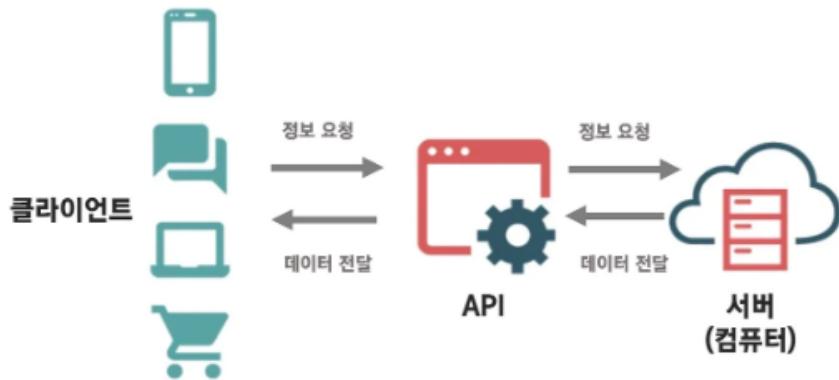


API

API의 정의



API(application programming interface)

: 애플리케이션이 어떤 프로그램이 제공하는 기능을 사용할 수 있게 만든 매개체

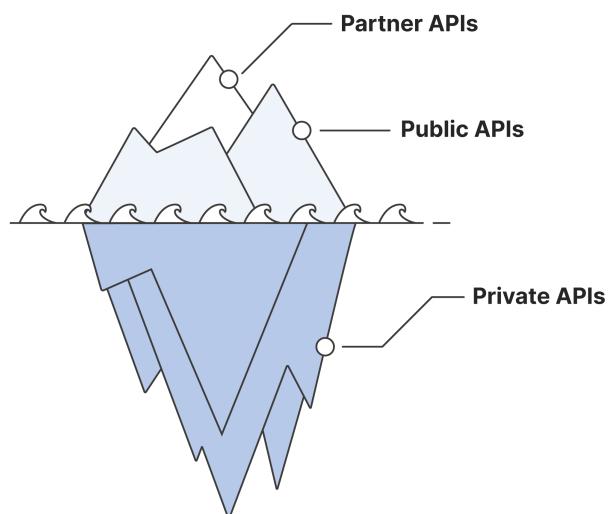
즉, 특정 목적을 가지고 탄생한 약속

: 역할

- 1) 서버와 데이터베이스에 대한 출입구 역할을 한다
- 2) 애플리케이션과 기기가 원활하게 통신할 수 있도록 한다
- 3) 모든 접속을 표준화 한다

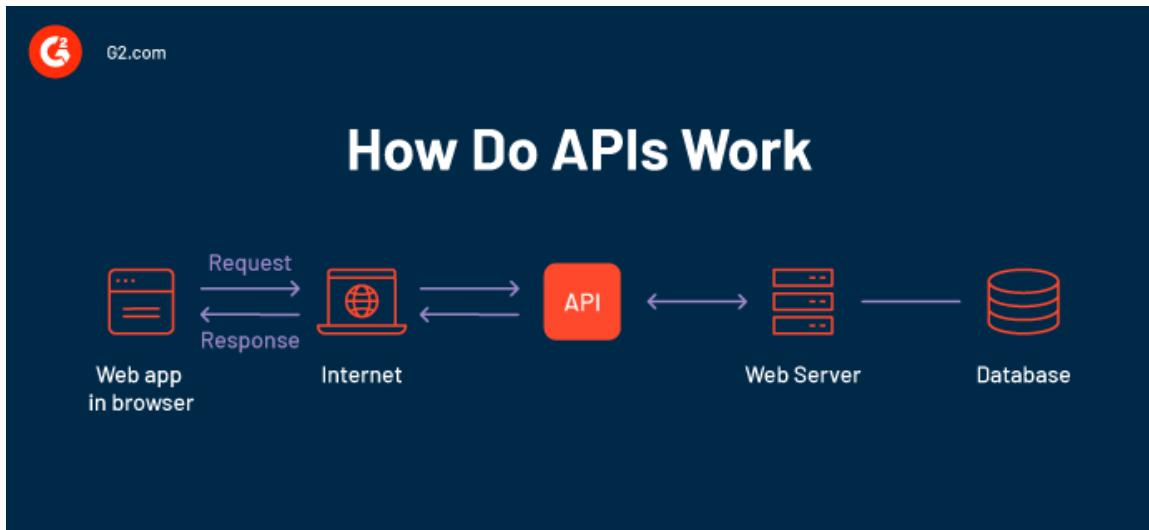
: 유형

- 접근 방식에 따라



- 1) private API - 내부 API로 제 3자에게 공개되지 않는다
 - 2) public API - 개방형 API로 모두에게 공개되며 누구나 해당 API를 사용할 수 있다
 - 3) partner API - 기업이나 데이터 공유에 동의하는 특정인들만 사용이 가능
- 아키텍처 스타일에 따라
SOAP, RPC, REST API, GraphQL이 존재하며 이 중 REST API가 가장 많이 사용된다

