

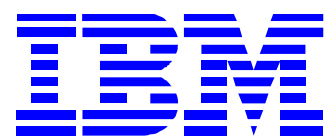
IBM API Connect v5.0

OAuth 2.0

(2016. 10.)

IBM Hybrid Cloud Technical Sales

이정운 차장(juwlee@kr.ibm.com)



0) OAuth 2.0 이란?

안녕하세요 이정운 입니다.

IBM API Connect 는 API 를 노출/관리/제어 하다보니 가장 중요한 부분중의 하나는 보안입니다. 특히, 보안중에서도 인증/인가 부분이 가장 밀접하게 해당 솔루션과 연관되어 있으며 다양한 사용자에게 단일 접점으로 노출되어 있기 때문에 인증/인가 되지 않은 사용자를 확인해서 쳐 내거나 인증/인가된 사용자면 Back-end 서버로 API 서비스를 가능하게 해야 합니다.

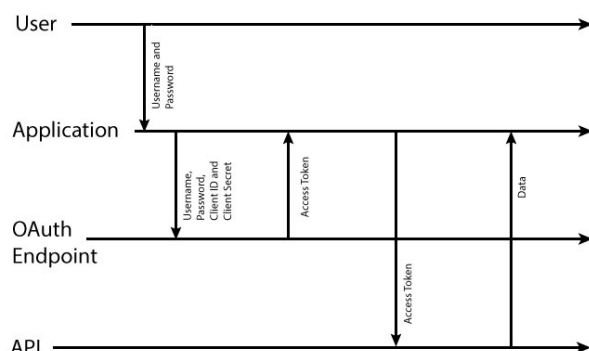
이때 가장 많이 사용하는 인증/인가에 대한 표준이 바로 OAuth 2.0 입니다. (지금까지 이해한 바로는 인증(Authentication) 보다는 인가(Authorization) 에 집중한 표준이며 인증을 하긴 하지만 결국은 요청된 scope 의 권한을 획득하도록 되어 있습니다.) OAuth 2.0 은 기존의 ID/Password 형태로 인증/인가를 수행하는 것이 아니라 접근 권한을 인증받기 위한 Token 을 발급 받아 이를 통해 인증/인가를 수행하는 방식입니다. 이미 Google, Facebook, Twitter 과 같이 거의 모든 대형 업체들이 해당 방식으로 API 사용에 대한 인증/인가를 수행하고 있기 때문에 이미 많이 사용되고 있는 IETF 글로벌 표준입니다.

OAuth 2.0 은 크게는 client ID 만 사용하는 public 방식과 client ID 와 client Secret 을 다 사용하는 confidential 방식으로 나누어지며 confidential 방식은 Application flow, Password flow, Access code flow 가 있으며 public 방식은 Implicit flow, Password flow, Access code flow 가 있습니다.

https://www.ibm.com/support/knowledgecenter/en/SSMNED_5.0.0/com.ibm.apic.toolkit.doc/tutorial_apionprem_security_OAuth.html

◦ Password flow

In the password flow scheme, the user provides the application with a user name and password that can be used to access the user's data. Following this, the client will directly contact the provider API to request an access token. In this case, trust must exist between user and application because the user's password is revealed to the application. However, this still has an advantage over the application using the password directly, because the validity of the access token or client ID can later be revoked without impacting other applications that do not need their access revoked. However, the application must be trusted to not store the user name and password.



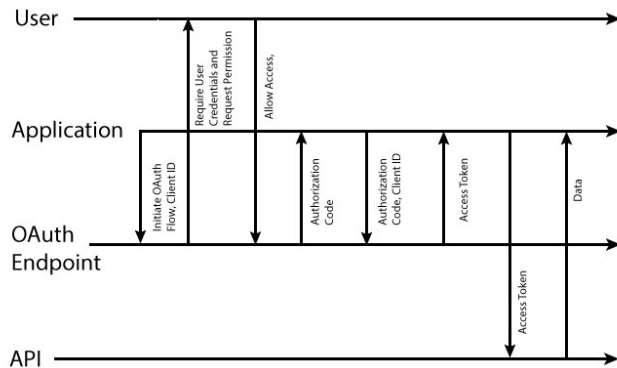
예를 들어서 위와 같은 Confidential 의 Password flow 는 Confidential 방법이기 때문에 Client ID 와 Client Secret 이 필요하며 2 legged 방식으로 사용자가 로그인 한 뒤에 애플리케이션이 OAuth 인증을 위해서 Username, Password, Client ID, Client Secret 을 OAuth Endpoint 에 던지면 사용자 동의 페이지 없이도 바로 Access Token 을 반환하게 됩니다.(내부적으로는 scope 도 같이 제공) 이후 해당 Access Token 을 가지고 Username/Password 없이 Access Token 만료시간(보통 1시간 정도) 전까지 별도의 인

증절차 없이 token 만으로 인증하여 API 호출을 수행하고 서비스가 가능합니다.

