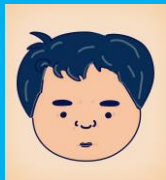


1.

오픈뱅킹 API

활용



<고 강 태>

gangtai.goh@gmail.com

목차

1. OpenBanking API
2. NH 오픈뱅킹 개발자센터

1. OpenBanking API

- 01. OpenBanking 시작
- 02. OpenBanking API
- 03. REST API 이해
- 04. 데이터 전송 형식
- 05. REST API Authentication
- 06. POSTMAN

01. OpenBanking 시작

■ 2019/12/17 "은행과 핀테크 기업 모두가 참여하는 오픈뱅킹 서비스 전면 시행"

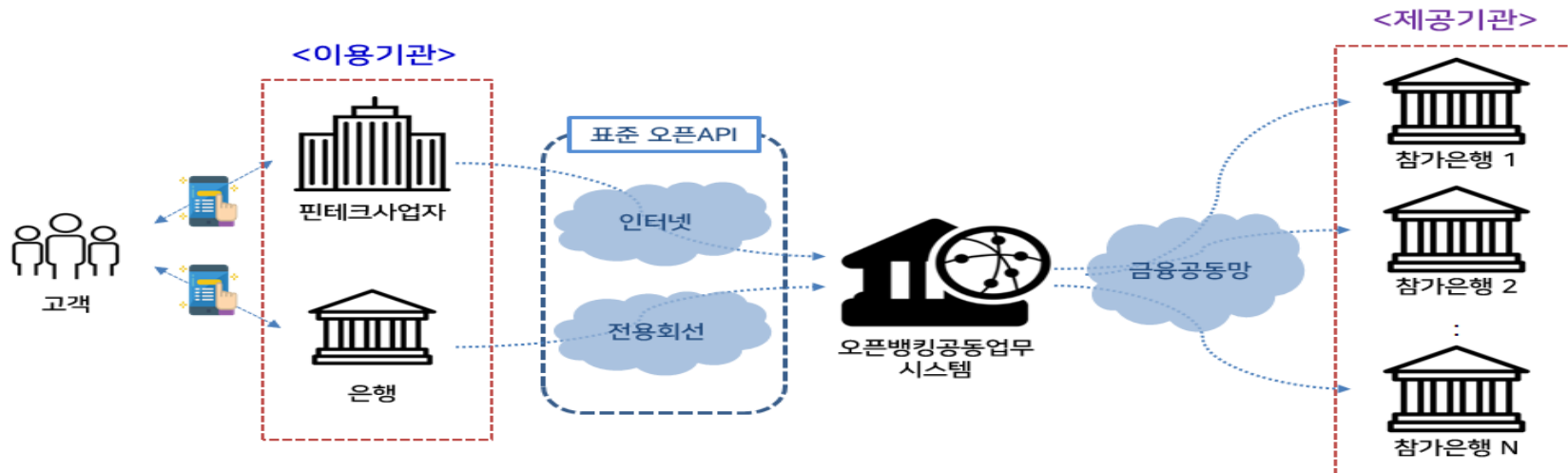
■ 제공 서비스

구분	세부 내용
조회	① 잔액조회: 사용자 본인계좌의 잔액 조회
	② 거래내역조회: 사용자 본인계좌 입출금내역 조회
	③ 계좌실명조회: 이용기관이 사용자 계좌의 유효성 및 예금주명 조회
	④ 송금인정보조회: 이용기관 계좌로 입금한 사용자명 및 송금계좌번호 조회
이체	⑤ 출금이체: 사용자 계좌에서 자금을 인출하여 이용기관 계좌로 입금
	⑥ 입금이체: 이용기관 지급계좌에서 자금을 인출하여 사용자 계좌로 입금

■ 해외 오픈뱅킹 사례

국가명	오픈뱅킹 관련 정책
EU	<ul style="list-style-type: none"> ○ PSD2(Payment Services Directive 2) 도입('18.1월) ✓ 은행 API를 핀테크 기업에 수수료 등 차별 없이 제공토록 의무화 ○ GDPR(General Data Protection Regulation)에서는 '개인정보이동권'을 도입하여 고객의 정보 자기결정권 강화('18.5월)
영국	<ul style="list-style-type: none"> ○ 9대 주요 은행 대상 오픈 API 서비스 실시('18.1월) ✓ 은행들이 오픈 API를 통해 타은행의 고객 정보를 받아, 타은행 계좌 접근 등 다양한 서비스 실시 → 은행의 결제기능 강화 및 경쟁 확대 ○ 英OBIE(Open Banking Implementation Entity)는 기술사양, 보안 등 API 표준요건을 담은 Open Banking Standard 3.0 발표('18.9월) ※ 英 오픈뱅킹 API 이용건수는 '19.10월 1.8억건으로 전년 동월(1,390만건)대비 약 13배 성장
호주	<ul style="list-style-type: none"> ○ 재무부에서 오픈뱅킹 구현을 위한 권고안 발표('18.2월) ✓ (4대 주요은행) '19.7월 신용·직불카드 및 예금·거래계좌부터 시작하여 '20.2월까지 주택담보대출 등으로 확대 예정 ✓ (기타 은행) 12개월 시차를 두고 점진적으로 확대, '22.7월 전 은행권의 전 금융상품에 대한 API 공개 예정
일본	<ul style="list-style-type: none"> ○ 은행법 개정을 통해 핀테크 기업에 API 제공 등 의무화('18.6월) ✓ 핀테크 기업에 대한 API 제공 기준 공시 및 기준 충족업체에 대해 API 제공 의무화 (법 시행 후 2년내 은행 API 구축 노력 명시) ↳ 日당국, '20년까지 110개 은행이 API 공개 완료 예상
싱가포르	<ul style="list-style-type: none"> ○ 아태지역 최초로 오픈뱅킹 지침 발표('16년) ✓ 정부 주도의 금융데이터 개방 유도 정책* 추진 중 * 은행(Citi, OCBC 등), 지급결제업체(NETS) 등이 거래내역 등 약 310여종 API 개방 중이며, MAS도 감독 관련 API를 개방하여 레그테크 부문 발전 도모
홍콩	<ul style="list-style-type: none"> ○ 홍콩금융관리국에서 'Open API Framework'를 마련('18.7월) ✓ 단계별* 오픈뱅킹 추진 중 * 1단계(은행상품 및 서비스 정보), 2단계(고객의 취득 및 신규 신청), 3단계(계좌정보), 4단계(거래처리)로 구분
미국	<ul style="list-style-type: none"> ○ 별도 규정 없이 시장 자율에 따라 금융회사별로 오픈뱅킹 추진

■ 오픈뱅킹 구조



○ (참여기관) 은행 및 대형 핀테크 업체도 참여 허용

■ 오픈뱅킹 수수료

업 무 \ 구 분		현행 비용	기본비용 (대형)	경감비용 (중소형*)
조회	잔액조회	10원	10원	5원
	거래내역조회	50원	30원	20원
	계좌실명조회	100원	50원	30원
	송금인정보조회	300원	50원	30원
이체	출금이체	500원	50원	30원
	입금이체	400원	40원	20원

