구글 로그인 과정

1. front 단에서 로그인 과정을 거치면 구글에서 설정해준 redirect\_url로 code값을 보낸다.

2. 받은 code, client\_id, client\_secret 값을 parameter로 보내고 구글에 등록한 redirect url로 accesstoken값을 얻는 과정을 수행한다.

-> accessToken값을 받아오면 로그인 성공

3. accessToken을 이용하면 user info를 받아 올 수 있다.

**<.View>**

|  |
| --- |
| 로그인을 수행하여 code값을 받아서 redirect\_url로 전송 |
| <div id="googleLoginBtn" style="cursor: pointer"> <p class="mt-5 login-form\_\_footer">google? <a href="#" class="text-primary">구글로 로그인</a> now</p></div>  <script>  const onClickGoogleLogin = (e) => {  //구글 인증 서버로 인증코드 발급 요청  window.location.replace("https://accounts.google.com/o/oauth2/v2/auth?client\_id=224677599374-p7ep3bont28baeu6l829pg50g74r96o1.apps.googleusercontent.com&redirect\_uri=http://localhost:8090/googleAuth&response\_type=code&scope=email%20profile%20openid&access\_type=offline")  }    const googleLoginBtn = document.getElementById("googleLoginBtn");  googleLoginBtn.addEventListener("click", onClickGoogleLogin);    </script> |

**<controller>**

|  |
| --- |
| - code를 받아서 accesstoken를 얻도록 service에 전달  - accesstoken값을 이용하여 사용자 정보를 얻기 위해 service에 전달 |
| @RequestMapping(value="/googleAuth", method=RequestMethod.GET)  public String googleAuth(HttpSession session,Model model, @RequestParam(value = "code") String authCode) throws JsonProcessingException{  System.out.println(authCode); // 로그인 과정으로 얻은 code값    // JsonNode트리형태로 토큰받아온다  JsonNode jsonToken = apiservice.getGoogleAccessToken(authCode);  // 여러 json객체 중 access\_token을 가져온다    accessToken = jsonToken.get("access\_token");  System.out.println("access\_token : " + accessToken);    session.setAttribute("google\_accessToken", accessToken.toString());    //accessToken을 이용하여 사용자 정보를 얻어온다.  JsonNode userInfo = apiservice.getGoogleUserInfo(accessToken.toString());  String google\_id = userInfo.path("id").asText();  String google\_name = userInfo.path("name").asText();  String google\_email = userInfo.path("email").asText();    System.out.println("id : " + google\_id);  System.out.println("name : " + google\_name);  System.out.println("email : " + google\_email);  } |

**<Service>**

|  |
| --- |
| - 로그인 과정으로 얻은 code값으로 rest-api를 호출하여 accessToken값을 얻는다. |
| public static JsonNode getGoogleAccessToken(String code) {    final String RequestUrl = "https://oauth2.googleapis.com/token"; // Host  final List<NameValuePair> postParams = new ArrayList<NameValuePair>();      postParams.add(new BasicNameValuePair("code", code)); // 로그인 과정중 얻은 code 값  postParams.add(new BasicNameValuePair("client\_id", "224677599374-p7ep3bont28baeu6l829pg50g74r96o1.apps.googleusercontent.com")); // REST API KEY  postParams.add(new BasicNameValuePair("client\_secret", "Ou2Hwlyp9kMXCxFVZJofiRQz"));  postParams.add(new BasicNameValuePair("redirect\_uri", "http://localhost:8090/googleAuth")); // 리다이렉트 URI  postParams.add(new BasicNameValuePair("grant\_type", "authorization\_code"));    final HttpClient client = HttpClientBuilder.create().build();  final HttpPost post = new HttpPost(RequestUrl);    JsonNode returnNode = null;    try {  post.setEntity(new UrlEncodedFormEntity(postParams));    final HttpResponse response = client.execute(post);  final int responseCode = response.getStatusLine().getStatusCode();    System.out.println("\nSending 'POST' request to URL : " + RequestUrl);  System.out.println("Post parameters : " + postParams);  System.out.println("Response Code : " + responseCode);    // JSON 형태 반환값 처리  ObjectMapper mapper = new ObjectMapper();    returnNode = mapper.readTree(response.getEntity().getContent());    } catch (UnsupportedEncodingException e) {  e.printStackTrace();  } catch (ClientProtocolException e) {  e.printStackTrace();  } catch (IOException e) {  e.printStackTrace();  } finally {  }    return returnNode;  } |

|  |
| --- |
| - accessToken값으로 rest-api를 호출하여 사용자 정보를 가져온다. |
| public JsonNode getGoogleUserInfo(String accessToken) {//String jsonMessage  JsonNode returnNode = null;  try{  OkHttpClient client = new OkHttpClient();  HttpUrl httpUrl = new HttpUrl.Builder()  .scheme("https")  .host("www.googleapis.com")  .addPathSegment("oauth2")  .addPathSegment("v2")  .addPathSegment("userinfo")  .addQueryParameter("access\_token", accessToken)  .build();    Request request = new Request.Builder()  .url(httpUrl)  .build();  //동기 처리시 execute함수 사용  okhttp3.Response response = client.newCall(request).execute();  ObjectMapper mapper = new ObjectMapper();  returnNode = mapper.readTree(response.body().string());  } catch (Exception e) {  System.err.println(e.toString());  }  return returnNode;  } |