.2.1. 인증 흐름	9
3.2.2. 인증 토큰 발급 요청	10
3.3. 인증 토큰 발급(CLIENT CREDENTIALS GRANT TYPE)	11
3.3.1. 인증흐름	11
3.3.2. 인증 토큰 발급 요청	12
3.4. 인증 토큰 발급(PASSWORD CREDENTIALS GRANT TYPE)	13
3.4.1. 인증흐름	14
3.4.2. 인증 토큰 발급 요청	14
3.5. REFRESH_TOKEN을 이용한 토큰 재발급	16
3.5.1. 인증 흐름	16
3.5.2. 인증 토큰 재발급 요청	16

## 1. 개요

OAuth Library는 “PHR 기반 개인맞춤형 건강관리 서비스 플랫폼”에서 제공하는 OPEN API, FHIR의 OAuth2.0 인증 관련 라이브러리입니다.

### 1.1. OAuth Library의 목적

OAuth Library는 “PHR 기반 개인 맞춤형 건강관리 서비스 플랫폼”을 기반으로 구성되며, 인증 토큰을 쉽고 편리하게 발급받기 위해 제공됩니다.

### 1.2. OAuth Library의 개정

OAuth Library는 필요하다고 판단되는 경우 개정이 가능하며, 그 절차는 “PHR 기반 개인맞춤형 건강관리 서비스 플랫폼 기술 고도화 및 실증서비스 개발연계지원”의 규정에 준합니다.

## 2. OAuth Library

OAuth Library를 사용하기 위해서 먼저 PHR 지원 포탈을 통해 서비스 사용 신청을 하고 관리자의 승인을 받아 인증정보를 발급받아야 합니다.

### 2.1. 사용 가이드

#### 2.1.1. oauth.js 불러오기

oauth.js 파일에는 [Authorization Code Grant](#), [Implicit Grant](#), [Client Credentials Grant](#), [Password Credentials Grant](#), [Refresh Grant](#) 방식의 인증 토큰 요청을 지원하는 함수가 정의되어 있습니다. OAuth Library를 쓰기 위해서는 인증 토큰을 요청하는 곳에 oauth.js의 파일을 불러와야 합니다. Node.js 환경에서 쓰기 위해서는 별도의 HTTP 통신라이브러리 Axios의 설치가 필요합니다.

##### ⊙ JavaScript 불러오기 예시

```
<html>
  <head>
    <script src="./oauth.js"> </script>
  </head>
  <body>
  </body>
</html>
```

##### ⊙ Node.js 불러오기 예시

```
var OAuth = require("./oauth");
```

- ⊙ Node.js 터미널에서 Aixos 설치

```
npm install --save axios
```

### 2.1.2. 인증정보 설정

PHR 지원 포탈을 통해 승인을 받은 인증정보는 인증 토큰 요청을 하기 위해서 꼭 필요한 정보입니다. 마찬가지로 인증 토큰 요청하는 곳에 설정을 해주셔야 합니다.

- ⊙ Property 설정

프로퍼티	타입	필수여부	설명
client_id	string	Y	승인 완료된 서비스 아이디
client_secret	string	Y	승인 완료된 클라이언트 시크릿
redirect_uri	string	Y	서비스 사용 신청 시 입력한 redirect_uri
authorize	string	Y	인증요청 URI({PHR 인증서버}/oauth/authorize)
token	string	Y	토큰요청 URI({PHR 인증서버}/oauth/token)
scope	string	Y	플랫폼 이용 시 사용할 권한 범위(phr.read phr.write)
state	string	N	클라이언트에서 서버의 응답 유효성을 확인하기 위한 값

- ⊙ 설정 예시

```
var config = {
  client_id: "oauthLibrary",
  client_secret: "23432fc-2341-15b3-96c3-23ce43ac6866",
  redirect_uri: "http://127.0.0.1:8080/callback.html",
  authorize: "https://oauth.redwoodhealth.kr/oauth/authorize",
  token: "https://oauth.redwoodhealth.kr/oauth/token",
  scope: "phr.read phr.write",
  state: "1234"
};
```

### 2.1.3. 인증 토큰 발급 요청

oauth.js 불러오기, 인증정보 설정이 완료되면 인증 토큰 발급을 요청하실 수 있습니다.