API의 작동방식



1. 클라이언트 프로그램이 정보를 받기 위해 API요청을 한다 (request)
2. API는 유효한 요청을 받은 후 제3의 앱 또는 웹서버를 호출한다
3. 서버는 요청받은 데이터를 전송함으로써 API에 응답한다
4. 데이터는 API를 통해 클라이언트 앱에 전송된다

API의 장단점

: 장점

- 1) 데이터 접속의 표준화와 편의성
 - 모든 접속을 표준화하기 때문에 조건만 맞다면 누구나 동일한 액세스를 약속함
 - 조직에서 애플리케이션을 개발할 때 기능적 API를 이용하면 자주 사용되는 기본 기능을 매번 개발 또는 업데이트를 할 필요가 없다
- 2) 자동화와 확장성
 - API를 통한 CRUD 처리에 따라 관련 데이터와 콘텐츠가 자동으로 생성되고 사용자의 환경에 맞춰서 정보가 전달됨
 - 개발 워크플로우가 간소화되고 애플리케이션 확장이 용이함
- 3) 적용력

→ API는 변화예측에 큰 도움이 된다

→ 유연한 서비스 환경을 구축할때, 소프트웨어를 통합할 때, 개발자들간의 협업이 필요할 때 용이함

: 단점

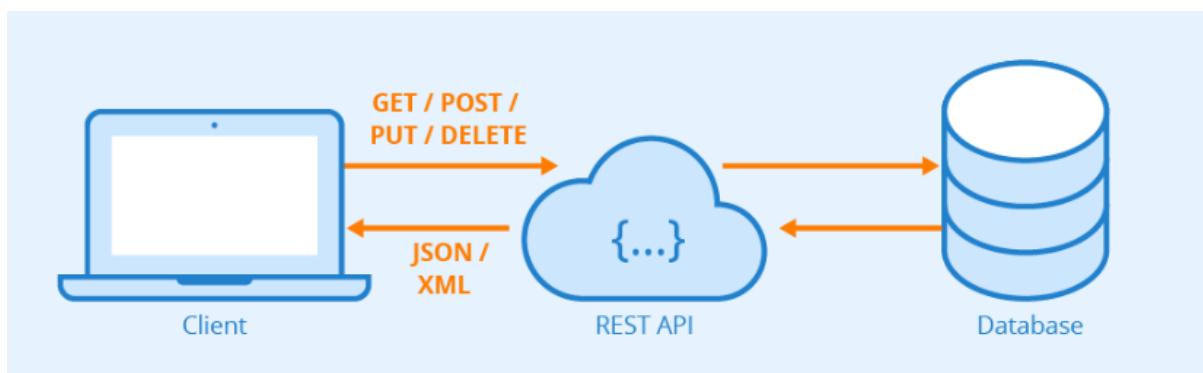
1) 보안성과 HTTP 방식의 제한문제

→ API의 구조적인 특성상 API의 단일 진입점 API게이트웨이는 해커의 타겟이 되어 CSRF, DDOS, XSS 등의 공격을 받을 수 있다

2) 표준의 부재와 개발 비용 문제

→ REST API 설계의 경우 공식화된 표준이 존재하지 않음

REST API



: REST API - REST 기반으로 서비스 API를 구현한 것

- REST = REpresentational State Transfer

자원의 이름으로 구분하여 해당 자원의 상태를 주고받는 모든 것

: 기본적으로 웹의 기존 기술과 HTTP 프로토콜을 그대로 사용하기 때문에 웹의 장점을 최대한 활용할 수 있다.

