

OPEN API 코드 리뷰

-네이버 아이디로 로그인 (네아로) API-

쌍용교육센터

문성재, 강태훈, 김선호, 안동현, 최유미

1. OPEN API 개요

'네이버 아이디로 로그인 서비스'(줄여서 '네아로', 이하 본문에서 '네아로' 로 표기)은 OAuth2.0 (Open Authorization, 로그인시 ID/PW 인증 이후, 권한을 체크하는 오픈 소스 라이브러리)기반의 사용자 인증 기능을 제공하여 네이버가 아닌 다른 서비스에서 네이버의 사용자 인증 기능을 이용하여 별도의 아이디나 비밀번호 등 회원의 정보 요구나 입력 없이 네이버 아이디로 간단하게 사이트를 이용할 수 있는 무료 API 입니다.

2. '네아로' OPEN API 명세

'네아로'는 네이버 로그인 인증 요청 API , 접근 토큰 발급/갱신/삭제 요청 API 로 구성되어 있습니다. 인증 요청 API 는 웹 또는 앱에서 네이버 로그인 화면을 띄우는 API 이며 사용자가 회원 인증 성공시 API 로부터 받은 코드 값을 이용하여 접근 토큰 발급 요청 API 를 호출 한 뒤, 요청 API 를 이용해 받은 접근 토큰 값으로 사용자의 프로필 조회 및 여러가지 로그인 오픈 API 를 호출하는데 사용할 수 있습니다.

3. 사용 요구 사항 및 필요 기술들

네아로 API 신청(Client ID 값, Client Secret 값), JDK 8.0 이상, 톰캣 서버, jQuery, Javascript, CSS, Bootstrap3, JSP, json.simple-1.1.1.jar, 서비스 URL(루프백 주소 사용 가능), 콜백(루프백 주소 사용 가능) URL, MySQL 데이터베이스(선택사항)

4. 전용키 발급 절차

4-1. 네아로 API 를 활용할 애플리케이션 등록

내 애플리케이션 애플리케이션 등록 Clova Platform Console β API 제휴 신청 계정 설정	애플리케이션 이름	애플리케이션 이름 ① • 네이버 아이디로 로그인할 때 사용자에게 표시되는 이름이므로 가급적 10자 이내의 간결한 이름을 사용해주세요. • 40자 이내의 영문, 한글, 숫자, 공백문자, "-", "_", "." 만 입력 가능합니다.
	사용 API	선택하세요. \blacktriangledown 네아로 (네이버 아이디로 로그인) <input checked="" type="checkbox"/> 사용자 고유 식별자(기본 정보) <input type="checkbox"/> 이메일 (추가권한) <input type="checkbox"/> 프로필 사진 (추가권한) <input type="checkbox"/> 성별 (추가권한) <input type="checkbox"/> 생일 (추가권한) <input type="checkbox"/> 회원이름 (추가권한) <input type="checkbox"/> 연령대 (추가권한) <input type="checkbox"/> 별명 (추가권한) [공지] 네아로 가이드 변경사항을 확인하세요
		환경 추가 \blacktriangledown

네아로 API 를 사용하기 위해선 네이버 개발자센터에 접속하여, 애플리케이션을 등록해야 합니다. 상단의 Application 탭을 선택하면 새로운 애플리케이션을 등록할 수 있습니다. 사용할 애플리케이션 이름과 사용할 API, 수집할 정보 등을 선택하고 애플리케이션을 등록하면 API 를 신청할 수 있습니다. 여기서, 네아로 API 를 선택하고, 수집 정보를 전체로 선택하고 진행합니다.

4-2. 네아로 API 를 활용할 환경 설정 및 서비스, 콜백 URL 등록

내 애플리케이션 애플리케이션 등록 Clova Platform Console β API 제휴 신청 계정 설정	로그인 오픈 API 서비스 환경 ①	[공지] 네아로 가이드 변경사항을 확인하세요 환경 추가 \blacktriangledown PC 웹 × \wedge 서비스 URL <input type="text" value="http://localhost:8080/NaverLogin/index.jsp"/> ✓ 서비스 URL 예시: (O) http://naver.com (X) http://www.naver.com 서비스 URL 값이 잘못 입력되어 있으면 정확한 값으로 수정하실 때 까지 네이버 로그인 사용이 일시적으로 제한됩니다. 불법/음란성 사이트 등 이용약관에 위배되는 사이트의 경우, 이용이 제한될 수 있습니다. 서비스하려는 사이트 URL과 동일한 사이트 URL로 해주셔야 네이버 아이디로 로그인 버튼이 노출됩니다. 네이버아이디로그인 Callback URL (최대 5개) <input type="text" value="Callback URL 예시: http://YOUR_DOMAIN/Api/Member/Oauth2C"/> ① 텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다. Callback URL은 네이버 로그인 후 이동할 페이지 URL입니다. Callback URL 값이 잘못 입력되어 있으면 정확한 값으로 수정하실 때 까지 네이버 로그인 사용이 일시적으로 제한됩니다. 입력한 주소와 다른 Callback URL로 리다이렉트 될 경우, 이용이 제한될 수 있습니다.
---	---------------------	--