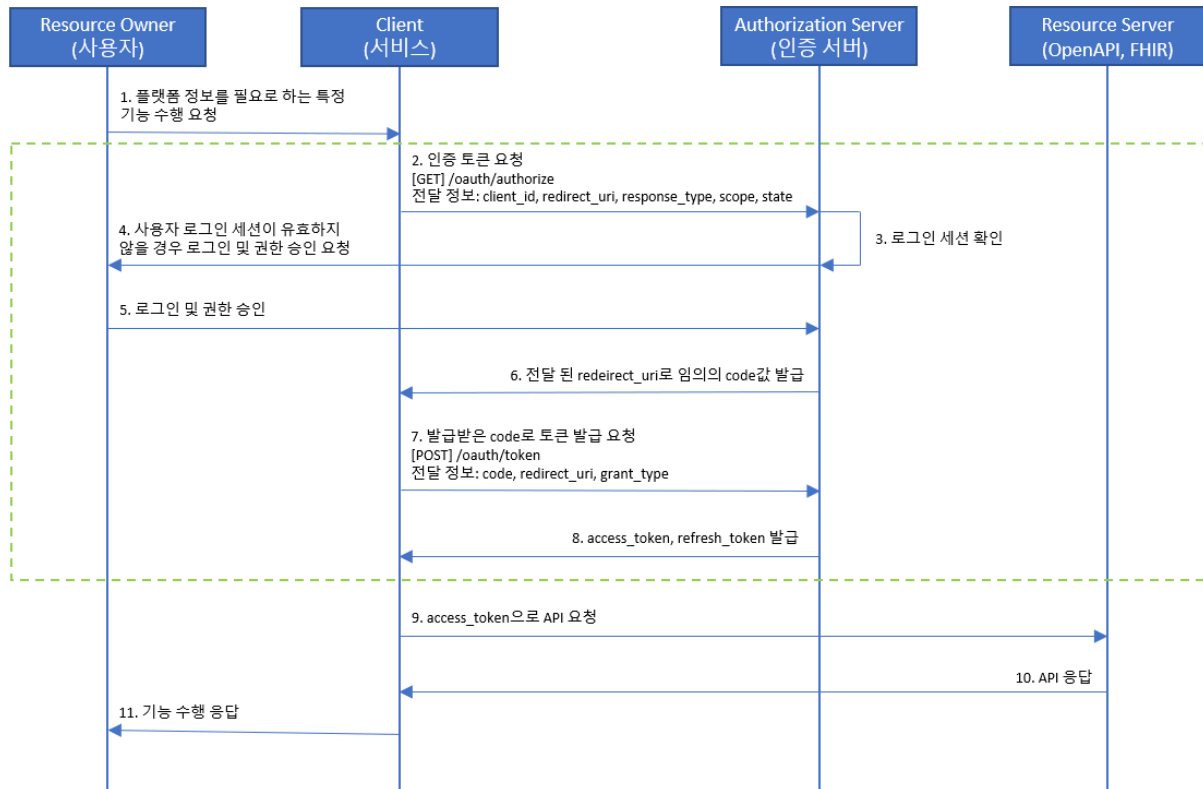
[Authorization Code Grant](#), [Implicit Grant](#), [Client Credentials Grant](#), [Password Credentials Grant](#), [Refresh Grant](#) 방식으로 요청하실 수 있습니다.

### 3. OAuth Library 요청

#### 3.1. 인증 토큰 발급(Authorization Code Grant Type)

OAuth Library 인증 방식 중 **Authorization Code Grant Type**을 이용한 경우입니다.