○ (수수료) 기존 대비 1/10 수준(중소형은 1/20)으로 조정









■ 사용자 오픈뱅킹

- 개별 은행과 제휴 없이도 오픈뱅킹 공동업무 시스템 접속으로 전체 참가은행(18개)이 연결될 수 있는 공동형 플랫폼



■ 은행 오픈뱅킹

○

① 자금집금 서비스	② 오픈뱅킹 특화상품
 <p>여러 은행 계좌를 한번에 이체하여 자금모으기</p>	 <p>오픈뱅킹 특화 예금상품 출시 (오픈뱅킹 통한 상품 가입 우대 금리 제공)</p>
③ 통합자산관리	④ 손쉬운 환전
 <p>오픈뱅킹 등록 타행계좌를 결합한 자산관리서비스 및 맞춤형 상품추천</p>	 <p>다수 은행 계좌에서 자금을 집금하여 환전·외화 송금</p>
⑤ 간편결제 충전	⑥ 더치페이
 <p>모바일 간편결제(페이서비스) 이용시 여러 은행 계좌에서 손쉬운 충전·결제</p>	 <p>여러 타행계좌를 활용하여 더치페이</p>
⑦ 결제대금 선결제	⑧ 지능형 납부기일관리
 <p>오픈뱅킹 타행출금 기능을 이용한 결제대금 선결제 가능 제공</p>	 <p>대출이자 납부일에 당행계좌 잔액 부족시 타행계좌 출금으로 연체 방지</p>

■ 핀테크 기업 오픈뱅킹

- 모든 은행의 송금 서비스를 제공한 경우
 - 핀테크 기업이 부담하는 수수료 비용이 1/10 수준으로 절감되어 무료송금 건수 확대 등 소비자혜택이 크게 확대될 전망
 - 예: 토스, 카카오페이
- 일부 은행과의 제휴만 이루어졌던 경우
 - 서비스가 제공되지 않았던 은행 계좌에 대해서도 연결이 가능해져 비제휴은행 계좌가 있었던 소비자도 서비스 이용가능
 - 예: 핀크,뱅크샐러드

02. OpenBanking API



<https://www.openbanking.or.kr/main>

■ 금융결제원 오픈API 개발자 사이트

- <https://developers.kftc.or.kr/dev/starter/starter>

오픈뱅킹 API

잔액조회

사용자가 이용기관이 제공하는 서비스를 통해 본인 계좌에 대한 잔액 및 출금 가능 금액을 조회하는 기능을 제공합니다.



거래내역조회

사용자가 이용기관이 제공하는 서비스를 통해 본인 계좌에 대한 잔액 및 거래내역을 조회하는 기능을 제공합니다.



계좌실명조회

이용기관이 자금을 수취할 수취인 또는 출금이체 신청을 한 사용자 계좌의 정상여부 및 실명을 실시간 조회하는 기능을 제공합니다.



송금인 정보조회

소액해외송금업자의 송금인 신분 확인을 위해 특정기간동안 입금된 송금인 성명과 계좌정보를 조회하는 기능을 제공합니다.



입금이체

이용기관의 지급 계좌에서 자금을 인출하여 수취인 계좌로 실시간 입금하는 기능입니다.



출금이체

이용기관의 출금에 동의한 사용자 계좌에서 출금하여 이용기관의 수납 계좌로 실시간 입금하는 기능입니다.

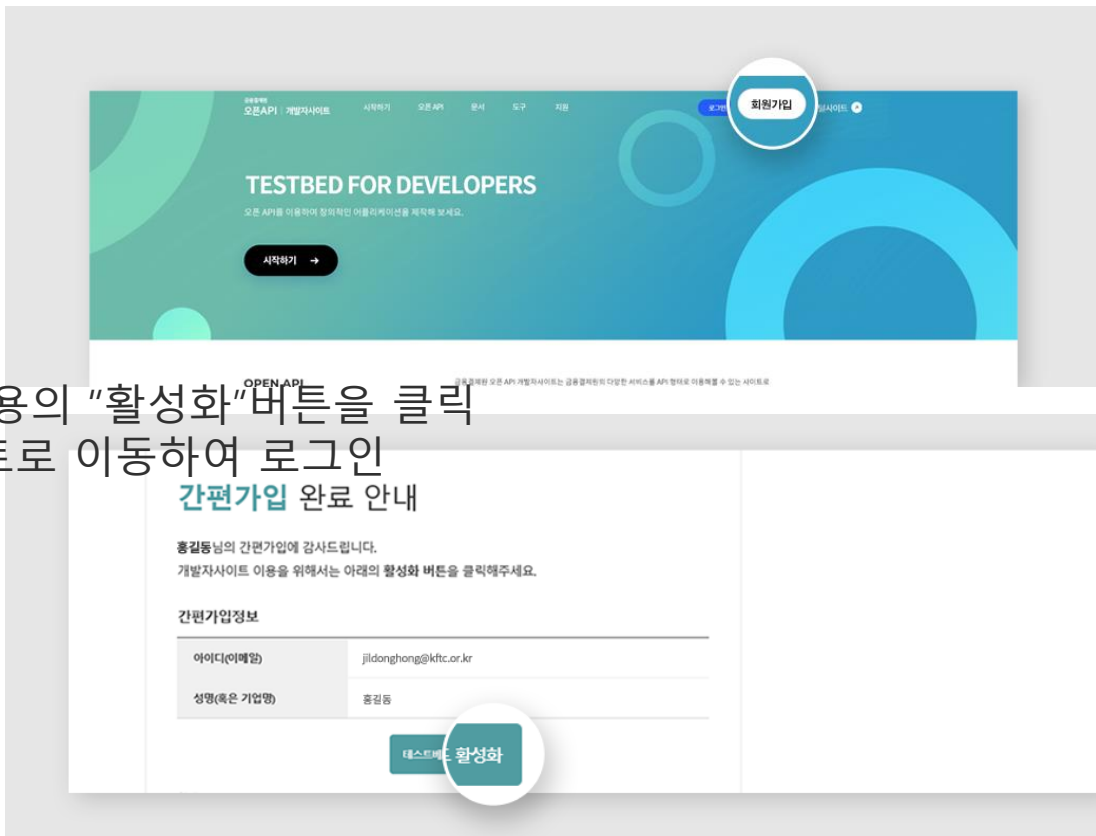


수취조회

이용기관이 입금이체 요청 전 수취계좌의 입금가능여부 및 수취인 성명을 사전에 조회하는 기능을 제공합니다.