애플리케이션을 등록할 때 서비스에 사용할 URL 과 API 를 호출하여 접근 토큰을 콜백 받을 콜백 주소를 등록해야 합니다. 여기선 네이로 로그인 연동 인증 API 를 호출할 서비스 URL 로 로컬 호스트 밑에 있는 index.jsp 파일의 위치를 연결하였고, 인증 토큰 발급 및 프로필 조회 콜백 url 로 navelogin.jsp 를 연결하였습니다. 특정 JSP 파일을 URL 로 사용하기 위해선 전체 로컬 호스트 주소를 입력해야 하며, 파일의 확장자까지 전부 입력해야 합니다.

4-3. API 활용을 위한 발급키 확인

내 애플리케이션

안전모

푸드릿

NearFarm

SIST 회원관리 프로그램

SIST 회원관리 프로그램2

애플리케이션 등록

Clova Platform Console *β*

API 제휴 신청

계정 설정

SIST 회원관리 프로그램2

개요

API 설정

네아로 검수상태

멤버관리

로그인 통계

API 통계

Playground (Beta)

애플리케이션 정보

Client ID

Client Secret

네이버 아이디로 로그인

로그 이미지

개발 상태

애플리케이션을 등록하고 나면, Client ID 와 Client Secret 두 가지 키가 필요로 합니다. 네아로 API 를 호출하기 위해선 JSP 코드상에서 서비스 URL, 콜백 URL, 클라이언트 아이디, 클라이언트 시크릿 키 모두 입력해야 합니다.

4-4. 관리자 및 테스트 아이디 등록

내 애플리케이션

안전모

푸드릿

NearFarm

SIST 회원관리 프로그램

SIST 회원관리 프로그램2

애플리케이션 등록

Clova Platform Console *β*

API 제휴 신청

계정 설정

SIST 회원관리 프로그램2

개요

API 설정

네아로 검수상태

멤버관리

로그인 통계

API 통계

Playground (Beta)

관리자 ID 등록 (최대 3개)

테스터 ID 등록 (최대 20개)

관리자 ID는 애플리케이션이 '개발 중' 상태일 때 해당 애플리케이션에 로그인하여 테스트할 수 있는 ID입니다. 최대 3개까지 등록 가능하며, 관리자 아이디는 등록하지 않아도 로그인이 가능합니다.

네아로 API 를 요청하면 네이로 서버 측에서 아이디를 통해 이미 생성된 고유한 8 자리 식별자 코드를 제공합니다. 정상적으로 개발이 이루어지고 있는지 확인하기 위해 이 식별자 코드를 테스트 단계에서 사용하기 위해선 테스트 아이디를 등록해야 하며, 총 20 개까지 등록 가능합니다. 관리자 아이디는 API 를 신청한 아이디가 자동으로 부여받게 됩니다.

5. 네아로 호출 API 종류

5-1. 네이버 아이디로 로그인 인증 요청 예시

4.1. 네이버 아이디로 로그인 인증 요청

네이버 아이디로 로그인 인증 요청 API를 호출했을 때 사용자가 네이버로 로그인하지 않은 상태이면 네이버 로그인 화면으로 이동하고, 사용자가 네이버에 로그인한 상태이면 기본 정보 제공 동의 확인 화면으로 이동합니다. 네이버 로그인과 정보 제공 동의 과정이 완료되면 콜백 URL에 code값과 state 값이 URL 문자열로 전송됩니다. code 값은 접근 토큰 발급 요청에 사용됩니다. API 요청 실패시에는 에러 코드와 에러 메시지가 전송됩니다.