##### 3.1.1. 인증 흐름



##### 3.1.2. 인증 토큰 발급 요청

⊙ 요청 변수 Parameters

프로퍼티	타입	필수여부	설명
client_id	string	Y	<a href="#">설정된 인증정보</a>
client_secret	string	Y	<a href="#">설정된 인증정보</a>
redirect_uri	string	Y	<a href="#">설정된 인증정보</a>
authorize	string	Y	<a href="#">설정된 인증정보</a>
token	string	Y	<a href="#">설정된 인증정보</a>
scope	string	Y	<a href="#">설정된 인증정보</a>
state	string	N	<a href="#">설정된 인증정보</a>

### ⊙ 요청 예시

*/\*설정한 인증정보\*/*

```
var config = {
  client_id: "oauthLibrary",
  client_secret: "23432fc-2341-15b3-96c3-23ce43ac6866",
  redirect_uri: "http://127.0.0.1:8080/callback.html",
  authorize: "https://oauth.redwoodhealth.kr/oauth/authorize",
  token: "https://oauth.redwoodhealth.kr/oauth/token",
  scope: "phr.read phr.write",
  state: "1234"
};
```

### ⊙ 응답 예시

HTTP/1.1 302 found

Location: http://localhost:8080/login

- ➔ 인증서버에 로그인 세션이 존재하지 않을 경우 사용자에게 login 및 권한 승인을 요청하는 페이지가 전달됩니다.
- ➔ 유효한 세션이 존재할 경우 입력한 redirect\_uri로 authorization\_code 값이 전달됩니다.

예)

① 127.0.0.1:7000/phrtest/receiveCode.html?code=2uTYqZ8&state=1234

해당 코드를 이용하여 access\_token 및 refresh\_token을 요청할 수 있게 됩니다.

### ⊙ 함수 및 사용 방법

함수명	사용 방법
authorizationCode	OAuth.authorizationCode(설정한 인증정보)

### ⊙ 사용 예시

*/\*config: 설정한 인증정보\*/*

```
OAuth.authorizationCode(config);
```

## 3.1.3. 로그인 및 권한 승인 후 Code를 이용한 인증 토큰 발급 요청

⊙ HTTP Headers

파라미터	타입	필수여부	설명
Content-Type	string	Y	application/x-www-form-urlencoded
Authorization	string	Y	<a href="#">설정된 인증정보</a>

⊙ 요청 변수 Parameters

파라미터	타입	필수여부	설명
code	string	Y	3.1.2. 인증 토큰 발급 요청 시 발급받은 code값

⊙ 요청 예시

[/\\*설정된 인증정보\\*/](#)

```
var config = {
  client_id: "oauthLibrary",
  client_secret: "23432fc-2341-15b3-96c3-23ce43ac6866",
  redirect_uri: "http://127.0.0.1:8080/callback.html",
  authorize: "https://oauth.redwoodhealth.kr/oauth/authorize",
  token: "https://oauth.redwoodhealth.kr/oauth/token",
  scope: "phr.read phr.write",
  state: "1234"
};
```

[/\\*요청 Parameters\\*/](#)

```
var code = "2uTYqZ";
```

⊙ 응답 예시

HTTP/1.1 200 OK

Content-Type: application/json;charset=UTF-8

```
{
  access_token: "be90a5f8-2be1-47ce-9cb9-02094cca9839"
  aud: "oauthLibrary"
  expires_in: 35999
  iss: "RedwoodPlatform"
  jti: "be44ea9c-3d0e-4c0d-bc13-6c404d3123c4f"
  name: "테스트사용자"
  refresh_token: "be90a5f8-2be1-47ce-9cb9-02094cca1234"
  scope: "phr.write phr.read"
  sub: "test"
```



```
token_type: "bearer"
}
```

⊙ 함수 및 사용 방법

함수명	사용 방법
getToken	OAuth.getToken(설정된 인증정보, 요청 Parameters)

