**코멘토 직무 부트캠프 – 필수 과제 : Rest API란 무엇인가?**

1) HTTP 통신

HTTP(HyperText Transfer Protocol)는 W3(World Wide Web) 상에서 정보를 주고받을 수 있는 전송규약이다. 주로 HTML 문서를 주고받는 데에 쓰인다.

HTTP는 클라이언트와 서버 사이에 이루어지는 요청/응답(request/response) 프로토콜이다. 예를 들어, 클라이언트인 웹 브라우저가 HTTP를 통하여 서버로부터 웹페이지(HTML)나 그림 정보를 요청하면, 서버는 이 요청에 응답하여 필요한 정보를 해당 사용자에게 전달하게 된다. 이 정보가 모니터와 같은 출력 장치를 통해 사용자에게 나타나는 것이다.

1-1) HTTPS

HTTPS(HyperText Transfer Protocl Secure)는 W3 통신 프로토콜인 HTTP의 보안이 강화된 버전이다. HTTPS는 데이터 전송의 보안을 강화하기 위해 암호화되며, 사용자가 은행 계좌, 이메일 서비스, 의료 보험 공급자에 로그인하는 등 중요한 데이터를 전송할 때 특히 중요하다.

로그인 자격 증명이 필요한 웹 사이트는 HTTPS를 사용해야하며, Google Chrome 및 기타 브라우저에서는 HTTPS가 아닌 모든 웹 사이트는 안전하지 않은 것으로 표시된다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

1-2) HTTP 요청 메서드

- GET : 특정 리소스의 표시를 요청. GET을 사용하는 요청은 오직 데이터를 받기만 한다.

- HEAD : GET 메서드의 요청과 동일한 응답을 요구하지만, 응답 본문을 포함하지 않는다.

- POST : 서버에 데이터를 추가하거나 작성. 데이터를 생성하는 것이기 때문에 요청 시에 Body 값과 Content-Type 값을 작성해야 한다.

- PUT : 리소스를 생성 / 업데이트하기 위해 서버로 데이터를 보내는 데 사용된다. POST와 마찬가지로 Body값과 Content-Type 값을 작성해야 한다.

- DELETE : 특정 리소스를 삭제한다.

- CONNECT : 프록시 동작의 터널 접속을 변경한다.

- OPTIONS : 목적 리소스의 통신을 설정하는 데 쓰인다.

- TRACE : 목적 리소스의 경로를 따라 메시지 loop-back 테스트를 한다.

- PATCH : 리소스의 일부분을 수정한다.

※ PUT과 PATCH는 차이점이 있다. PUT은 지정한 데이터를 전부 수정하지만, PATCH는 정보의 일부분이 변경되는 방법이다. 그렇기에 PUT은 멱등하지만[[1]](#footnote-1), PATCH는 멱등하다고 볼 수 없다.

2) 웹 브라우저에서 URL을 입력하여 서버에서 응답하는 과정

1. 웹 브라우저 주소창에 URL을 입력한다.

2. 웹 브라우저가 도메인의 IP 주소를 찾기 위해 캐시에서 DNS 기록을 조회한다.

- DNS(Domain Name System)는 웹사이트의 IP 주소와 도메인 주소를 연결해주는 시스템이다. 인터넷의 URL에 할당되어 있는 IP 주소를 외우지 않고도 사이트에 쉽게 접속할 수 있게끔 매핑해주는 역할이다.

- DNS 기록을 찾기 위해서 브라우저는 네 개의 캐시(브라우저 캐시, OS 캐시, 라우터 캐시, ISP 캐시)를 확인한다.

2-2. 요청한 URL이 캐시에 없다면, ISP의 DNS 서버가 DNS 쿼리로 도메인을 호스팅하는 서버의 IP주소를 찾는다.

- DNS 쿼리로 웹 사이트에 대한 올바른 IP 주소를 찾을 때까지 인터넷에서 여러 DNS 서버를 검색한다. 필요한 IP 주소를 찾거나, 찾을 수 없다는 오류 응답을 반환할 때까지 한 DNS 서버에서 다른 DNS 서버로 검색이 반복적으로 계속되기 때문에 이 유형의 검색을 재귀적 질의(Recursive Query)라고 한다.