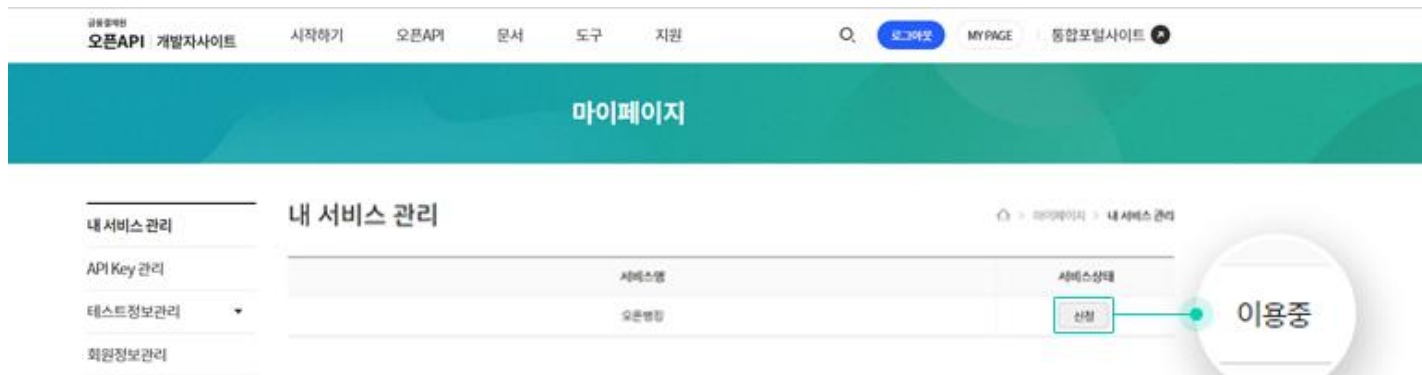
■ 금융결제원 오픈API 개발자사이트 가입



■ 금융결제원 오픈API 개발자 서비스

○ 서비스 신청

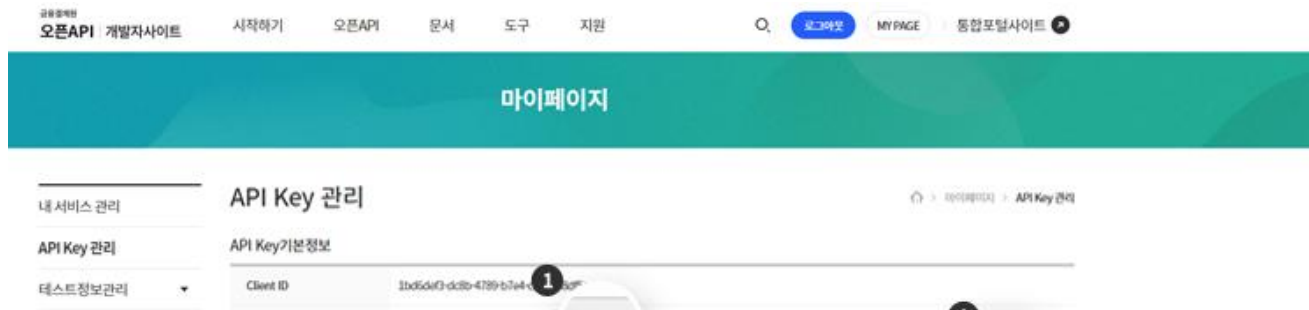
- "마이페이지-내 서비스 관리" 메뉴를 통해 금융결제원에서 제공하는 서비스를 확인하고 이용 희망 서비스를 신청



■ 금융결제원 오픈API 개발자 서비스

○ API Key

- "마이페이지 - API Key 관리" 메뉴에서 API Key를 발급
- 클라이언트 아이디(ID: client_id), 비밀번호(Secret: client_secret)
- Callback URL



■ 금융결제원 오픈API 개발자 서비스

○ API Key

– Callback URL

✓ OAuth 기반 사용자 인증 시 각 서비스의 서버와 정보를 주고받을 때 사용되는 정보로 API 테스트를 진행하기 전에 필수로 설정되어야 합니다.

Callback URL은 최대 10개까지 등록할 수 있습니다.

금융결제원
오픈API 개발자사이트

시작하기 오픈API 문서 도구 지원

로그아웃 MY PAGE 통합포털사이트

마이페이지

내 서비스 관리

API Key 관리

테스트정보관리

회원정보관리

API Key 관리

API Key 기본정보

Client ID	1bd5dcf3-dc8b-4789-b7e4-...
Client Secret	***** 조회
Callback URL	http://localhost:8880/html/callback

저장

■ 금융결제원 오픈API 개발자 서비스

○ 서비스에 API Key 등록

- “내 서비스 관리” 메뉴에서 신청한 서비스에 API Key를 각각 등록

내 서비스 관리

API Key 관리

테스트정보관리

회원정보관리

API Key 관리

API Key 기본정보

Client ID

1bd5def3-dc8b-4789-b7e4-c2045a8d5ee

Client Secret

.....

조회

Callback URL

http://localhost:8880/html/callback.html

+

http://localhost:8880/html/callback2.html

-

저장

3

이용중인 서비스

서비스명	API명	API Key 등록	Calb
오픈뱅킹	잔액조회, 거래내역조회, 계좌잔액조회, 송금인정보조회, 수취조회, 입금이체, 출금이체	등록	완료

■ 금융결제원 오픈API 개발자 서비스

○ 테스트 정보 관리

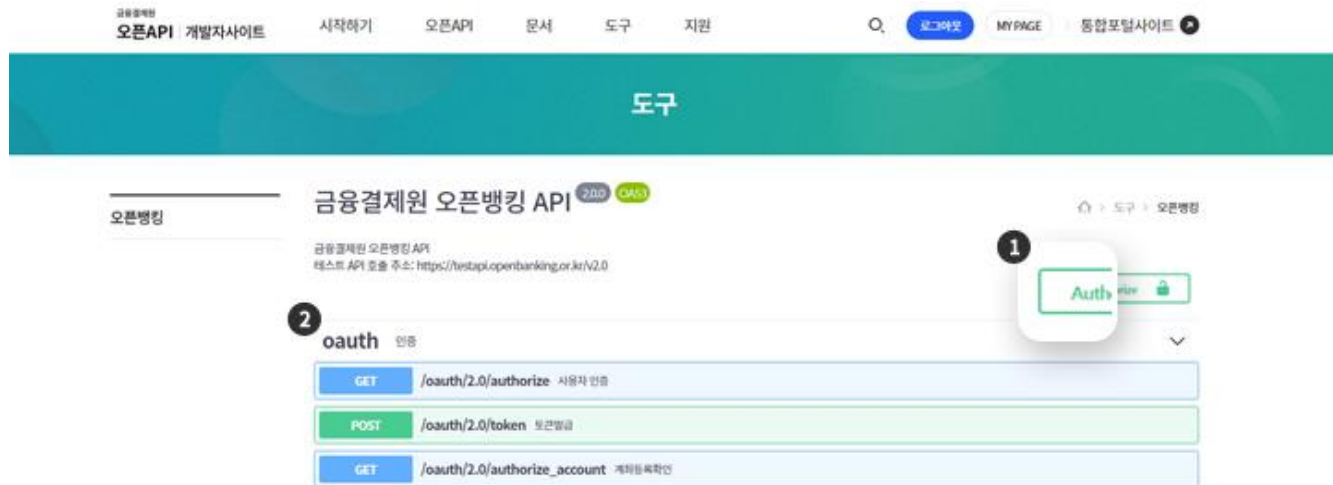
- 오픈API를 테스트하기 전에 개발자가 사전에 API 요청에 대한 응답 데이터를 등록하고 등록된 데이터를 활용하여 테스트를 수행할 수 있다.

The screenshot shows the '마이페이지' (My Page) interface of the OpenAPI Developer Service. The main section is '응답정보 관리' (Response Information Management). On the left, there is a sidebar with navigation links: '내 서비스 관리' (My Service Management), 'API Key 관리' (API Key Management), '테스트정보관리' (Test Information Management), and '회원정보관리' (Member Information Management). Under '테스트정보관리', there is a sub-link '응답정보관리' (Response Information Management). The main content area shows a dropdown menu for 'API명' (API Name) with '송금이체' (Remittance) selected. Below this is a table with columns: '출금기관 대표코드' (Debit Institution Representative Code), '출금 계좌번호' (Debit Account Number), '출금 계좌번호 (출입용)' (Debit Account Number (In/Out)), '출금인 설명' (Debit Description), '거래금액' (Transaction Amount), '입금계좌 인자내역' (Credit Account Detail), '수취인성명' (Payee Name), and '삭제' (Delete). A button '데이터추가' (Add Data) is located above the table. A note at the bottom states: '사용자당연번호가 없으시면 사용자인증 API를 통하여 신규사용자 등록 및 계좌등록을 먼저 진행하십시오.' (If you do not have a user ID number, please register a new user and account through the user authentication API).