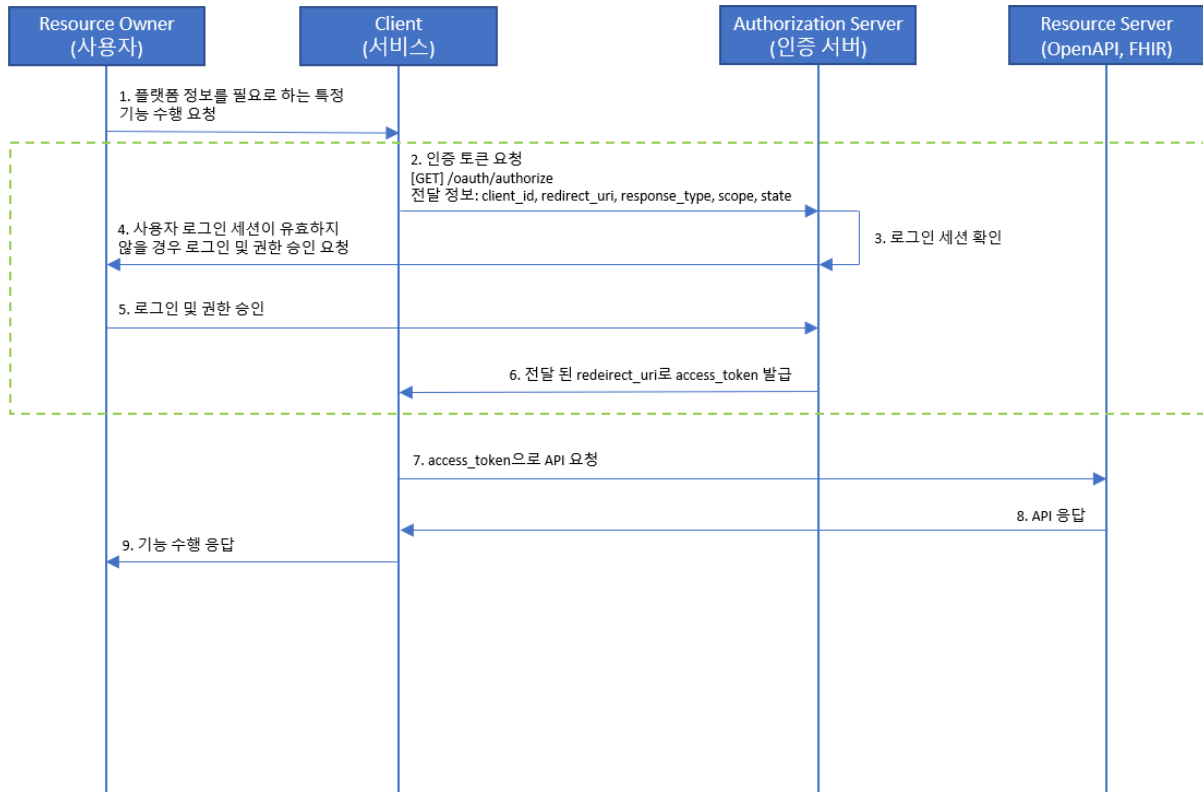
⊙ 사용 예시

```
/*config: 설정한 인증정보, code: 요청 Parameters*/
OAuth.getToken(config, code)
.then(response => {
    console.log("response", response);
})
.catch(error => {
    console.log("error", error);
});
```

## 3.2. 인증 토큰 발급(Implicit Grant Type)

OAuth Library 인증 방식 중 **Implicit Grant Type**을 이용한 경우입니다.

### 3.2.1. 인증 흐름



### 3.2.2. 인증 토큰 발급 요청

#### ◎ 요청 변수 Parameters

프로퍼티	타입	필수여부	설명
client_id	string	Y	<a href="#">설정된 인증정보</a>
client_secret	string	Y	<a href="#">설정된 인증정보</a>
redirect_uri	string	Y	<a href="#">설정된 인증정보</a>
authorize	string	Y	<a href="#">설정된 인증정보</a>
token	string	Y	<a href="#">설정된 인증정보</a>
scope	string	Y	<a href="#">설정된 인증정보</a>
state	string	N	<a href="#">설정된 인증정보</a>

#### ◎ 요청 예시

*/\*설정된 인증정보\*/*

```

var config = {
  client_id: "oauthLibrary",
  client_secret: "23432fc-2341-15b3-96c3-23ce43ac6866",
  redirect_uri: "http://127.0.0.1:8080/callback.html",
  authorize: "https://oauth.redwoodhealth.kr/oauth/authorize",
  token: "https://oauth.redwoodhealth.kr/oauth/token",

```

```
scope: "phr.read phr.write",
state: "1234"
};
```

#### ⊙ 응답 예시

HTTP/1.1 302 found

Location: http://localhost:8080/login

- ➔ 인증서버에 로그인 세션이 존재하지 않을 경우 사용자에게 login 및 권한 승인을 요청하는 페이지가 전달됩니다.
- ➔ 유효한 세션이 존재할 경우 입력한 redirect\_uri로 access\_token 값이 전달됩니다.