: REST의 특징

- 1) Uniform (유니폼 인터페이스)
- 2) Stateless(무상태성)
- 3) Cacheable(캐시가능)
- 4) Self-descriptiveness(자체 표현구조)
- 5) Client-Server구조
- 6) 계층형 구조

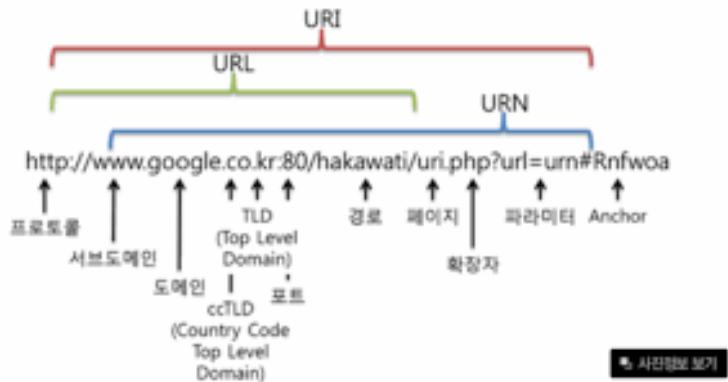
: REST API 디자인 가이드

= REST API 설계 시 가장 중요한 항목

1. URI는 정보의 자원을 표현해야 한다

*모든 resource(자원)에 고유한 ID가 존재하고 이러한 자원을 구별하는 ID는 HTTP URI

*URI 구조(출처. [URI \(velog.io\)](#))



▼ REST 구성요소 (해당 링크 참고)

REST 구성 요소

1. 자원(Resource): URI

- 모든 자원에 고유한 ID가 존재하고, 이 자원은 Server에 존재한다.
- 자원을 구별하는 ID는 '/groups/:group_id'와 같은 HTTP URI다.
- Client는 URI를 이용해서 자원을 지정하고 해당 자원의 상태(정보)에 대한 조작을 Server에 요청한다.

2. 행위(Verb): HTTP Method

- HTTP 프로토콜의 Method를 사용한다.
- HTTP 프로토콜은 GET, POST, PUT, DELETE 와 같은 메서드를 제공한다.

3. 표현(Representation of Resource)

- Client가 자원의 상태(정보)에 대한 조작을 요청하면 Server는 이에 적절한 응답 (Representation)을 보낸다.
- REST에서 하나의 자원은 JSON, XML, TEXT, RSS 등 여러 형태의 Representation으로 나타내어 질 수 있다.
- JSON 혹은 XML를 통해 데이터를 주고 받는 것이 일반적이다.

[Network] REST란? REST API란? RESTful이란? - Heee's Development Blog

Step by step goes a long way.

☞ <https://gmlwj9405.github.io/2018/09/21/rest-and-restful.html>



resource는 동사보다는 명사를, 대문자보다는 소문자를 사용한다.

ex. <https://mysite.com/getPosts> (X)

<https://mysite.com/createPost> (X)

<https://mysite.com/posts> (O)

resource의 도큐먼트 이름으로는 단수 명사를 사용해야 한다.

resource의 컬렉션 이름으로는 복수 명사를 사용해야 한다.

resource의 스토어 이름으로는 복수 명사를 사용해야 한다

ex. [GET /Member/1](#) (X)

[GET /members/1](#) (O)

2. 자원에 대한 행위는 HTTP Method(get, post, put delete)로 표현

METHOD	역할
POST	POST를 통해 해당 URI를 요청하면 리소스를 생성합니다.
GET	GET를 통해 해당 리소스를 조회합니다. 리소스를 조회하고 해당 도큐먼트에 대한 자세한 정보를 가져온다.
PUT	PUT를 통해 해당 리소스를 수정합니다.
DELETE	DELETE를 통해 리소스를 삭제합니다.

- RESTful

: 일반적으로 REST라는 아키텍처를 구현하는 웹 서비스를 나타내기 위해 사용되는 용어

: 공식적으로 발표된 용어는 아니지만 REST원리를 따르는 시스템을 RESTful하다고 말한다

참고자료

[API\(애플리케이션 프로그래밍 인터페이스\)란 - 개념, 기능, 장점 \(redhat.com\)](#)

[What Is an API? How APIs Improve Application Development \(g2.com\)](#)

[API 문서 투아보기 \(kakaoenterprise.com\)](#)

[\[간단정리\] REST, REST API, RESTful 특징 — 넌 잘하고 있어 \(tistory.com\)](#)

[\[Network\] REST란? REST API란? RESTful이란? - Heee's Development Blog \(gmlwj9405.github.io\)](#)