이외에도 기 언급한 것처럼 OAuth 2.0 은 다양한 flow 를 지원합니다.

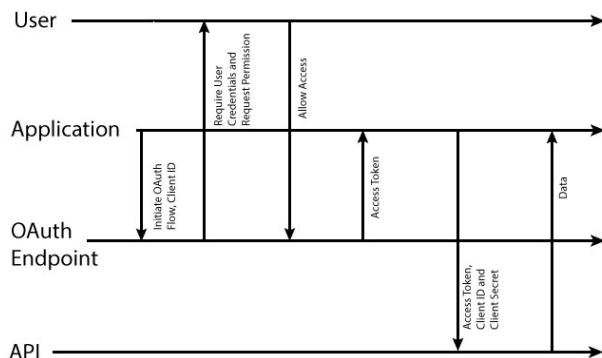
◦ Access code flow

In the access code flow, the application has the user provide authorization through a form provided by the gateway server, which, if they grant authorization, provides an authorization code to the user. The user then passes this authorization code to the application, and the application sends the authorization code to the provider API and is granted an access token in return.



◦ Implicit flow

In the implicit flow scheme, the application requests an access token from the gateway server and the user grants permission, at which point an access token is provided to the user, who must then pass the token to the application



OAuth 의 각 flow 마다 이러한 인증 흐름이 있으며 우선은 가장 심플한 Password flow 에 대한 것을 살펴보고 이후에 시간이 되면 다른 flow 도 한번 다뤄 보도록 하겠습니다.

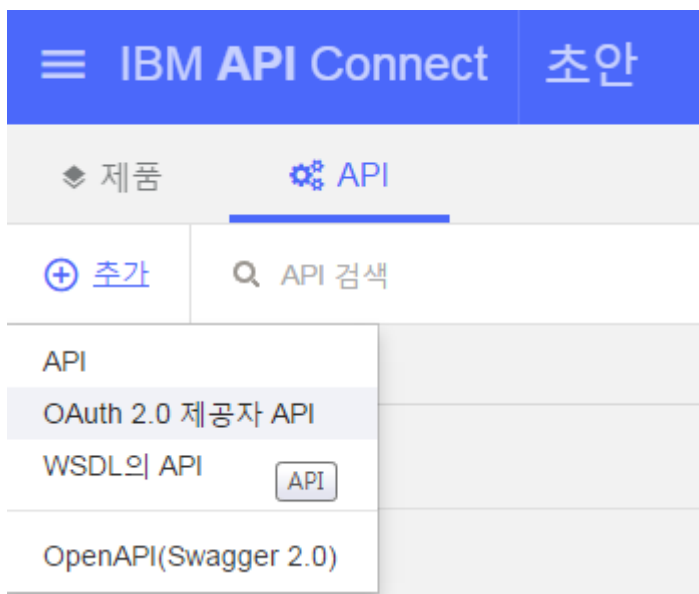
그럼 백문이 불여일타! OAuth 2.0 에 대한 강좌를 진행하도록 하겠습니다.

1) OAuth 2.0 설정 하기

Creating an OAuth 2.0 provider API

https://www.ibm.com/support/knowledgecenter/en/SSMNED_5.0.0/com.ibm.apic.toolkit.doc/tutorial_apionprem_security_OAuth.html

IBM API Connect 에서 API 서비스를 만든 후에 OAuth 2.0 사용을 위해서는 먼저 OAuth 2.0 제공자 API 를 만들어야 합니다. API Manager 콘솔에서 API > 추가를 선택하여 OAuth 2.0 제공자 API 를 선택합니다.



제목을 넣어주면 자동으로 동일하게 이름과 기본경로가 맞추어지며 Create API 버튼을 클릭하여 API 를 만듭니다.

