**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.

**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 보내진다 생각하면 된다.

POST 로 데이터를 전송할 때 길이 제한이 따로 없어 용량이 큰 데이터를 보낼 때 사용하거나 GET처럼 데이터가 외부적으로 드러나는건 아니라서 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.

**POST의 특징**

* POST 요청은 캐시되지 않는다.
* POST 요청은 브라우저 히스토리에 남지 않는다.
* POST 요청은 북마크 되지 않는다.
* POST 요청은 데이터 길이에 제한이 없다.

**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.

DB로 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 똑같을 것 이다. 반대로 POST는 리소스를 새로 생성하거나 업데이트할 때 사용되기 때문에 멱등이 아니라고 볼 수 있다. (POST 요청이 발생하면 서버가 변경될 수 있다.)

GET과 POST는 이런 차이들이 있기 때문에 사용하려는 목적에 맞는 확인한 후에 사용해야한다.

GET과 POST 이외에도 PUT , DELETE 등을 적절히 사용하는게 좋은데 예를들어 봇의 경우에 사이트를 돌아다니면서 GET 요청을 날린다. 이럴 때 DELETE 등을 GET으로 처리하면 봇에 의해 서버에 있는 리소스들이 삭제 되는 상황이 일어 날수 있다! 항상 기술을 사용할 때 **왜** 를 잊지말자.

출처: <https://noahlogs.tistory.com/35> [인생의 로그캣]  
  
출처: <https://noahlogs.tistory.com/35> [인생의 로그캣]