■ 금융결제원 오픈API 개발자 서비스

○ 오픈 API 테스트

- "도구" 메뉴에서 제공하는 REST API 테스트 기능을 활용하여 테스트를 진행
 - ✓ 토큰발급: 접근토큰(Access Token) 발급 필요
 - ✓ REST API 테스트 도구 활용

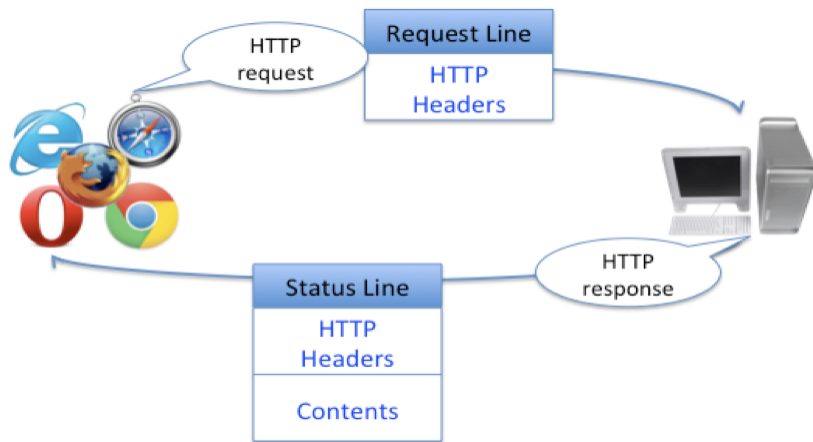


03. REST API 이해

3 HTTP protocol

■ HTTP Hyper Text Transfer Protocol 은 TCP/IP 를 기반으로 네트워크 통신을 제공하는 프로토콜

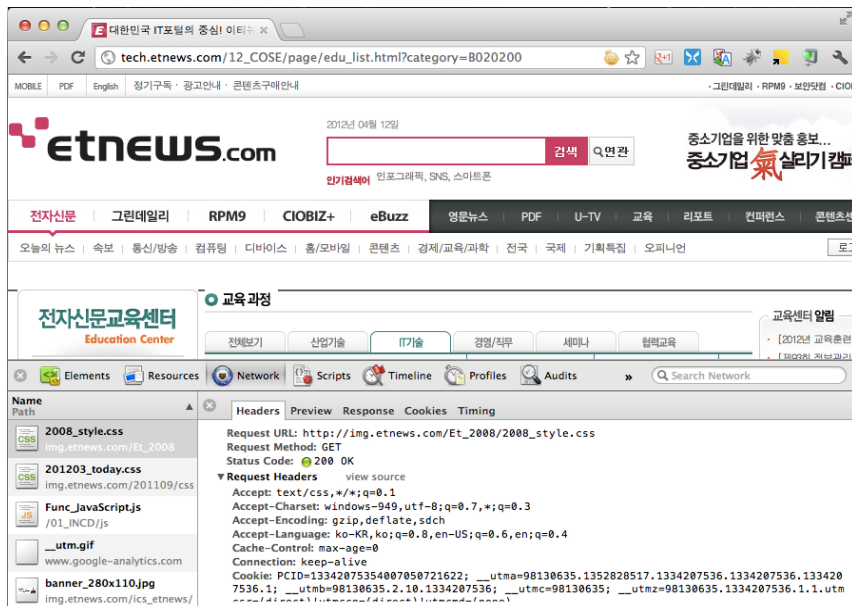
- HTTP는 RESTFull 기반으로 인터넷 상의 리소스 자원을 요청하면 해당 하는 콘텐츠 자원을 담아 응답하고 연결을 종료하는 프로토콜.



<http://nodejs.org/api/http.html>

■ 크롬브라우저의 개발자도구

- 웹 브라우저에서 서버에 요청한 결과의 네트워크 헤더를 살펴볼 수 있다.
- Network 탭을 선택해서 요청한 주소를 클릭하면 아래 화면같이 요청 헤더, 미리보기, 응답, 쿠키 및 시간 소요 그래프 이용.



■ Web Service API

- Web API 를 지향 (네트워크)
- 대표적 SOAP 과 REST
-

Web API 구현

#	SOAP	REST
1	A XML-based message protocol	An architectural style protocol
2	Uses WSDL for communication between consumer and provider	Uses XML or JSON to send and receive data
3	Invokes services by calling RPC method	Simply calls services via URL path
4	Does not return human readable result	Result is readable which is just plain XML or JSON
5	Transfer is over HTTP. Also uses other protocols such as SMTP, FTP, etc.	Transfer is over HTTP only
6	JavaScript can call SOAP, but it is difficult to implement	Easy to call from JavaScript
7	Performance is not great compared to REST	Performance is much better compared to SOAP - less CPU intensive, leaner code etc.

■ RESEFull API는 기술적으로 다음 같이 발전해 왔다.

Web API 구현

○HTTP/1.0 1994-1996

- Roy T. Fielding, [HTTP Object Model](#)
- "How do i improve HTTP without breaking the Web?"

○REST 1998

- Roy T. Fielding, Microsoft Research [발표](#)
- "Representational State Transfer:
An Architectural Style for Distributed Hypermedia Interaction"

○REST 2000

- Roy T. Fielding, 박사 논문으로 [발표](#)
- "Architectural Styles and the Design of Network-based Software Architectures"

○참고: <http://slides.com/eungjun/rest#/9>

■ Microsoft에서도 XML-RPC 1998를 통해 Web API를 구현하고자 했다.

○Salesforce API

○Salesforce API, 2000 은 이런 방식을 처음 도입했다.

– <https://web.archive.org/web/20001018092215/http://salesforce.com/>

○flickr API

– flickr의 SOAP

– <https://web.archive.org/web/20040921173223/http://www.flickr.com:80/services/api/response.soap.html>

○flickr의 REST API

– <https://web.archive.org/web/20040821031039/http://www.flickr.com:80/services/api/response.rest.html>

■ Oracle REST Architecture Overview 자료 참고

- 공유 폴더: REST_Overview.pptx

■ MS의 REST API 가이드 라인

- <https://github.com/Microsoft/api-guidelines>

04. 데이터 전송 형식

■ CSV 형식

○CSV(Comma Separated Value)는 각 항목을 쉼표로 구분해 데이터를 표현하는 방법

- CSV 형식은 데이터의 가독성이 떨어짐
- CSV 형식의 문자열은 다음과 같이 split () 메서드로 분해

	A	B	C	D	E	F	G	H	I	
1	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl
2	audi	a4	1.8	1999	4	auto(l5)	f		18	29 p
3	audi	a4	1.8	1999	4	manual(m5)	f		21	29 p
4	audi	a4	2	2008	4	manual(m6)	f		20	31 p
5	audi	a4	2	2008	4	auto(av)	f		21	30 p
6	audi	a4	2.8	1999	6	auto(l5)	f		16	26 p
7	audi	a4	2.8	1999	6	manual(m5)	f		18	26 p
8	audi	a4	3.1	2008	6	auto(av)	f		18	27 p
9	audi	a4 quattr	1.8	1999	4	manual(m5)		4	18	26 p
10	audi	a4 quattr	1.8	1999	4	auto(l5)		4	16	25 p

```

1  manufacturer,model,displ,year,cyl,trans,drv,cty,hwy,fl,class
2  audi,a4,1.8,1999,4,auto(l5),f,18,29,p,compact
3  audi,a4,1.8,1999,4>manual(m5),f,21,29,p,compact
4  audi,a4,2,2008,4>manual(m6),f,20,31,p,compact
5  audi,a4,2,2008,4,auto(av),f,21,30,p,compact
6  audi,a4,2.8,1999,6,auto(l5),f,16,26,p,compact
7  audi,a4,2.8,1999,6>manual(m5),f,18,26,p,compact
8  audi,a4,3.1,2008,6,auto(av),f,18,27,p,compact