- ISP의 DNS 서버를 DNS 리커서(DNS Recursor)라고 부르는데, DNS 리커서는 인터넷의 다른 DNS 서버에 답변을 요청하여 의도된 도메인 이름의 적절한 IP 주소를 찾는 일을 담당한다.

3. 웹 브라우저가 찾은 IP 주소를 기반으로 서버와의 TCP 연결을 시작한다.

- 브라우저가 올바른 IP 주소를 수신하면 IP 주소와 일치하는 서버와 연결해 정보를 전송한다. 브라우저는 인터넷 프로토콜을 사용하여 연결을 구축하며, 사용할 수 있는 여러가지 인터넷 프로토콜이 있지만 일반적으로 HTTP 요청에서는 TCP(Transmission Control Protocol)라는 전송 제어 프로토콜을 사용한다.

- 클라이언트와 서버 간에 데이터 패킷을 전송하려면 TCP 연결을 해야 한다. 연결은 TCP/IP 3-way handshake 라는 연결 과정을 통해 이루어진다. 클라이언트와 서버가 SYN(연결 요청) 및 ACK(승인) 메시지를 교환하여 연결을 설정하는 3단계 프로세스이다.

a) 클라이언트는 인터넷을 통해 서버에 SYN 패킷을 보내 새 연결이 가능한지 여부를 묻는다.

b) 서버에 새 연결을 수락할 수 있는 열린 포트가 있는 경우, SYN/ACK 패킷을 사용하여 SYN 패킷의 ACK(승인)으로 응답한다.

c) 클라이언트는 서버로부터 SYN/ACK 패킷을 수신하고 ACK 패킷을 전송하여 승인한다.

4. 웹 브라우저가 HTTP 요청을 서버로 전송한다. (필요한 경우, HTTPS 보안 통신이 진행된다.)

- TCP 연결이 설정되면 데이터 전송이 시작되고, 브라우저는 웹 페이지를 요청하는 GET 요청을 보낼 것이다. 만약 자격 증명을 입력하거나 form을 제출하는 경우 POST 요청을 사용할 수 있다.

5. 웹 서버가 요청을 처리하고 응답을 다시 웹 브라우저로 전송한다.

- 브라우저로부터 요청을 수신하고, 해당 내용을 Request handler에 전달하여 응답을 읽고 생성하는 역할을 한다. 그런 다음 response를 특정 포맷(JSON, XML, HTML)으로 작성한다.

- 서버 응답에는 요청한 웹 페이지와 함께 상태 코드, 압축 유형, 페이지 캐싱 방법, 설정할 쿠키, 개인 정보 등이 포함된다.

5-1. 상태코드 설명

- 1xx (Information Response) : 정보 메시지만을 나타낸다. 서버가 요청을 받았으며 서버에 연결된 클라이언트는 계속해서 작업을 하라는 뜻.

- 2xx (Successful Response) : 서버와의 요청이 성공함을 나타냄.

- 3xx (Redirection Message) : 요청 완료를 위해 추가 작업 조치가 필요함을 의미함.

- 4xx (Client Error Response) : 클라이언트의 Request에 에러가 있음을 의미함.

- 5xx (Server Error) : 서버 측의 오류로 request를 수행할 수 있음.

6. 웹 브라우저가 전송 받은 콘텐츠를 렌더링한다.

- 브라우저는 GET 요청을 통해 HTML을 반환하고, HTML을 파싱하고 렌더링할 떄 CSS, 이미지 및 데이터 등 추가 요청을 보내고 마지막으로 페이지를 사용자에게 보여준다.

**참고 자료**

<https://ko.wikipedia.org/wiki/HTTP>

<https://www.cloudflare.com/ko-kr/learning/ssl/what-is-https/>

<https://developer.mozilla.org/ko/docs/Web/HTTP/Methods>

<https://velog.io/@yh20studio/CS-Http-Method-%EB%9E%80-GET-POST-PUT-DELETE>

<https://trend21c.tistory.com/2300>

<https://aws.amazon.com/ko/blogs/korea/what-happens-when-you-type-a-url-into-your-browser/>

1. 연산을 여러 번 적용하더라도 결과가 달라지지 않는 성질. HTTP 메서드에서는 멱등성이 유지되어야 하는 경우, 같은 행위를 여러 번 반복하더라도 서버의 상태가 동일하게 남아야 한다. [↑](#footnote-ref-1)