

## HTTP verbs

우선 기본적인 지식부터 확인하고 가자. HTTP 리퀘스트에는 메소드(method) 개념이 있다. html에서 폼요청시 mehtod 속성을 설정하는데 보통 GET, POST를 사용한다. 이외에도 표준 스펙에서는 HEAD, PUT, DELETE, TRACE, OPTIONS, CONNECT 등 10가지 이상의 메소드를 정의한다. 서비스를 구현할 때는 보통 REST API를 위해 4가지 메소드만 사용해도 충분하다.

- POST
- GET
- PUT
- DELETE

이 메소드는 "동사 역할"을 한다. 동사에 대한 목적어가 바로 URI다. <sup>①</sup> 예를 들어 GET 메소드를 이용해 /users 라는 URI를 호출하면 유저 정보를 조회를 요청하는 프로토콜이라고 해석한다. <sup>②</sup> 같은 URI에 POST 메소드로 호출하면 새로운 유정 정보를 "서버에 추가"하는 요청이다. 이러한 구조로 익스프레스의 라우팅을 구현한다.

과거에 5년차 이상 서버 개발자와 일한 경험이 있는데, 로그인 API를 요청했는데 GET 과 POST 둘 다 요청이 가능하도록 만들어 주는 것이었다... 암호화를 한것도 아니라서 POST도 완벽한 보안이 되는건 아니지만 정말 최소한의 보안 의식도 없는건가 아니면 구분을 못해서 그런건가 싶었다. 이렇듯 HTTP 통신을 제대로 사용하기 위해서도 둘의 차이를 아는 게 중요하다.

## GET 이란?

---

GET 은 클라이언트에서 서버로 어떠한 리소스로 부터 정보를 요청하기 위해 사용되는 메서드이다.

예를들면 게시판의 게시물을 조회할 때 쓸 수 있다.

GET을 통한 요청은 URL 주소 끝에 파라미터로 포함되어 전송되며, 이 부분을 쿼리 스트링 (query string) 이라고 부른다.

방식은 URL 끝에 "? " 를 붙이고 그다음 변수명1=값1&변수명2=값2... 형식으로 이어 붙이면 된다.

예를들어 다음과 같은 방식이다.

`www.example.com/show?name1=value1&name2=value2`

서버에서는 name1 과 name2 라는 파라미터 명으로 각각 value1 과 value2 의 파라미터 값을 전달 받을 수 있다.

## GET의 특징

---

- GET 요청은 캐시가 가능하다. : GET을 통해 서버에 리소스를 요청할 때 웹 캐시가 요청을 가로채(서버로부터 리소스를 다시 다운로드하는 대신) 리소스의 복사본을 반환한다. HTTP 헤더에서 cache-control 헤더를 통해 캐시 옵션을 지정할 수 있다.
- GET 요청은 브라우저 히스토리에 남는다.
- GET 요청은 북마크 될 수 있다.
- GET 요청은 길이 제한이 있다. : GET 요청의 길이 제한은 표준이 따로 있는건 아니고 브라우저마다 제한이 다르다고 한다.
- GET 요청은 중요한 정보를 다루면 안된다. ( 보안 ) : GET 요청은 파라미터에 다 노출되어 버리기 때문에 최소한의 보안 의식이라 생각하자.
- GET은 데이터를 요청할때만 사용 된다.

When an HTML form is submitted without specifying a method, what is the default HTTP method used? GET or POST?

Has this behaviour ever changed between HTML standards?

Please cite a W3C standard document if possible.

If not specified, the default is GET. I see no indication that this behaviour has ever been different. And it makes sense the GET is the default, as it specifies that method should be used for actions without side effects.

## POST 란?

POST는 클라이언트에서 서버로 리소스를 생성하거나 업데이트하기 위해 데이터를 보낼 때 사용 되는 메서드다. 예를들면 게시판에 게시글을 작성하는 작업 등을 할 때 사용할 된다.

POST는 전송할 데이터를 HTTP 메시지 body 부분에 담아서 서버로 보낸다. ( body 의 타입은 Content-Type 헤더에 따라 결정 된다 .)

GET에서 URL 의 파라미터로 보냈던 name1=value1&name2=value2 가 body에 담겨 보내진다고 생각하면 된다.

POST 로 데이터를 전송할 때 길이 제한이 따로 없어 용량이 큰 데이터를 보낼 때 사용하거나 GET처럼 데이터가 외부적으로 드러나는 건 아니어서 보안이 필요한 부분에 많이 사용된다.

★ 하지만 데이터를 암호화하지 않으면 body의 데이터도 결국 볼 수 있는건 똑같다. )

POST를 통한 데이터 전송은 보통 HTML form 을 통해 서버로 전송된다.

## GET 과 POST 의 차이점

GET과 POST의 특징만 보아도 차이가 나긴하지만 추가적으로 차이점을 정리해보면 다음과 같다.

- **사용목적** : GET은 서버의 리소스에서 데이터를 요청할 때, POST는 서버의 리소스를 새로 생성하거나 업데이트할 때 사용한다. DB 로 따지면 GET은 SELECT 에 가깝고, POST는 Create 에 가깝다고 보면 된다.
- **요청에 body 유무** : GET 은 URL 파라미터에 요청하는 데이터를 담아 보내기 때문에 HTTP 메시지에 body가 없다. POST 는 body 에 데이터를 담아 보내기 때문에 당연히 HTTP 메시지에 body가 존재한다.
- **멱등성 (idempotent)** : GET 요청은 멱등이며, POST는 멱등이 아니다.

### 멱등이란?

멱등의 사전적 정의는 연산을 여러 번 적용하더라도 결과가 달라지지 않는 성질을 의미한다.

GET은 리소스를 조회한다는 점에서 여러 번 요청하더라도 응답이 똑같은 것이다. 반대로 POST는 리소스를 새로 생성하거나 업데이트할 때 사용되기 때문에 멱등이 아니라고 볼 수 있다. (POST 요청이 발생하면 서버가 변경될 수 있다.)

	GET	POST
캐시	○	×
브라우저 기록	○	×
북마크 추가	○	×
데이터 길이 제한	○	×
HTTP 응답 코드	200(Ok)	201(Created)
언제 주로 사용하는가?	리소스 요청	리소스 생성
리소스 전달 방식	쿼리스트링	HTTP Body
idempotent	○	×