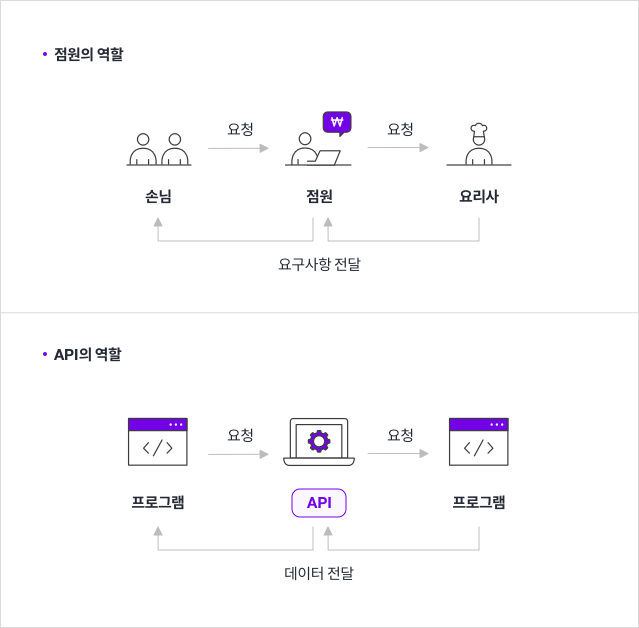
**1. 서버API 통신이란 무엇이고, REST API란 무엇인가.**



점원은 손님에게 메뉴를 알려주고, 주방에 주문받은 요리를 요청합니다. 그다음 주방에서 완성된 요리를 손님께 다시 전달하지요. API는 점원과 같은 역할을 합니다.

API는 손님(프로그램)이 주문할 수 있게 메뉴(명령 목록)를 정리하고, 주문(명령)을 받으면 요리사(응용프로그램)와 상호작용하여 요청된 메뉴(명령에 대한 값)를 전달합니다.

쉽게 말해, **API는 프로그램들이 서로 상호작용하는 것을 도와주는 매개체**로 볼 수 있습니다.

**1. API는 서버와 데이터베이스에 대한 출입구 역할을 한다.**

: 데이터베이스에는 소중한 정보들이 저장되는데요. 모든 사람들이 이 데이터베이스에 접근할 수 있으면 안 되겠지요. API는 이를 방지하기 위해 여러분이 가진 서버와 데이터베이스에 대한 출입구 역할을 하며, 허용된 사람들에게만 접근성을 부여해줍니다.

**2. API는 애플리케이션과 기기가 원활하게 통신할 수 있도록 한다.**

: 여기서 애플리케이션이란 우리가 흔히 알고 있는 스마트폰 어플이나 프로그램을 말합니다. API는 애플리케이션과 기기가 데이터를 원활히 주고받을 수 있도록 돕는 역할을 합니다.

**3. API는 모든 접속을 표준화한다.**

API는 모든 접속을 표준화하기 때문에 기계/ 운영체제 등과 상관없이 누구나 동일한 액세스를 얻을 수 있습니다. 쉽게 말해, API는 범용 플러그처럼 작동한다고 볼 수 있습니다.

**REST(Representational State Transfer)**는 **네트워크를 통해서 컴퓨터들끼리 통신할 수 있게 해주는 아키텍처 스타일**입니다.

REST API는 인터넷 식별자(URI)와 HTTP 프로토콜을 기반으로 합니다. REST는 HTTP 프로토콜 덕분에 ‘단순함’이 핵심이라고 할 수 있습니다. 데이터 포맷으로는 브라우저 간 호환성이 좋은 제이슨(JSON)을 사용합니다. REST API는 구축과 확장이 간단하지만, 크고 복잡하게 만들 수도 있습니다. 이는 API를 어떻게 만들고, 무엇을 추가하고, 어떤 목적으로 설계되었는지에 따라 달려있습니다.

REST API는 클라이언트와 서버 사이에서 통신할 수 있게 하고, 아키텍처를 만들 수 있게 해줍니다. REST 방식의 API라면, 클라이언트-서버 모델로 구축되었다는 것을 의미하며, 정보의 페이로드(실제 전달하려는 내용)가 두 지점 사이를 왕복하게 됩니다.

REST API는 단일한 인터페이스를 사용합니다. 이러한 점 때문에 해당 API를 사용하는 애플리케이션들이 동일한 경로를 통해서 접속해야 하고, 그 방식이 단순하게 됩니다. 여기에는 장점도 있고, 단점도 있기 때문에 향후 개발 과정에서 어떤 영향이 있는지에 대해서 알고 싶다면 전문가와 상의해보는 것이 좋습니다.

REST는 웹에 최적화되어 있고, 데이터 포맷이 JSON이기 때문에 브라우저들 간에 호환성이 좋습니다. 또한, 그 성능과 확장성이 뛰어난 것으로도 알려져 있죠. 하지만 다른 기술들과 마찬가지로 그 자체의 기능이 정지되거나 여러분의 앱을 먹통으로 만들 수도 있습니다. 그래서 REST로는 풀지 못하는 문제들을 해결하기 위해서 그래프 QL과 같은 언어가 생겨난 것입니다.

<https://gmlwjd9405.github.io/2018/09/21/rest-and-restful.html>