OAuth 2.0 제공자 API

정보	제목 *
	oauth20
	이름 *
	oauth20
	기본 경로
	/oauth20
	버전 *
	1.0.0

취소
Add a product...
Create API

그러면 내부적으로 OAuth flow 에 필요한 경로나 정의들이 이미 들어가 있는 템플릿을 사용하여 OAuth 2.0 제공자 API 가 생성됩니다.

IBM API Connect

oauth20 1.0.0

← 모든 API

디자인

<> 소스

어셈블

정보

스킴

호스트

기본 경로

OAuth 2

이용

생성

라이프사이클

정책 어셈블리

보안 정의

보안

확장기능

특성

경로

/oauth2/authorize

/oauth2/token

매개 변수

정의

access_token_response

서비스

정보

제목 *

oauth20

이름 *

oauth20

버전 *

1.0.0

설명

연락처

이름

이메일

URL

이용 약관 및 라이선스

이용 약관

우선 기본으로 들어가있는 호스트 정보는 삭제합니다.

IBM API Connect

oauth20 1.0.0

← 모든 API

디자인

<> 소스

어셈블

정보

스킴

호스트

기본 경로

OAuth 2

이용

호스트

호스트 *

기본 경로

기본 경로

/oauth20

이제 본격적으로 OAuth 2 설정을 위해서 왼쪽 메뉴에서 OAuth 2 를 선택하면 다음과 같이 OAuth 2 설정을 위한 다양한 파라미터를 확인할 수 있습니다.

먼저 클라이언트 유형은 이전 파트에서 언급한 confidential 이나 public 을 정하는 것입니다. 이번 강좌에서는 confidential 형태의 password flow 를 설정 및 테스트 해볼 것이라 기밀을 선택합니다.

OAuth 2

클라이언트 유형

공용

범위

기밀

다음으로는 OAuth 2.0 제공자 API 가 권한을 줄 수 있는 scope 을 넣으면 됩니다. Scope 은 서비스 API 의 기본 경로(예:/ServiceAPI)를 의미하며 하나의 경로에 대해 scope 을 줄수 도 있고 하단과 같이 여러 scope 을 줄 수도 있습니다. (예를 들어 서비스 권한을 주고자 하는 API 의 기본 경로가 /authtest3 이라면 authtest3 를 입력하면 됩니다.)

범위

범위 이름	설명	
scope1	Description 1	
범위 이름	설명	
scope2	Description 2	
범위 이름	설명	
scope3	Description 3	

다음으로는 Grant 설정이며 상단의 파트에서 언급한 flow 를 선택하는 것으로 password flow 설정이므로 다른 것은 해제하고 비밀번호만 남겨둡니다.

부여

☒ 암시 ☒ 비밀번호 ☒ 애플리케이션 ☒ 액세스 코드

다음으로 ID 추출 방식을 선택할 수 있습니다. 테스트이기 때문에 기본을 선택합니다. (여기서 기본 방식은 브라우저의 기본 폼을 이용하는 방식입니다.)

ID 추출

인증

기본 양식

기본

사용자 정의 양식

경로 재지정

인증은 사용자 레지스트리(LDAP) 이나 인증 URL 을 선택할 수 있으며 인증 URL 방식은 ID/Password 를 가지고 지정된 URL 로 호출을 수행하여 해당 URL 이 200 정상 값을 반환하면 인증이 되었다고 판단하며 그 이외에 401, 403 등을 반환하면 인증에 실패했다고 판단하는 방식입니다. 해당 방식은 사용자 레지스트리 방식에 비해서 훨씬 유연하게 각 사마다 필요한 로직을 자유롭게 추가 할 수 있습니다.

인증

사용 중인 애플리케이션 사용자 인증 *

인증 URL

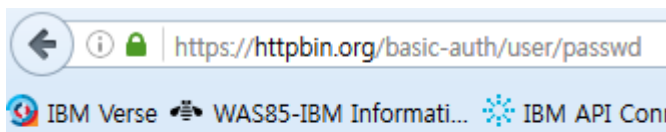
인증 URL

https://example.com/auth/url

TLS 프로파일

(인터넷이 되는 환경이라면 httpbin.org 에서 간단한 Basic Auth 서비스를 제공합니다. 하단과 같은 경로를 주면 username 을 user 로 주고 password 는 passwd 로 입력하면 하단과 같이 200 OK 와 함께 인증됨이라는 결과를 돌려주므로 데모용이라면 간단히 해당 서비스를 사용하셔도 됩니다.)