```

■ XML 형식

- eXtentable Markup Language
 - HTML 의 모태가 된 정의 언어
 - <https://ko.wikipedia.org/wiki/XML>

```
<?xml version="1.0"?>
<quiz>
  <qanda seq="1">
    <question>
      Who was the forty-second
      president of the U.S.A.?
    </question>
    <answer>
      William Jefferson Clinton
    </answer>
  </qanda>
  <!-- Note: We need to add
  more questions later.-->
</quiz>
```

XML

■ JSON 형식

○JSON은 자바스크립트에서 사용하는 객체 형태로 데이터를 표현하는 방법

- Javascript Object Notation
- Key : Value 형식의 사전형식
- CSV 형식과 XML 형식의 단점을 장점을 포함

```
1  {  
2    "이름": "홍길동",  
3    "나이": 25,  
4    "성별": "여",  
5    "주소": "서울특별시 양천구 목동",  
6    "특기": ["농구", "도술"],  
7    "가족관계": {"#": 2, "아버지": "홍판서", "어머니": "춘섬"},  
8    "회사": "경기 수원시 팔달구 우만동"  
9  }
```

■ JSON 확장

○ <https://ko.wikipedia.org/wiki/JSON>

- JSON 스트리밍
- 관련 포맷
 - GeoJSON
 - JSON-LD
 - JSON-RPC
 - JsonML
 - S-표현식
 - SOAPjr
 - GBSON - annotation format of nucleic acid sequences (DNA, RNA)^[12]
- 관련 바이너리 인코딩
 - BSON
 - CBOR
 - MessagePack
 - 스마일 - JSON 기반
 - UBJSON
 - EXI4JSON (EXI for JSON)
- 구현체:
 - 잭슨

05. REST API Authentication

■ 거의 모든 REST API에는 일종의 인증이 있어야 합니다.

○가장 일반적인 헤더 중 하나는 권한 부여 호출입니다.

■ Authentication vs. Authorization

○인증과 권한 부여의 구분은 RESTful API가 작동하는 방식과 연결 시도가 수락되거나 거부되는 이유를 이해하는 데 중요합니다.

■ Authentication

○연결 시도의 자격 증명을 확인하는 것. 이 프로세스는 인증 프로토콜을 사용하여 일반 텍스트 또는 암호화된 형식으로 원격 액세스 클라이언트에서 원격 액세스 서버로 자격 증명을 보내는 것으로 구성

■ Authorization

○연결 시도가 허용되는지 확인하는 것입니다. 인증 성공 후 권한 부여가 발생합니다.

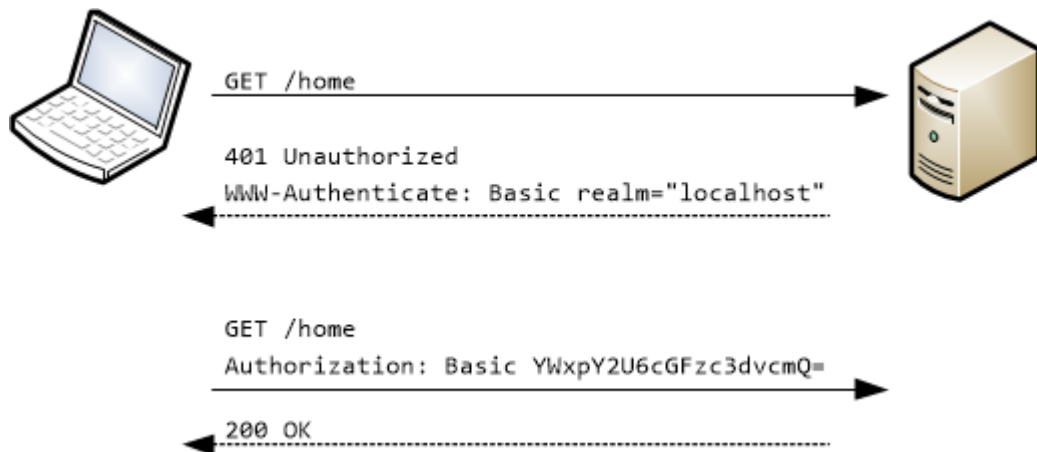
■ Basic Authenticon

- base64로 인코딩된 'username:password'를 추가하는 특수 HTTP 헤더를 사용
-

```
GET / HTTP/1.1  
Host: example.org  
Authorization: Basic Zm9vOmJhcg==
```

■ Basic Authenticon

- 자격 증명이 인코딩된 경우에도 암호화되지 않습니다! 기본 인증에서 사용자 이름과 비밀번호를 검색하는 것은 매우 쉽습니다.



■ HMAC (hash based message authentication)

○ 보내야 하는 암호 대신 더 많은 정보와 함께 암호의 해시된 버전을 보낸다

– 접근 리소스: /users/username/account

1) hmac을 생성

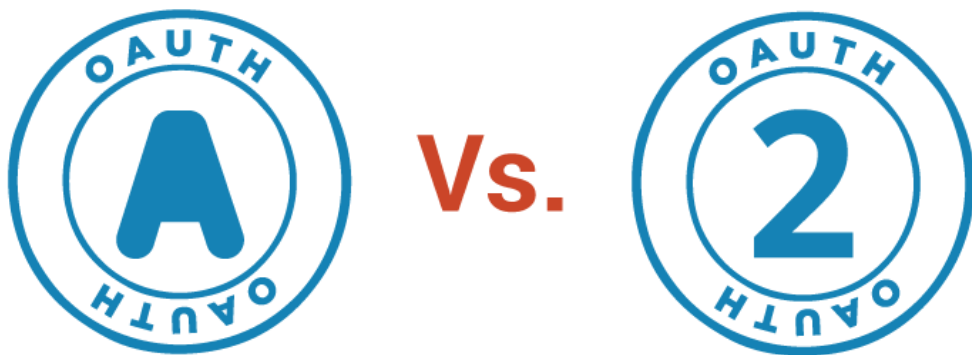
```
digest = base64encode(hmac("sha256", "secret",  
"GET+/users/username/account"))
```

