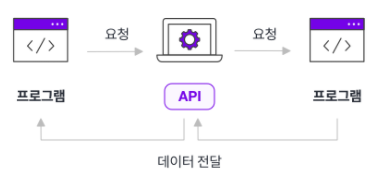
**API (Application Programming Interface)**



API는 **응용 프로그램에서 사용할 수 있도록 운영체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스**를 뜻한다. 예를들어 기상청에서 날씨정보를 얻고싶다면 다양한 방법이 있을 수 있다. 실시간으로 업데이트된 문서 다운로드, html로 작성된 파일 복사 등등이 있는데 API는 REST, SOAP 등의 방식으로 규격화하여 정보를 제공한다. API를 통해 요청하게되면 원하는 데이터를 Json이나 XML의 형태로 받을 수 있다.

​

**REST API**

어느정도 규격화한 것이 API지만 API도 종류가 있다. 대표적으로 REST, SOAP가 대표적이다.

REST API는 한 마디로 Restful한 API를 의미한다. 그럼 Restful하다는 것은 무엇일까?

​

**첫 번째,** URI는 정보의 자원을 표현해야 한다.

**두 번째,** 자원에 대한 행위는 HTTP Method(GET, POST, PUT, DELETE)로 표현한다.

이외에 여러가지 특징이 있지만 요약했을 때 이 2가지를 갖추면 **Restful**하다고한다.

​

정보의 자원을 표현해야한다는 말의 의미는 "GET, .com/members/delete/1"의 형태로 사용하지 말라는 것이다. URI는 단순히 자원만을 표현해야하고 CRUD와 같은 기능적인 내용은 HTTP Method를 활용하여 표현해야 한다.

​

**GET /members/delete/1 (x)**

**​**

**DELETE /members/1 (o)**

​

**POST, GET, PUT, DELETE**

|  |  |
| --- | --- |
| **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. |

**GET**

**리소스를 조회하고 가져온다. (SELECT)**

서버(혹은 DB)의 resource는 클라이언트로 전달만 될 뿐 변경되지 않는다.

예를 들어 웹브라우저에http://example.com/exmaple.png를 입력하면 해당 그림 파일이 표시되고, http://example.com/something을 입력하면 서버가 해당 [route](http://127.0.0.1:3001/ko/stubs/Route-Router)에 표시되어야 하는 페이지를 찾아 보여준다.

​

**POST**

**리소스를 생성한다. (INSERT)**

서버에게 resource를 보내면서 생성해 달라고 요청한다. 예를들어 회원가입을 하면 DB에 새로운 회원정보가 등록되고, 사진을 업로드 하면 그 사진이 웹사이트에 등록된다.

​

**PUT**

**리소스를 수정한다. (UPDATE)**

서버에게 resource의 업데이트 하거나 resource가 없다면 새로운 resource를 생성해 달라고 요청한다. 회원정보 수정 등에 사용된다.

​

**DELETE**

**리소스를 삭제한다. (DELETE)**

서버에게 resource의 삭제를 요청합니다.

​

​

**정리**

누군가 표준화하여 만든 데이터를 가져올때 표준화한 방식이 Restful하다면 이것을 REST API라고 부른다.