예)

```
receiveCode.html#access_token=f8e44056-f8b9-4958-b53c-a54f2c636357&token_type=bearer&expires_in=34812
```

redirect\_uri 뒤에 연결된 '#' 이하로 전달된 access\_token을 이용하여 api를 호출할 수 있습니다.

#### ⊙ 함수 및 사용 방법

함수명	사용 방법
implicit	OAuth.implicit(설정된 인증정보)

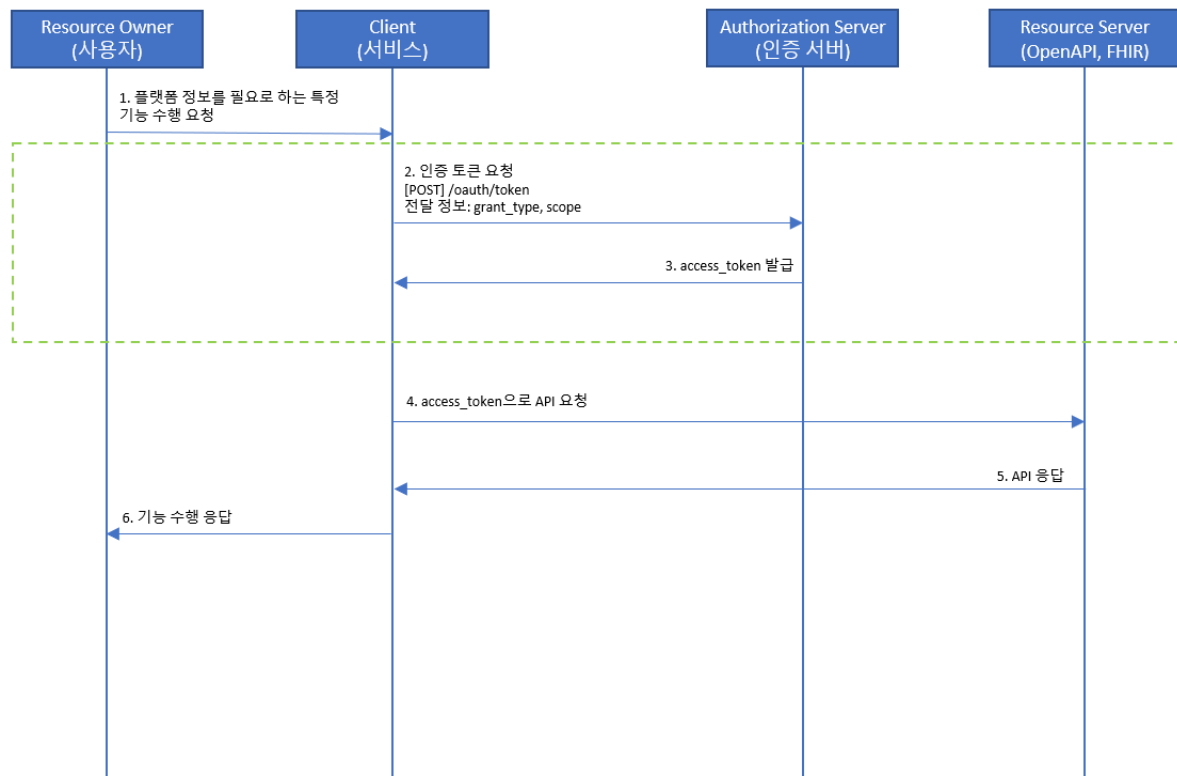
#### ⊙ 사용 예시

```
/*config: 설정한 인증정보*/
OAuth.implicit(config);
```

### 3.3. 인증 토큰 발급(Client Credentials Grant Type)

OAuth Library 인증 방식 중 **Client Credentials Grant Type**을 이용한 경우입니다.