2) 다이제스트를 HTTP 헤더로 보낸다

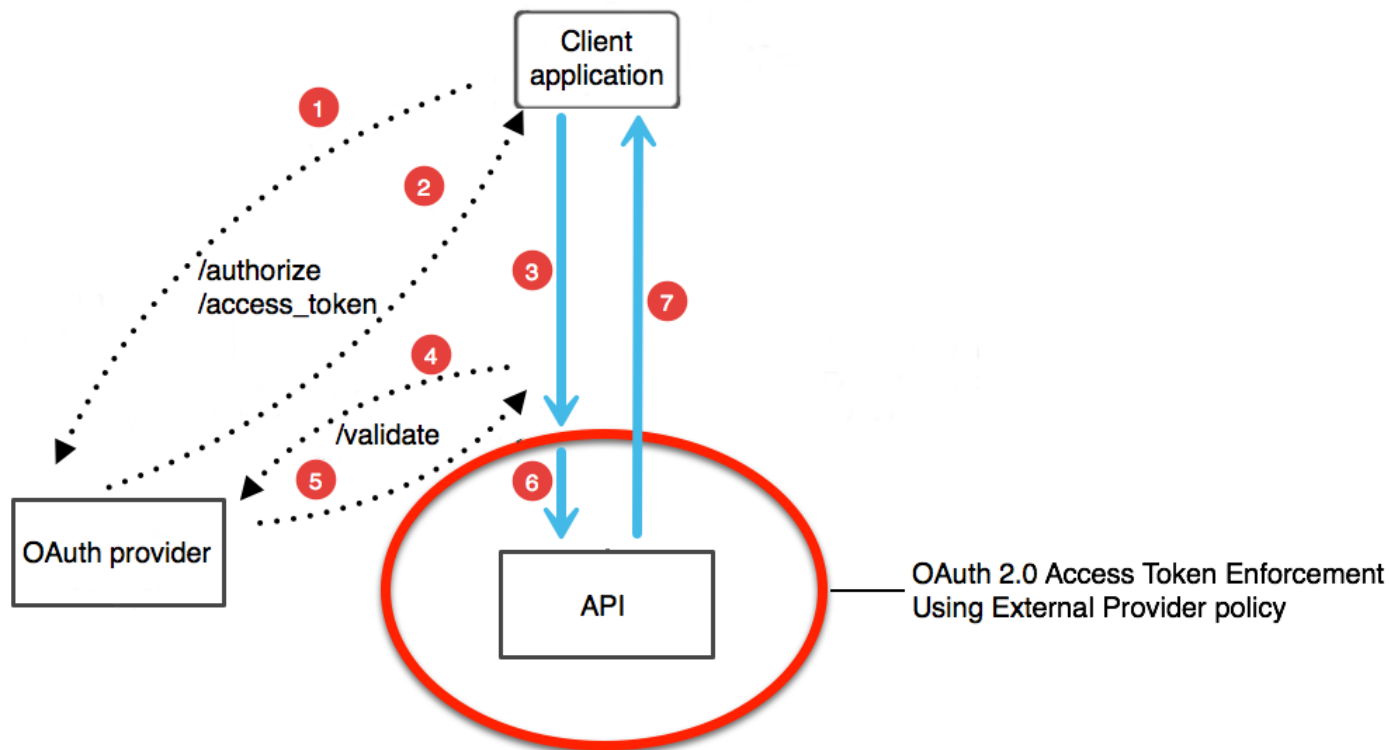
```
GET /users/username/account HTTP/1.1  
Host: example.org  
Authentication: hmac username:[digest]
```

■ OAuth

- 2007년 12월 OAuth 1.0 디지털 서명 기반 프레임워크
- Google은 2008년에 OAuth 1.0 지원을 시작. 2010년까지 Twitter는 모든 타사 앱이 OAuth 1.0 구현을 사용하도록 강제했다.
 - Google은 2012년 4월에 OAuth 1.0 지원 종료
- 2012년 10월에 OAuth 2.0이 출시



■ OAuth 인증 절차



■ OAuth2 특징

- 전송 종속 : 대부분의 보안 방어는 HTTPS/TLS에 위임. 오타, 부적절한 TLS 구성, 인증서 유효성 검사 실패 또는 기본 라이브러리의 취약점으로 인해 MiTM(Man-in-the-Middle) 공격이 발생하여 모든 OAuth 통신이 손상될 수 있다.
- 무기명 토큰 중심 : 통합하기 쉽지만 보안에는 좋지 않습니다. 무기명 토큰은 내부 보안 메커니즘을 제공하지 않습니다. 복사하거나 훔칠 수 있지만 구현하기가 더 쉽습니다.
- 더 쉽게 : OAuth 2.0은 훨씬 더 유용하지만 안전하게 빌드하기가 훨씬 더 어렵습니다.
- 유연성 : OAuth 1.0은 웹 워크플로만 처리했지만 OAuth 2.0은 웹이 아닌 클라이언트도 고려.
- 더 나은 업무 분리 : OAuth 2.0에서는 리소스 요청 처리와 사용자 권한 처리를 분리할 수 있습니다.

06. POSTMAN

■ POSTMAN이라는 REST Client

- chrome 브라우저의 확장 프로그램
- 단독 데스크탑 앱
- <https://www.postman.com/>



API Tools

A comprehensive set of tools that help accelerate the API Lifecycle - from design, testing, documentation, and mocking to discovery.



Workspaces

Organize your API work and collaborate with teammates across your organization or stakeholders across the world.



API Repository

Easily store, iterate and collaborate around all your API artifacts on one central platform used across teams.



Intelligence

Improve API operations by leveraging advanced features such as search, notifications, alerts and security warnings, reporting, and much more.

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

My Workspace

New Import

Overview

GET query=신혜선

No Environment

KakaoApi / query=신혜선

GET query=신혜선

GET image=신혜선

t.thinkbee/api

Todo

Blockchain_exam

bookdiary

Career_API

epost

Facebook

Jwt

Postman Echo

t.thinkbee/login

Xml_examples

문화데이터

Params

Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
<input checked="" type="checkbox"/> query	신혜선	필수, 검색어	
<input checked="" type="checkbox"/> sort		X (accuracy) accuracy (정확도순) or recen...	
<input checked="" type="checkbox"/> page	1	1~50 페이지 수	
<input checked="" type="checkbox"/> size	15	1~50 페이지 문서 개수	
Key	Value	Description	

Body Cookies Headers (11) Test Results

200 OK 91 ms 9.26 KB Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "documents": [
3      {
4        "contents": "로그인 후 편집 가능한 문서입니다. [ 글치기 · 집기 ] Artist <b>신혜선</b> 김인권 김현주
임세미 박경리 채종협 <b>신혜선</b> 申惠善 Shin Haesun[1] 출생 1989년 8월 31일 (32세)
서울특별시 국적 대한민국 신체 172~3...",
5        "datetime": "2021-10-16T00:00:00.000+09:00",
6        "title": "<b>신혜선</b> - 나무위키",
7        "url": "https://namu.wiki/w/%EC%98%A0%ED%98%9C%EC%84%A0"
      }
    ]
  }

```

Find and Replace Console

Bootcamp Runner Trash

■ HTTP methods

- HTTP methods 중 CRUD를 다룰 수 있는 메서드

Http Method	Description	
POST	Create	
GET	Retrieve	
PUT	Create or Update	
DELETE	Delete	
OPTION	Get available Method	

■ 연습: httpbin.org

○ GET: parameters

The screenshot shows the Postman interface for a GET request to `https://httpbin.org/get` with query parameters `param1=파라미터` and `value1=값`. The response status is 200 OK, and the body is a JSON object containing the request details.

Request: GET `https://httpbin.org/get?param1=파라미터&value1=값`

Query Params:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> param1	파라미터	
<input checked="" type="checkbox"/> value1	값	
Key	Value	Description

Body: Pretty, Raw, Preview, Visualize, JSON

