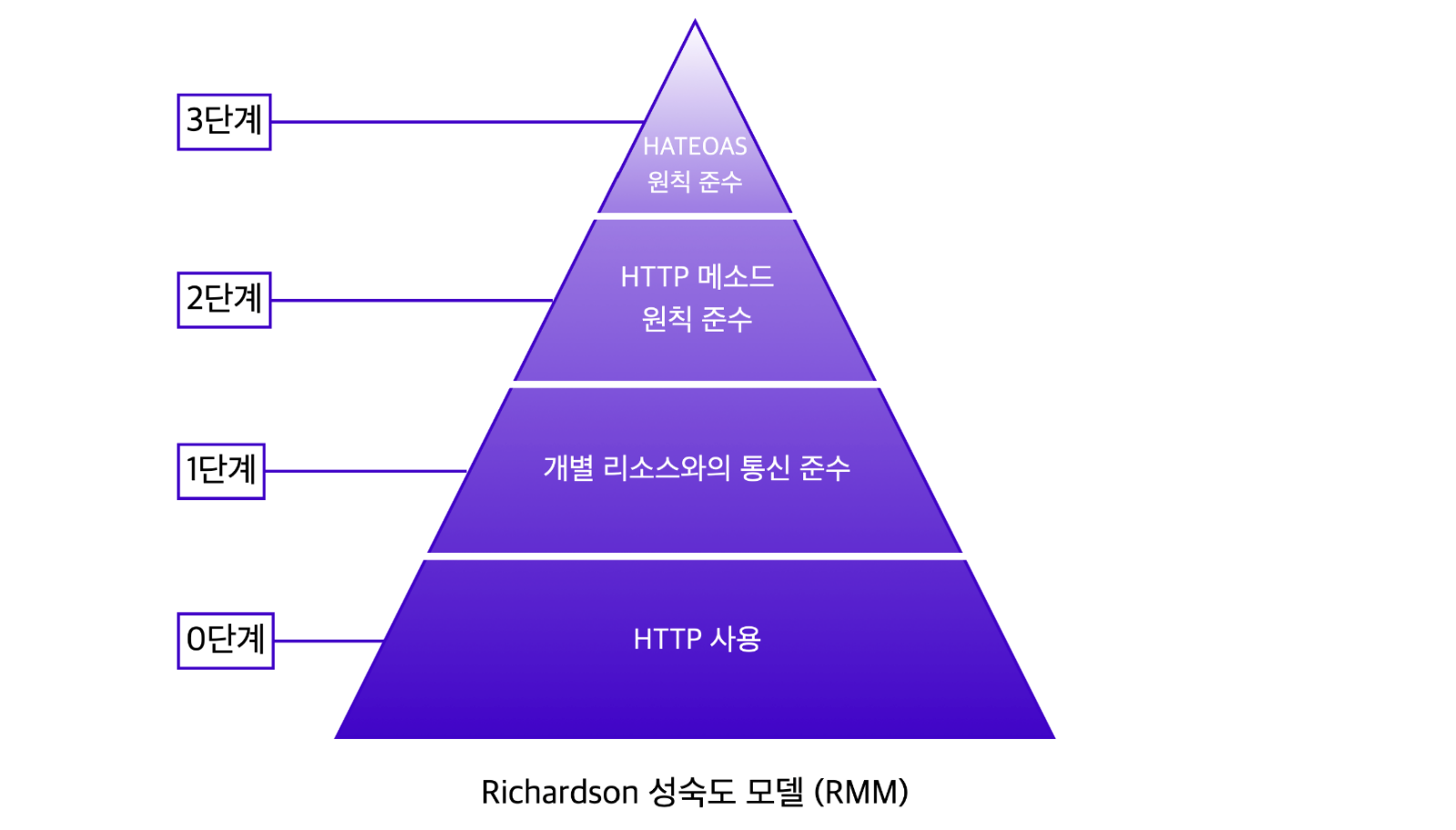
[네트워크 기초]-REST API, Open API

클라이언트와 서버의 통신을 담당하는 API는 요청과 응답을 통해 통신하는 과정에서 각각의 메시지를 잘 보낼 수 있는 방법이 존재합니다.