<https://httpbin.org/basic-auth/user/passwd>



```
{
  "authenticated": true,
  "user": "user"
}
```

권한은 인증이 되며 자동으로 권한이 있다고 판단하는 '인증됨' 이 있고 추가적으로 양식을 지정해서 받을 수 있습니다. 이번 강좌에서는 단순히 '인증됨'을 사용하도록 하겠습니다.

권한

기본 양식

사용자 정의 양식

인증됨

마지막으로는 실제 Token 에 대한 설정이며 Access token 에 대한 만료시간을 지정할 수 있습니다. (기본은 1시간) 또한, refresh token 을 사용할지 여부나 취소(revocation) 을 사용할지 여부도 지정할 수 있습니다. Refresh token 에 대해 좀 더 자세히 말씀드리면 해당 token 은 보다 긴 만료시간(약 한달)을 가지고 있으며 Access token 을 받기 위해 매번 OAuth flow 를 타는 것이 아니라 Access token 이 만료되었을 때 refresh token 만을 던져서 Access token 을 다시 받아 올 수 있습니다.

토큰	엑세스 토큰
	TTL(Time to Live)(초)
	3600
<input checked="" type="checkbox"/>	새로 고치기 토큰 사용
계수	TTL(Time to Live)(초)
2048	2682000
<input checked="" type="checkbox"/>	취소 사용
<input type="radio"/>	DataPower Gateway 사용
<input checked="" type="radio"/>	취소 URL 사용
	취소 URL
	TLS 프로파일
<input type="checkbox"/>	Enable token introspection
	Enabling this option inserts token introspect operation. It will also insert client ID and client secret header-based security definitions.

그리고 취소는 사용자별 각 Access token 을 개별적으로 제어하고 싶을때 사용하는 기능입니다. 만료와 상관없이 매 Access token 을 받을때 마다 확인/취소가 가능하여 개별적으로 Access token 을 제어 가능합니다. 이번 강좌에서는 사용하지 않기 때문에 해당 설정은 disable 합니다.

여기까지 하셨다면 기본 OAuth 2.0 설정은 완료하신 것입니다.

마지막으로는 좀 더 효율적인 호출을 위해 /authorize, /token 경로의 oauth2 를 빼고 단순하게 변경하고 저장을 수행합니다.(이렇게 하면 기본 경로인 /oauth20 에 각 경로가 붙는 형태, /oauth20/authorize 나 /oauth20/token 으로 호출이 가능합니다.)

<div>Info</div> <div>Schemes</div> <div>Host</div> <div>Base Path</div> <div>OAuth 2</div> <div>Consumes</div> <div>Produces</div> <div>Lifecycle</div> <div>Policy Assembly</div> <div>Security Definitions</div> <div>Security</div> <div>Extensions</div> <div>Properties</div> <div>Paths</div> <div>/authorize</div> <div>/token</div>	<div>Paths</div> <div>/authorize</div> <div>Path *</div> <div>/authorize</div> <div>Parameters</div> <div>No parameters defined</div> <div>GET /authorize</div> <div>POST /authorize</div> <div>/token</div> <div>Path *</div> <div>/token</div> <div>Parameters</div>
---	--