#### 3.3.1. 인증 흐름



### 3.3.2. 인증 토큰 발급 요청

#### ⊙ 요청 변수 Parameters

프로퍼티	타입	필수여부	설명
client_id	string	Y	<a href="#">설정된 인증정보</a>
client_secret	string	Y	<a href="#">설정된 인증정보</a>
redirect_uri	string	Y	<a href="#">설정된 인증정보</a>
authorize	string	Y	<a href="#">설정된 인증정보</a>
token	string	Y	<a href="#">설정된 인증정보</a>
scope	string	Y	<a href="#">설정된 인증정보</a>
state	string	N	<a href="#">설정된 인증정보</a>

#### ⊙ 요청 예시

*/\*설정된 인증정보\*/*

```

var config = {
  client_id: "oauthLibrary",
  client_secret: "23432fc-2341-15b3-96c3-23ce43ac6866",
  redirect_uri: "http://127.0.0.1:8080/callback.html",
  authorize: "https://oauth.redwoodhealth.kr/oauth/authorize",
  token: "https://oauth.redwoodhealth.kr/oauth/token",

```

```
scope: "phr.read phr.write",
state: "1234"
};
```

#### ⊙ 응답 예시

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

{
  access_token: "be90a5f8-2be1-47ce-9cb9-02094cca9839"
  aud: "oauthLibrary"
  expires_in: 35999
  iss: "RedwoodPlatform"
  jti: "be44ea9c-3d0e-4c0d-bc13-6c404d3123c4f"
  scope: "phr.write phr.read"
  sub: "test"
  token_type: "bearer"
}
```

#### ⊙ 함수 및 사용 방법

함수명	사용 방법
clientCredentials	OAuth.clientCredentials(설정된 인증정보)

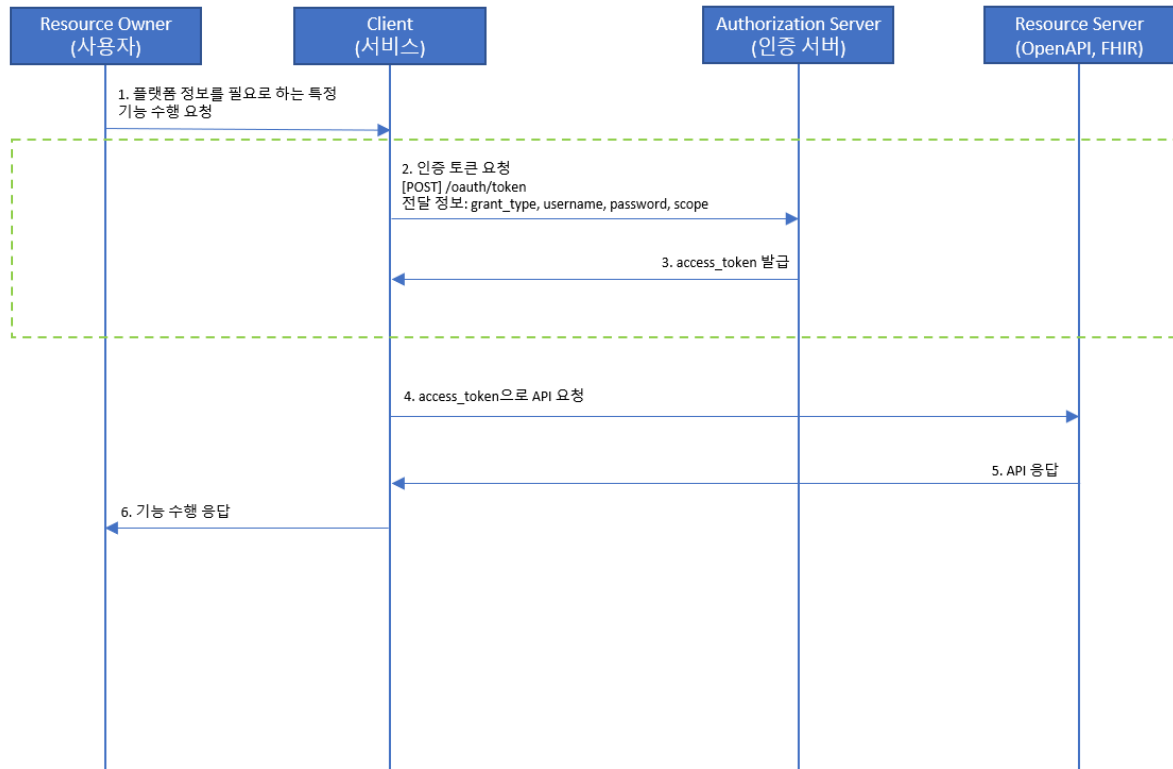
#### ⊙ 사용 예시