- API 요청 성공시 : `http://콜백URL/redirect?code={code값}&state={state값}`
- API 요청 실패시 : `http://콜백URL/redirect?state={state값}&error={에러코드값}&error_description={에러메시지}`

필드	타입	설명
code	string	네이버 아이디로 로그인 인증에 성공하면 반환받는 인증 코드, 접근 토큰(access token) 발급에 사용
state	string	사이트 간 요청 위조 공격을 방지하기 위해 애플리케이션에서 생성한 상태 토큰으로 URL 인코딩을 적용한 값
error	string	네이버 아이디로 로그인 인증에 실패하면 반환받는 에러 코드
error_description	string	네이버 아이디로 로그인 인증에 실패하면 반환받는 에러 메시지

네이버 아이디로 로그인 기능을 요청하기 위해서 앞서 발급받은 API 키 값과 Secret 값이 필요합니다. 네아로 API 를 호출했을 때 정상적으로 api 호출이 이루어지면, GET 방식으로 상태 값과 인증 코드를 콜백 URL 로 전달합니다. 만약, 정상 처리가 이루어지지 않으면, error 코드를 전달합니다. 여기서 서비스 URL 은 index.jsp 에 해당하고, 콜백 URL 은 naverlogin.jsp 에 해당합니다.

5-2. 네이버 아이디로 로그인 프로필 조회 요청 예시

요청 예시

```
GET v1/nid/me HTTP/1.1
Host: openapi.naver.com
User-Agent: curl/7.43.0
Accept: */*
Content-Type: application/xml
Authorization: Bearer {네이버 아이디로 로그인 인증 후 받은 접근 토큰 값}
```

응답 예시

```
{
  "resultcode": "00",
  "message": "success",
  "response": {
    "email": "openapi@naver.com",
    "nickname": "OpenAPI",
    "profile_image": "https://ssl.pstatic.net/static/pwe/address/nodata_33x33.gif",
    "age": "40-49",
    "gender": "F",
    "id": "32742776",
    "name": "오픈 API",
    "birthday": "10-01"
  }
}
```

네아로 api 호출이 정상적으로 이루어지면, 전달 받은 접근 토큰을 이용해 프로필 조회 api를 호출할 수 있습니다. 프로필 조회 요청 코드도 마찬가지로 naverlogin.jsp 파일에서 작성되며, 위 예시처럼 GET 방식으로 api를 요청하면, JSON 형식으로 회원의 프로필 데이터를 돌려줍니다. 반환받은 JSON 데이터를 분석해서 유니크한 식별코드를 제공받으며 session에 저장하여 회원을 구분하고, DB에 저장할 수 있습니다.

6. 네아로 테스트 코드(개발 부분)

index.jsp

```
<%
//네아로 콜백 URL 인코딩 필요.
String redirectURI = URLEncoder.encode("http://localhost:8080/NaverLogin/Login/naverlogin.jsp", "UTF-8");
//사이트 위조 공격 방지를 위해 난수 생성.
SecureRandom random = new SecureRandom();
String state = new BigInteger(130, random).toString();
//실제 요청할 api 주소.
String apiURL = "https://nid.naver.com/oauth2.0/authorize?response_type=code";
//GET 방식으로 요청.
apiURL += "&client_id=클라이언트 ID"; //애플리케이션 클라이언트 아이디값;
apiURL += "&redirect_uri=" + redirectURI;
apiURL += "&state=" + state;
%>

<div id="naver_id_login" style="display: inline-block;">
<a href="<%=apiURL%>"></a></div>
```