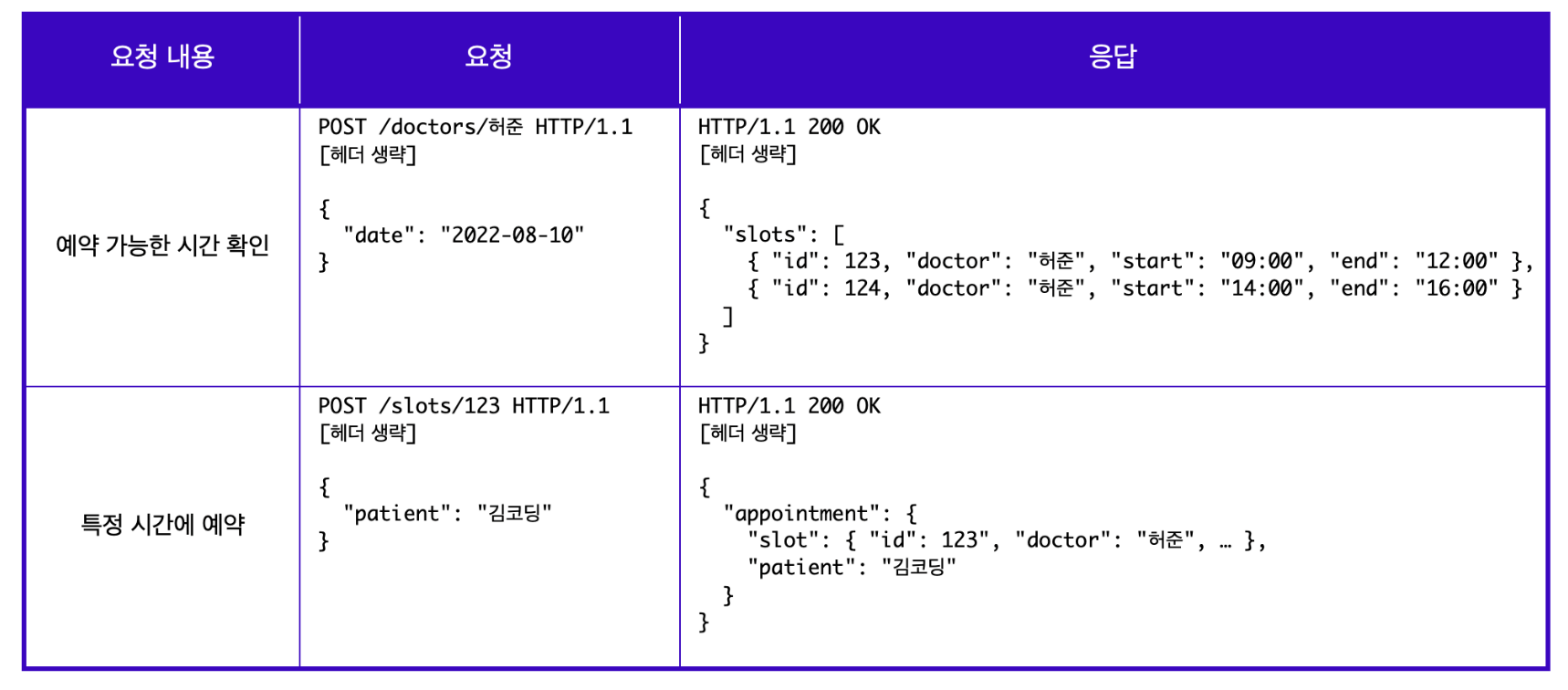
REST API에서 REST는 Representational State Transfer의 약자로 웹에서 사용되는 데이터나 자원을 HTTP URI로 표현하고, HTTP 프로토콜을 통해 요청과 응답을 정의하는 방식을 말합니다.



이런 REST API를 작성하기 위해 지켜야 할 몇 가지 규칙이 존재합니다.

REST 성숙도 모델은 위 그림에서 볼 수 있는 것처럼 0~3단계로 이루어져 있지만, 실제로는 3단계까지 엄격하게 적용하기 어렵기 때문에 2단계까지만 적용해도 좋은 API 디자인이라고 할 수 있고, 이런 API를 HTTP API라고도 부릅니다.

 0단계에서는 요청과 응답에 단순히 HTTP 프로토콜을 사용하기만 해도 되며, 이 단계가 REST API를 작성하기 위한 기본이 됩니다.

 1단계에서는 단순히 HTTP 프로토콜만 사용하는 것이 아니라, 개별 리소스와의 통신을 준수해야 합니다.

REST API는 웹에서 사용되는 모든 데이터나 자원을 HTTP URI로 표현해야 하기 때문에, 모든 자원은 개별 리소스에 맞는 엔드포인트를 사용해야 하며 요청에 대한 정보를 응답으로 전달해야 합니다.

엔드포인트는 어떤 리소스를 변화시키는지 혹은 어떤 응답이 제공되는지에 따라 달라집니다. 엔드포인트 작성 시에는 동사나 HTTP 메소드, 어떤 행위에 대한 단어 사용은 지양하고, 리소스에 집중해 명사 형태로 작성하는 것이 바람직합니다. 또한 리소스 사용에 관한 요청을 받았을 시, 사용된 리소스에 대한 정보와 함께 리소스 사용에 대한 성공/실패 여부를 반환해야 합니다.

REST 2단계에서는 CRUD에 맞게 적절한 HTTP 메소드를 사용하는 것이 중요합니다. 리소스에 대한 클라이언트의 각각의 요청이 CRUD 중 어느 부분에 속하는지에 따라 그에 맞는 메소드를 사용하고, 응답 역시 상태 코드를 사용해서 요청의 결과를 정확하게 알려주어야 합니다.

2단계까지 충족했다면 대체적으로 잘 작성된 API라고 할 수 있습니다.

 마지막 단계는 HATEOAS(Hypermedia As The Engine Of Application State)라는 약어로 표현되는 하이퍼미디어 컨트롤을 적용합니다. 3단계의 요청은 2단계와 동일하지만 응답은 리소스의 URI를 포함한 링크 요소를 삽입해 작성해야 합니다. 위 이미지에서 볼 수 있는 것처럼 응답에 새로운 링크를 넣어 새로운 기능에 접근할 수 있도록 하는 것이 3단계의 핵심입니다.