```
/*config: 설정한 인증정보*/
OAuth.clientCredentials(config)
.then(response => {
  console.log("response", response);
})
.catch(error => {
  console.log("error", error);
});
```

### 3.4. 인증 토큰 발급(Password Credentials Grant Type)

OAuth Library 인증 방식 중 **Password Credentials Grant Type**을 이용한 경우입니다.

### 3.4.1. 인증흐름



### 3.4.2. 인증 토큰 발급 요청

#### ⊙ HTTP Headers

파라미터	타입	필수여부	설명
Content-Type	string	Y	application/x-www-form-urlencoded
Authorization	string	Y	<u>설정된 인증정보</u>

#### ⊙ 요청 변수 Parameters

파라미터	타입	필수여부	설명
userId	string	Y	사용자 아이디
userPw	string	Y	사용자 비밀번호(sha256 인코딩)

#### ⊙ 요청 예시

/\*설정된 인증정보\*/

```

var config = {
  client_id: "oauthLibrary",
  client_secret: "23432fc-2341-15b3-96c3-23ce43ac6866",
  redirect_uri: "http://127.0.0.1:8080/callback.html",

```

```

    authorize: "https://oauth.redwoodhealth.kr/oauth/authorize",
    token: "https://oauth.redwoodhealth.kr/oauth/token",
    scope: "phr.read phr.write",
    state: "1234"
  };

  /*요청 Parameters*/
  var userInfo = {
    userId: "test",
    userPw: "2fa1ecf9826474fe89112323sdwd3r5t8e19cd118"
  };

```

#### ⊙ 응답 예시

```

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

{
  access_token: "be90a5f8-2be1-47ce-9cb9-02094cca9839"
  aud: "oauthLibrary"
  expires_in: 35999
  iss: "RedwoodPlatform"
  jti: "be44ea9c-3d0e-4c0d-bc13-6c404d3123c4f"
  name: "테스트사용자"
  refresh_token: "be90a5f8-2be1-47ce-9cb9-02094cca1234"
  scope: "phr.write phr.read"
  sub: "test"
  token_type: "bearer"
}

```

#### ⊙ 함수 및 사용 방법

함수명	사용 방법
passwordCredentials	OAuth.passwordCredentials(설정된 인증정보, 요청 Parameters)

#### ⊙ 사용 예시

```

/*config: 설정한 인증정보, userInfo: 요청 Parameters*/
OAuth.passwordCredentials(config, userInfo)
  .then(response => {

```