naverlogin.jsp

```
<%
request.setCharacterEncoding("UTF-8");
String contextRoute = request.getContextPath();
String code = request.getParameter("code");
String state = request.getParameter("state");
//콜백 URL 인코딩 필요.
String redirectURI = URLEncoder.encode("http://localhost:8080/NaverLogin/Login/naverlogin.jsp", "UTF-8");
String apiURL;
//전달 받은 접근 상태 코드를 통해 인증 값 호출.
apiURL = "https://nid.naver.com/oauth2.0/token?grant_type=authorization_code&";
apiURL += "client_id=클라이언트 ID"; //애플리케이션 클라이언트 아이디값";
apiURL += "&client_secret=클라이언트 Secret"; //애플리케이션 클라이언트 시크릿값";
apiURL += "&redirect_uri=" + redirectURI;
apiURL += "&code=" + code;
apiURL += "&state=" + state;
String access_token = "";
String refresh_token = "";
StringBuffer res = new StringBuffer();
try {
    URL url = new URL(apiURL);
    HttpURLConnection con = (HttpURLConnection) url.openConnection();
    con.setRequestMethod("GET");
    int responseCode = con.getResponseCode();
    System.out.println("responseCode=" + responseCode);
    BufferedReader br;
    if (responseCode == 200) { // 정상 호출
        br = new BufferedReader(new InputStreamReader(con.getInputStream()));
    } else { // 에러 발생
        br = new BufferedReader(new InputStreamReader(con.getErrorStream()));
    }
    String inputLine;
    int temp = 0;
    while ((inputLine = br.readLine()) != null) {
        res.append(inputLine);
        if (temp == 1) {
            //데이터 저장.
            access_token = inputLine;
            break;
        }
        ++temp;
    }
    br.close();
    if (responseCode == 200) {
```

```

//에러 없이 정상 작동.
} else {
//에러 발생시 바로 리디렉션 실행.
response.sendRedirect(contextRoute + "/Login/loginfailform.jsp");
}
} catch (Exception e) {
System.out.println(e);
}

String str = access_token.replace("access_token", "");
access_token = str.replace(".", "");
str = access_token.replace("\\", "");
access_token = str.replace(",", "");

String token = access_token; // 네이버 로그인 접근 토큰
String header = "Bearer " + token; // Bearer 다음에 공백 추가
//System.out.println("final header : " + header);

try {
apiURL = "https://openapi.naver.com/v1/nid/me";
URL url = new URL(apiURL);
URLConnection con = (URLConnection) url.openConnection();
con.setRequestMethod("GET");
con.setRequestProperty("Authorization", header);
int responseCode = con.getResponseCode();
BufferedReader br;
if (responseCode == 200) { // 정상 호출
br = new BufferedReader(new InputStreamReader(con.getInputStream()));
} else { // 에러 발생
br = new BufferedReader(new InputStreamReader(con.getErrorStream()));
}

String inputLine;
String ret = ""; //json 포맷으로 만들기 위한 임시 변수 선언.
while ((inputLine = br.readLine()) != null) {
ret += inputLine; //json 타입 받아오기.
}

//문자열 형식 데이터를 json.simple 라이브러리를 통해 JSON 포맷으로 변환.
JSONParser jsonParser = new JSONParser();
Map json = (Map) jsonParser.parse(ret);
JSONObject jsonObj = (JSONObject) json.get("response");
//JSON 형식 데이터를 String 형으로 변환.

String birthday = (String) jsonObj.get("birthday") == null ? "" : (String) jsonObj.get("birthday");
String profile_image = (String) jsonObj.get("profile_image") == null ? "" : (String) jsonObj.get("profile_image");
String gender = (String) jsonObj.get("gender") == null ? "" : (String) jsonObj.get("gender");
String nickname = (String) jsonObj.get("nickname") == null ? "" : (String) jsonObj.get("nickname");
String id = (String) jsonObj.get("id") == null ? "" : (String) jsonObj.get("id");
String age = (String) jsonObj.get("age") == null ? "" : (String) jsonObj.get("age");
String email = (String) jsonObj.get("email") == null ? "" : (String) jsonObj.get("email");

```

```

//네이버 프로필 객체 생성.
NaverResponse naver = new NaverResponse.NaverProfileBuilder()
    .birthday(birthday)
    .profile_image(profile_image)
    .gender(gender).nickname(nickname)
    .id(id)
    .age(age)
    .email(email)
    .build();
LoginDAO dao = new LoginDAO();
//DB 에 데이터가 존재하는지 확인.
String selectRet = dao.select(id);
int naverRet = 0;
if (selectRet == null) {
    //DB 에 데이터가 존재하지 않음.
    Login l = new Login.LoginBuilder().setNaver(naver).build();
    //DB 에 프로필 정보 저장.
    naverRet = dao.insert(l);
}
//DB 에 데이터가 존재함.
//세션에 네아로 로그인 정보 저장.
session.setAttribute("naverInfo", naver);
//GET 방식 전달.
response.sendRedirect(contextRoute + "/success.jsp?naver=" + naverRet);

} catch (Exception e) {
    System.out.println(e);
}
%>

```

success.jsp

```

if (naverSess != null && naverSess instanceof NaverResponse){
    NaverResponse naverResponse = (NaverResponse) naverSess;
    id = naverResponse.getEmail();
    name_ = naverResponse.getNickname();
    String uniqueAdmin = (String)naverResponse.getId();
    logoutAtag = "https://nid.naver.com/nidlogin.logout";
    if(uniqueAdmin.equals("관리자 유니크 키")){
        admin = "관리자로 로그인 중입니다.";
    }
}

```

7. 네아로 테스트 코드(DB 부분)