```
1 {
2   "args": {
3     "param1": "파라미터",
4     "value1": "값"
5   },
6   "headers": {
7     "Accept": "*/*",
8     "Accept-Encoding": "gzip, deflate, br",
9     "Host": "httpbin.org",
```

■ 연습: httpbin.org

- Requestbody (주로 POST, PUT 등의 방식)
 - form-data
 - ✓ form 태그의 정보, Key - Value 조합을 submit
 - x-www-form-urlencoded:
 - ✓ form-data 형식과 같으나, 영문자를 제외한 글자는 모두 인코딩 한다.
 - ✓ Key - Value 조합으로 작성하며 텍스트 타입 전송만 가능하다. (파일 전송 불가)
 - raw:
 - ✓ 파라미터 형식과 내용을 직접 작성하는 경우 이쪽을 택한다.
 - binary:
 - ✓ 파일을 전송할 때 사용된다.

■ 연습: httpbin.org

- POST: xml-form-urlencoded 에 폼의 키/값을 주고 전송

The screenshot shows the Postman interface for a POST request to `https://httpbin.org/post`. The request body is set to `x-www-form-urlencoded` and contains the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	인공지능 핀테크	
<input checked="" type="checkbox"/> completed	false	
<input checked="" type="checkbox"/> note	튜토리얼 완성	
Key	Value	Description

The response is a JSON object, displayed in the 'Body' tab. The status is 200 OK, with a time of 621 ms and a size of 897 B. The JSON response is:

```
1 {
2   "args": {},
3   "data": "",
4   "files": {},
5   "form": {
6     "completed": "false",
7     "name": "인공지능 핀테크",
8     "note": "튜토리얼 완성"
9   }
10 }
```

■ 연습: httpbin.org

- POST: form-data 폼의 키/값을 주고 전송

The screenshot shows the Postman interface for a POST request to `https://httpbin.org/post`. The request is configured with the following details:

- Method:** POST
- URL:** `https://httpbin.org/post`
- Body Type:** form-data (selected)
- Body Data:**

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	인공지능 핀테크	
<input checked="" type="checkbox"/> completed	true	
<input checked="" type="checkbox"/> note	2022년 핀테크 아이디어	
Key	Value	Description

The response is displayed in the "Body" tab, showing the raw JSON data:

```

11  {
12    "Accept-Encoding": "gzip, deflate, br",
13    "Content-Length": "429",
14    "Content-Type": "multipart/form-data; boundary=-----255677451770090439833510",
15    "Host": "httpbin.org",
16    "Postman-Token": "702a1226-461c-4249-a99c-fd116a72ddfe",
17    "User-Agent": "PostmanRuntime/7.28.4",
18    "X-Amzn-Trace-Id": "Root=1-617296b0-7f27f01061b1afea570123d3"
19  },
20  "json": null,
  
```

Additional response details shown at the top right of the response section:

- Status: 200 OK
- Time: 200 ms
- Size: 966 B
- Save Response

■ 연습: httpbin.org

○ PUT: xml-form-urlencoded 사용해 수정할 사항을 전송

The screenshot shows a Postman interface for a PUT request to `https://httpbin.org/put`. The request body is set to `x-www-form-urlencoded` and contains the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	인공지능 핀테크	
<input checked="" type="checkbox"/> completed	true	
<input checked="" type="checkbox"/> note	완성	
Key	Value	Description

The response is shown in the bottom panel, displaying the raw JSON output:

```

11  {
12    "Accept-Encoding": "gzip, deflate, br",
13    "Content-Length": "110",
14    "Content-Type": "application/x-www-form-urlencoded",
15    "Host": "httpbin.org",
16    "Postman-Token": "af79d665-0d63-48dd-943c-ba3233981be2",
17    "User-Agent": "PostmanRuntime/7.28.4",
18    "X-Amzn-Trace-Id": "Root=1-617297cf-238e68f545a2d3ff283b0239"
19  },
20  "json": null,
  
```




2. NH 오픈뱅킹 개발자센터

01. 가입

02. API 사용

01. 가입

■ NH Developers

- <https://developers.nonghyup.com/>



■ NH Developers

○ <https://developers.nonhvu.com/>

STEP 01

개발자센터 회원가입 및 테스트용 인증키 발급

개발자센터 회원가입 후
개발, 테스트용 인증키를 발급

회원가입하기

STEP 02

API 개발 및 테스트

API 개발 가이드 및 오픈소스를 통해
API 개발 및 테스트를 진행

- 테스트 키 발급
- 테스트 계좌 및 환경 제공

STEP 03

NH 오픈플랫폼 약정 심사 및 승인

기본서류 제출 및 핀테크 서비스 적정성 심사,
정보보호 자가진단표 등 추가 서류 확인 후
서비스 약정 승인 및 계약사무 체결

- 서류 제출 및 심사
- 약정 승인(기관, 서비스)

STEP 04

실 보안인증키 발급 및 운영적용단계

보안인증키 발급 및 운영적용

- 인증키 발급
- 시스템운영 전환

■ 서비스 관리 화면

서비스 관리

회원 정보

예금주명

고푸릿

생년월일

1950-10-22



계좌정보

은행	구분	계좌번호	잔액
농협은행	모계좌	3020000005117	1,000,000,000
농협은행	자계좌	3020000005119	1,000,000,000
농협상호금융	모계좌	3510000005118	1,000,000,000

기관코드

Iscd

00

인증키

AccessToken

6614c

※ API 테스트 시, Request 값을 입력할때, 필수적으로 변경해야하는 항목 (Iscd, AccessToken)

카드번호

카드번호

9411123456782718

핀어카운트 발급번호

FinAcno

02. API 사용

REST API도구

- 별도의 클라이언트 프로그램 없이, Restful방식의 오픈API를 웹에서 테스트 할 수 있도록 도구와 환경을 제공

POST /OpenFinAccountDirect.nh 핀-어카운트직접발급

핀테크기업이 고객으로부터 직접 접수한 농협계좌에 대해 핀-어카운트 발급을 요청한다.

Parameters [Try it out](#)

Name	Description
param * required	
object (body)	param Example Value Model

```
{
  "Header": {
    "ApiVer": "OpenFinAccountDirect",
    "Isymd": "오늘날짜를입력하세요",
    "TrnBr": "112428",
    "Isd": "가관코드를입력하세요",
    "IntechApson": "001",
    "ApiSvcCd": "DrawingTransferA",
    "IS1und": "0000",
    "AccessToken": "인증키를입력하세요"
  },
  "DtrRgn": "Y",
  "BntBnm": "생년월일을입력하세요",
  "Bnd": "은행코드를입력하세요",
  "Acno": "계좌번호를입력하세요"
}
```

Parameter content type
application/json

■ REST API도구 : 핀-어카운트 직접발급 API

- 요청 URL: <https://developers.nonghyup.com/OpenFinAccountDirect.nh>
- 공통 헤더와 + API 필드

```
{
  "Header": {
    "ApiNm": "OpenFinAccountDirect",
    "Tsymd": "오늘날짜를입력하세요 YYYYMMDD",
    "Trtm": "112428",
    "Iscd": "기관코드",
    "FintechApsno": "001",
    "ApiSvcCd": "DrawingTransferA",
    "IsTuno": "0000",
    "AccessToken": "인증키"
  },
  "DrtrRgyn": "Y",
  "BrdtBrno": "생년월일",
  "Bncd": "은행코드",
  "Acno": "계좌번호"
}
```

#요청시 변경

테스트시 "ApiSvcCd":
"DrawingTransferA",

서비스관리
- 기관코드
- 인증키
- 계좌번호

■ REST API도구: 핀-어카운트 직접발급 API

oRgno 필드에 등록번호 발급

200
Undocumented

Response body