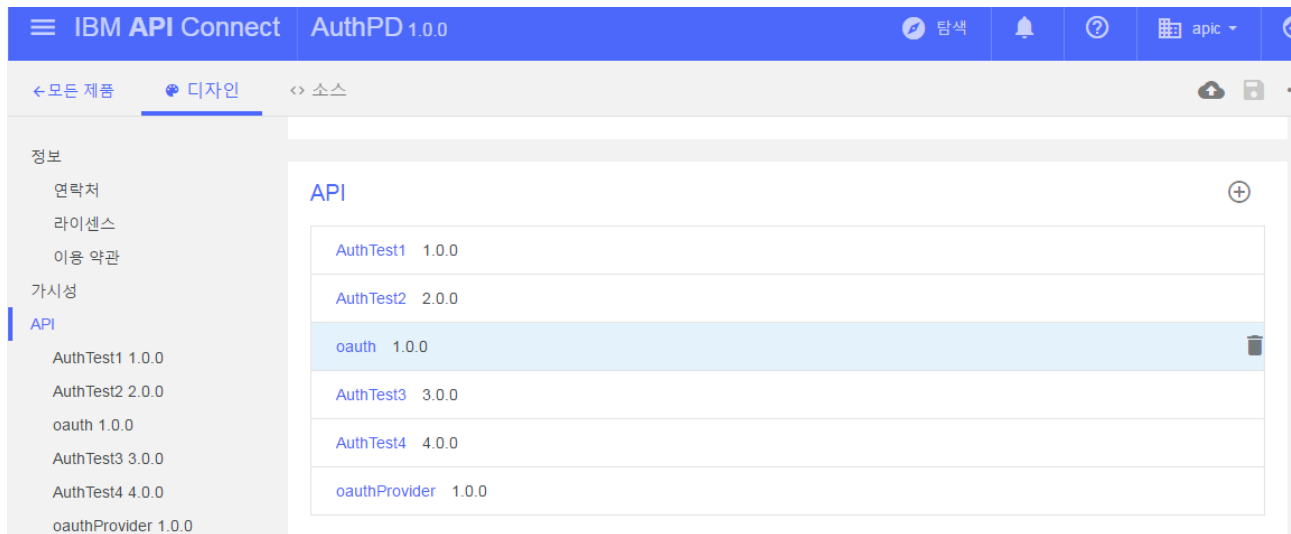
이렇게 OAuth 2.0 제공자 API 생성을 완료하였으면 이제 다음 단계는 서비스 API 의 보안 설정에 OAuth 설정을 추가하는 것 입니다.

보안 정의에서 OAuth 를 추가하고 이미 만들어둔 OAuth 2.0 제공자 API 에 맞추어서 사용하고자 하는 flow 나 토큰 URL, scope 등을 지정해주시면 됩니다.

보안 정의가 완료되면 보안에 사용을 위해서 해당 OAuth 보안 정의를 enable 해주시면 됩니다.

마지막으로 서비스 API 의 각 Operation 의 세부로 들어가서 보안 적용 설정이 하단과 같이 잘 되어있는 확인한 후 해당 서비스를 저장하면 됩니다.

이렇게 설정이 완료된 후에는 서비스 하는 제품에 OAuth 2.0 제공자 API 도 포함한 후 배포 해주시면 됩니다.



2) OAuth 2.0 테스트 하기

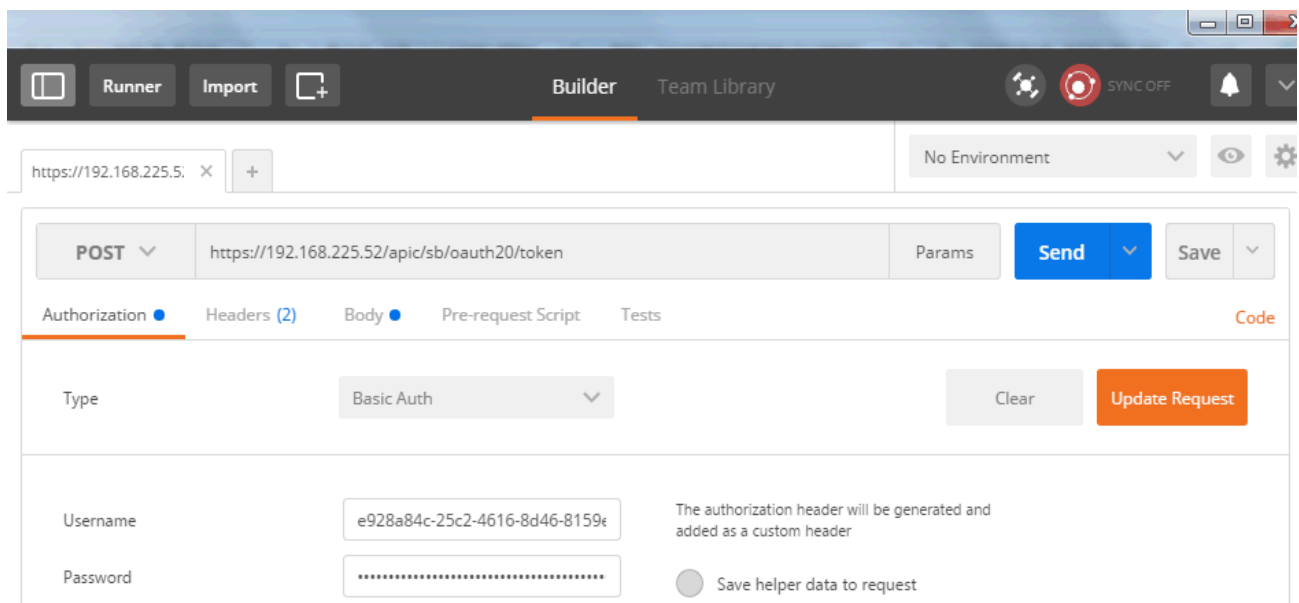
OAuth 2.0 을 테스트하기 위해서는 샘플 애플리케이션을 사용하시거나 간단하게 온라인에서 돌아다니는 REST client 를 가지고 테스트가 가능합니다. 이번 강좌에서는 개발자들이 많이 사용하는 chrome 의 postman 을 가지고 테스트를 수행해보도록 하겠습니다.