테이블 구성

```

/*회원 아이디, 비밀번호*/
CREATE TABLE LOGIN(
    id VARCHAR(20),
    pw VARCHAR(20)
);
ALTER TABLE LOGIN
ADD CONSTRAINT login_id_pk PRIMARY KEY (id);

/*기존 회원 정보*/
CREATE TABLE USERS (
    idx VARCHAR(10) NOT NULL,
    name_ VARCHAR(50) NULL,
    email VARCHAR(100) NULL,
    nickname VARCHAR(50) NULL,
    phone VARCHAR(20) NULL,
    regDate DATETIME NULL,
    id VARCHAR(20)
);
ALTER TABLE USERS
ADD CONSTRAINT users_idx_pk PRIMARY KEY (idx);

ALTER TABLE USERS
ADD CONSTRAINT users_id_fk FOREIGN KEY (id) REFERENCES LOGIN (id);

/*네아로 로그인 정보*/
CREATE TABLE `SNS_INFO` (
    idx VARCHAR(10) NOT NULL,
    email VARCHAR(255) NOT NULL,
    age VARCHAR(10) NULL,
    nickname VARCHAR(255) NULL,
    profile_image VARCHAR(255) NULL,
    gender VARCHAR(255) NULL,
    birthday VARCHAR(10) NULL,
    KEY `idx01_idx` (idx),
    KEY `idx02_sns_idx` (email)
)

```

데이터 저장 예시 1.

idx	id	pw	name_	email	nickname	phone	regDate
12345678	test	test123!	tester	seongjae.m@gmail.com	test	01012341234	2018-03-28 00:00:00
5	master	master123!	seongjaemoon	seongjae.m@gmail.com	moon	01012341234	2018-03-22 00:00:00

데이터 저장 예시 2.

idx	email	age	nickname	profile_image	gender	birthday
	clear219@naver.com	20-29	Moon	https://ssl.pstatic.net/static/pwe/address/img_pr...	M	02-21
	donghyun_zzang@hanmail.net	30-39	??	https://ssl.pstatic.net/static/pwe/address/img_pr...	M	12-24
	youmechoi1129@gmail.com	30-39	p****	https://ssl.pstatic.net/static/pwe/address/img_pr...	F	11-29
NULL	NULL	NULL	NULL	NULL	NULL	NULL

8. 네아로 테스트 적용 화면

테스트 로그인 화면.

회원관리 쌍용교육센터

로그인

정상적인 서비스 이용을 위해서 로그인이 필요합니다.

아이디:

아이디를 입력해주세요.

비밀번호:

비밀번호를 입력해주세요.

☐ 로그인 정보 기억하기

로그인

회원가입

로그인

[회원 정보를 잊어버리셨습니까?](#)

네아로 연동 로그인 화면.

NAVER

네이버 로그인으로 **SIST 회원관리 프로그램2** 서비스를 이용하실 수 있습니다.

clear219

.....

로그인

☒ 로그인 상태 유지

IP보안 **ON**

[일회용 로그인](#) ?

로그인 성공 및 프로필 화면.

쌍용교육센터 회원관리 v1.0

회원 관리


[Moon/clear219@naver.com/관리자로 로그인 중입니다.]로그아웃

방명록 글목록

번호	글쓴이	글내용	작성일	Client IP	Blind	
TotalCount 0	Count 0	BlindCount 0	Previous	Next	Name	Search

9. 네아로 실사용 사례

9-1. 서울시 모바일 플랫폼 사용 사례



로그인 회원가입 서울특별시

앱 공모전 주요기능 API 둘러보기 참여하기 활용하기 소통하기 마이페이지

HOME > 로그인





LOGIN

가입한 이메일 주소를 입력해주세요.

로그인

비밀번호를 입력해주세요

☐ 아이디저장 아이디 찾기 | 비밀번호 찾기

다른계정으로 로그인    

아직 서울시 모바일플랫폼 회원이 아니신가요?

회원가입

9-2. 번개장터 사용 사례



 네이버 아이디로 시작하기

 카카오톡 아이디로 시작하기

 페이스북 아이디로 시작하기

휴대폰번호 (숫자만 허용)

비밀번호 (8~16자의 영문,숫자,특수문자 조

로그인

[비밀번호 찾기](#) | [회원가입](#)

10. 네아로 참고 URL

네아로 프로필 조회 :<https://developers.naver.com/docs/login/profile/>

네아로 API 명세 <https://developers.naver.com/docs/login/api/>

네아로 Web 튜토리얼 <https://developers.naver.com/docs/login/web/>

이상으로 네아로 코드리뷰를 마치겠습니다.

감사합니다.