```
{
  "Rgno": "20211",
  "Header": {
    "Trtm": "112428",
    "Rsm": "정상처리 되었습니다.",
    "ApiNm": "OpenFinAccountDirect",
    "IsTuno": "0000",
    "Tsymd": "20211022",
    "FintechApsno": "001",
    "IsCd": "001162",
    "Rpd": "00000",
    "ApiSvcCd": "DrawingTransferA"
  }
}
```

Download

Response headers

```
access-control-allow-origin: *
connection: keep-alive
content-type: application/json;charset=UTF-8
date: Fri, 22 Oct 2021 11:56:55 GMT
server: nginx/1.16.1
transfer-encoding: chunked
vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers
```

서비스관리에서
확인 가능

핀어카운트 발급번호

FinAcno

00820100

■ REST API 도구

○ 예: 핀-어카운트 직접발급확인 API

- 요청 URL: <https://developers.nonghyup.com/CheckOpenFinAccountDirect.nh>
- 공통 헤더와 + API 필드

```
{
  "Header": {
    "ApiNm": "CheckOpenFinAccountDirect",
    "Tsymd": "오늘날짜를입력하세요 YYYYMMDD",
    "Trtm": "112428",
    "Iscd": "기관코드",
    "FintechApsno": "001",
    "ApiSvcCd": "DrawingTransferA",
    "IsTuno": "0000",          # 요청시 변경
    "AccessToken": "인증키"
  },
  "Rgno": "등록번호",
  "BrdtdBrno": "생년월일"
}
```

서비스관리

- 기관코드
- 인증키
- 계좌번호

■ REST API도구

○ 예: 핀-어카운트 직접발급확인 API

- 요청 URL: <https://developers.nonghyup.com/CheckOpenFinAccountDirect.nh>
- 공통 헤더와 + API 필드

200
Undocumented

Response body

```
{
  "FinAcno": "000201000",
  "Header": {
    "Trtm": "112428",
    "Rsms": "정상처리 되었습니다.",
    "ApiNm": "CheckOpenFinAccountDirect",
    "IsTuno": "0002",
    "Tsynd": "20211022",
    "FintechApno": "001",
    "Tscd": "001162",
    "Rpcd": "00000",
    "ApiSvcCd": "DrawingTransferA"
  },
  "RgsnYmd": "20211022"
}
```

Download

Response headers

```
access-control-allow-origin: *
connection: keep-alive
content-type: application/json;charset=UTF-8
date: Fri, 22 Oct 2021 12:48:43 GMT
server: nginx/1.16.1
transfer-encoding: chunked
vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers
```

■ REST API 도구

○ 예: 핀-어카운트 연결계좌 잔액조회 API

- 요청 URL: <https://developers.nonghyup.com/InquireBalance.nh>
- 공통 헤더와 + API 필드

```
{
  "Header": {
    "ApiNm": "InquireBalance",
    "Tsymd": "오늘날짜를입력하세요",
    "Trtm": "112428",
    "Iscd": "기관코드를입력하세요",
    "FintechApsno": "001",
    "ApiSvcCd": "ReceivedTransferA",
    "IsTuno": "0000",
    "AccessToken": "인증키를입력하세요"
  },
  "FinAcno": "핀어카운트"
}
```

서비스관리

- 기관코드
- 인증키
- 계좌번호

■ REST API도구

○ 예: 핀-어카운트 연결계좌 잔액조회 API

- 요청 URL: <https://developers.nonghyup.com/InquireBalance.nh>
- 공통 헤더와 + API 필드

200
Undocumented

Response body

```
{
  "FinAcno": "0082010",
  "Header": {
    "Trtm": "112428",
    "Rsm": "정상처리 되었습니다.",
    "ApiNm": "InquireBalance",
    "IsTuno": "0003",
    "Tsynd": "20211022",
    "FintechApsno": "001",
    "Tscd": "001162",
    "Rpcd": "00000",
    "ApiSvcCd": "ReceivedTransferA"
  },
  "Lcbl": "1000000000",
  "RlpmAbamt": "1000000000"
}
```

Download

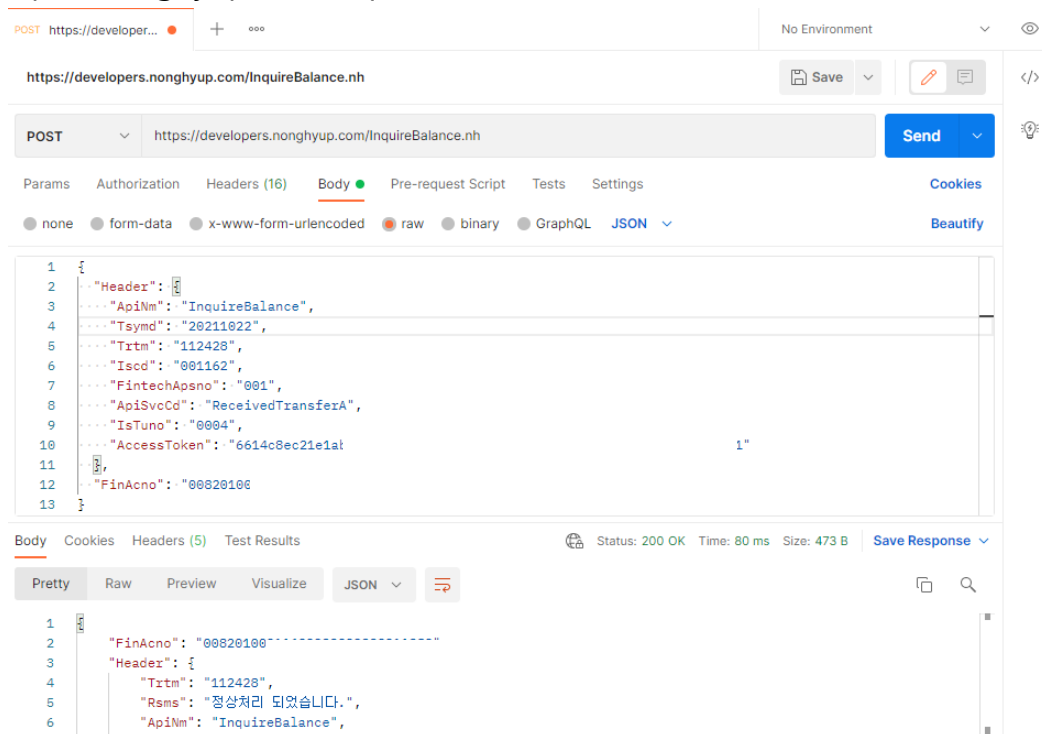
Response headers

```
access-control-allow-origin: *
connection: keep-alive
content-type: application/json;charset=UTF-8
date: Fri, 22 Oct 2021 12:55:04 GMT
server: nginx/1.16.1
transfer-encoding: chunked
vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers
```

POSTMAN 에서 API 사용

○ 핀-어카운트 연결계좌 잔액조회 API

– <https://developers.nonghyup.com/InquireBalance.nh>



POST의 raw

■ Python 에서 API 사용

- 핀-어카운트 연결계좌 잔액조회 API
- 깃헙 링크 노트북 활용
 - https://github.com/thinkbee/lecture_Fintech_2021