가장 먼저 API Portal 에 접속하여 하단과 같이 client ID/client Secret 을 얻기 위하여 애플리케이션을 하나 생성합니다. 당연히 해당 애플리케이션은 서비스 API 에 대한 구독신청도 완료되어야 합니다.(이전 강좌 참고)

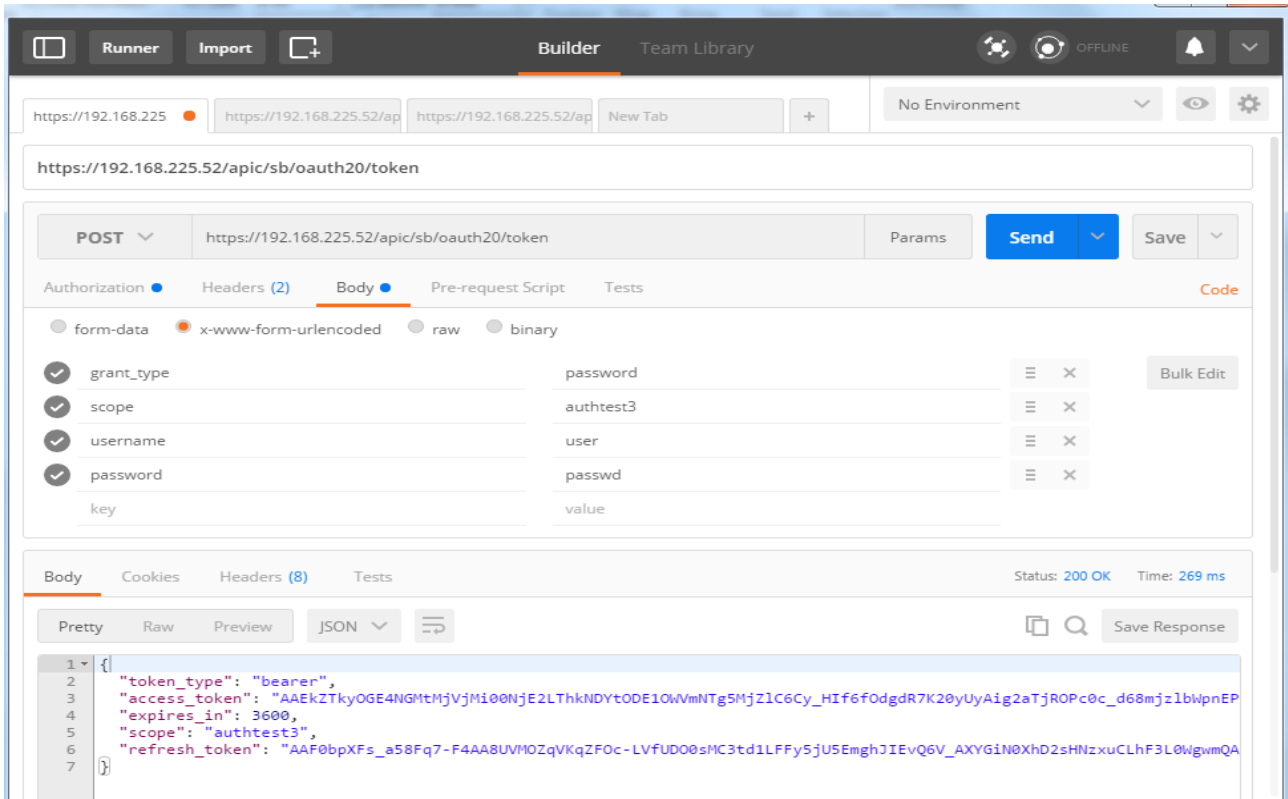
< 애플리케이션으로 돌아가기



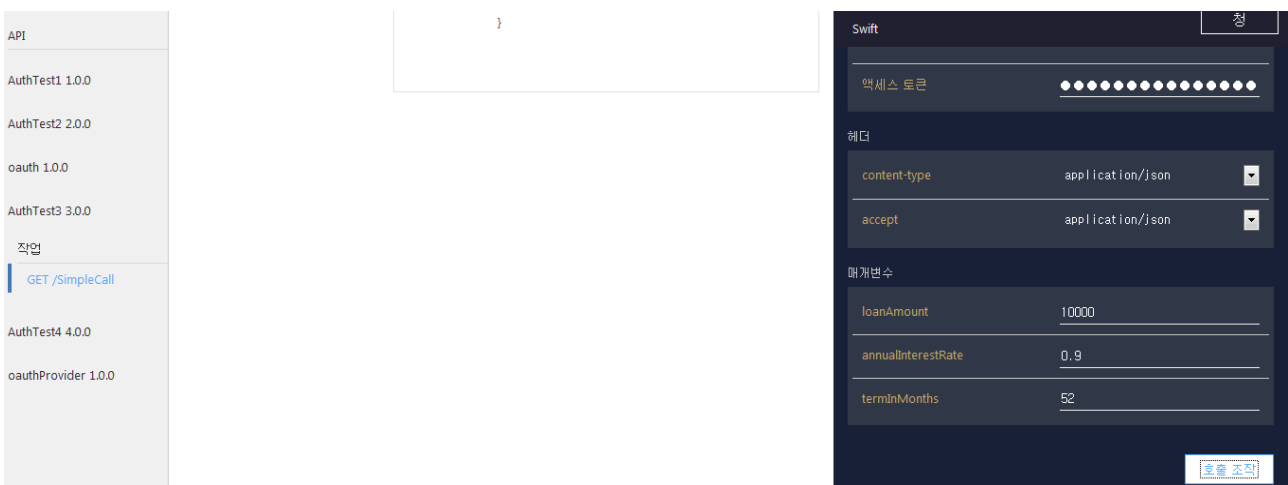
다음으로 POST 형태의 호출을 위해서 OAuth 2.0 제공자 API 의 token URL 을 입력하고(URL 을 잘 모르시면 API Portal 에서 확인이 가능합니다.) Authorization 항목은 Basic Auth 로 하고 username/password 에는 테스트하고자 하는 App을 위해 받아 둔 client ID/client Secret 을 입력합니다.



Access token 을 받아오기 위한 추가 파라미터(grant_type, scope, username, password)를 넣으신 후에 Send 를 누르면 Access Token 을 하단과 같이 가지고 옵니다. (보시면 잘 아시겠지만 refresh token 도 같이 받아옵니다.)



이렇게 Access token 을 얻게 되면 해당 토큰의 만료 전까지는 애플리케이션이나 API Portal 에서 Access token 을 넣고 하단과 같이 서비스 API 를 정상적으로 호출 테스트가 가능하며 그 결과를 받아 올 수 있습니다.



여기까지 잘 따라오셨다면 OAuth 2.0 을 맛보기 위한 Confidential 방식의 Password flow 형태의 OAuth 2.0 설정 및 테스트를 잘 완료하신 것입니다.

9) 참고 자료

1. IBM API Connect 5.0 온라인 메뉴얼(Knowledge Center) – 한글 제공
https://www.ibm.com/support/knowledgecenter/ko/SSMNED_5.0.0/mapfiles/ic_home.html
2. Securing an API by using OAuth 2.0 - IBM API Connect 5.0 온라인 메뉴얼
https://www.ibm.com/support/knowledgecenter/en/SSMNED_5.0.0/com.ibm.apic.toolkit.doc/tutorial_apionprem_security_OAuth.html