```

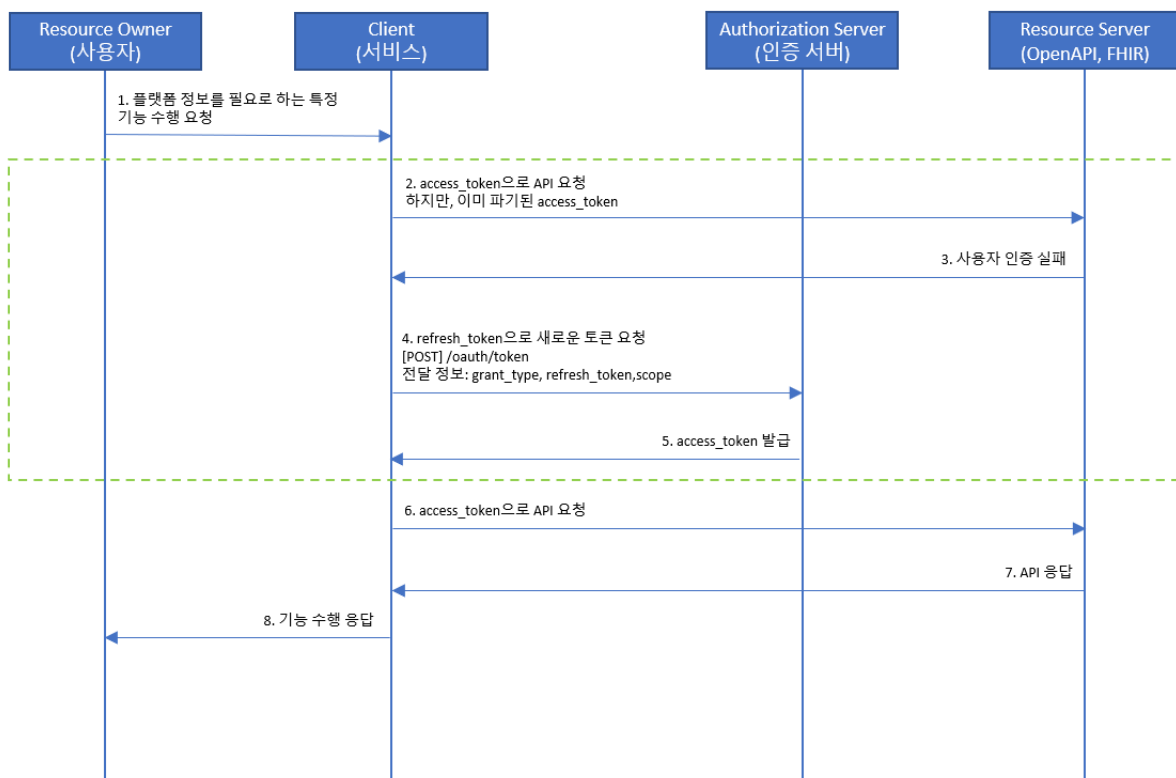
        console.log("response", response);
    })
    .catch(error => {
        console.log("error", error);
    });

```

### 3.5. Refresh\_token을 이용한 토큰 재발급

OAuth Library 인증 방식 중 **Refresh Grant Type**을 이용한 경우입니다.

#### 3.5.1. 인증 흐름



#### 3.5.2. 인증 토큰 재발급 요청

##### ⊙ HTTP Headers

파라미터	타입	필수여부	설명
Content-Type	string	Y	application/x-www-form-urlencoded
Authorization	string	Y	<u>설정된 인증정보</u>

##### ⊙ 요청 변수 Parameters



파라미터	타입	필수여부	설명
refresh_token	string	Y	발급받은 refresh_token값 입력

⊙ 요청 예시

*/\*설정된 인증정보\*/*

```
var config = {
  client_id: "oauthLibrary",
  client_secret: "23432fc-2341-15b3-96c3-23ce43ac6866",
  redirect_uri: "http://127.0.0.1:8080/callback.html",
  authorize: "https://oauth.redwoodhealth.kr/oauth/authorize",
  token: "https://oauth.redwoodhealth.kr/oauth/token",
  scope: "phr.read phr.write",
  state: "1234"
};
```

*/\*요청 Parameters\*/*

```
var refresh_token= "be90a5f8-2be1-47ce-9cb9-02094cca1234";
```

⊙ 응답 예시

HTTP/1.1 200 OK

Content-Type: application/json;charset=UTF-8

```
{
  access_token: "be90a5f8-2be1-47ce-9cb9-02094cca9839"
  aud: "oauthLibrary"
  expires_in: 35999
  iss: "RedwoodPlatform"
  jti: "be44ea9c-3d0e-4c0d-bc13-6c404d3123c4f"
  name: "테스트사용자"
  refresh_token: "be90a5f8-2be1-47ce-9cb9-02094cca1234"
  scope: "phr.write phr.read"
  sub: "test"
  token_type: "bearer"
}
```

⊙ 함수 및 사용 방법

함수명	사용 방법
refresh	OAuth.refresh(설정된 인증정보, 요청 Parameters)

⦿ 사용 예시

```
/*config: 설정한 인증정보, refresh_token: 요청 Parameters*/
OAuth.refresh(config, refresh_token)
.then(response => {
    console.log("response", response);
})
.catch(error => {
    console.log("error", error);
});
```