



Form과 Method, 헤더

허강준

충남대학교 정보보호동아리 ARGOS

2021. 04. 10

Review: HTTP 패킷의 구조

■ 패킷 = 헤더 + 본문(Body)

```
GET https://www.argos.or.kr/ HTTP/1.1
Host: www.argos.or.kr
Connection: keep-alive
Cache-Control: max-age=0
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; win64; x64) App
Accept: text/html,application/xhtml+xml,application/xml;q
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: ko,en;q=0.9,en-US;q=0.8
Cookie: PHPSESSID= ; mobile=fals
```

```
HTTP/1.1 200 OK
Server: nginx/1.4.6 (Ubuntu)
Date: Tue, 23 Mar 2021 05:05:38 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.5.9-1ubuntu4.29
Expires: Mon, 26 Jul 1997 05:00:00 GMT
Last-Modified: Tue, 23 Mar 2021 05:05:38 GMT
Cache-Control: no-store, no-cache, must-revalidate
Cache-Control: post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 43873

<!DOCTYPE html>
<html lang="ko">
<head>
<!-- META -->
<meta charset="utf-8">
<meta name="Generator" content="XpressEngine">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

■ 요청 패킷 (GET)

■ 응답 패킷

HTML Form 태그

- Form: 서식
- 회원가입, 로그인, 게시물 작성, 검색창, 등등...
- 사용자와 서버를 연결해주는 UI(User Interface)를 구성함
- 입력창, 체크박스, 버튼 등등...



The image shows the Naver login interface. At the top is the green 'NAVER' logo. Below it are two input fields: '아이디' (ID) and '비밀번호' (Password). A large green button labeled '로그인' (Login) is positioned below the password field. At the bottom left, there is a link for '로그인 실패 유지' (Keep login failed) and at the bottom right, a link for 'IP 보안 ON' (IP Security ON).



The image shows a member registration form titled '회원가입' (Sign Up). It contains several input fields with labels: '이름' (Name), '비밀번호' (Password), '비밀번호 확인' (Confirm Password), '이메일 주소' (Email Address), '입학연도' (Enrollment Year), '이메일' (Email), and '닉네임' (Nickname). Below the email address field, there is a text line: '2010년 입학생의 입학연도 = 10 2015년 입학생의 입학연도 = 15'. At the bottom, there are input fields for '이름' (Name) and '닉네임' (Nickname).



The image shows the Google search homepage. It features the multi-colored 'Google' logo at the top. Below the logo is a large search bar with a magnifying glass icon on the right. Under the search bar are two buttons: 'Google Search' and 'I'm Feeling Lucky'. At the very bottom, there is a small line of text: 'Google offered in: 한국어 | 검색 | 즐겨찾기'.

HTML Form 태그

```
<form action="target_address" method="method">  
  <!-- 여기에 input 등의 태그를 집어넣기 -->  
</form>
```

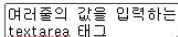
- action: 폼 데이터가 전송될 위치(주소)
- method: 폼 데이터를 전송하는 방법(Method)
GET, POST, PUT, DELETE 등.. (뒤에 자세히 설명)

HTML Form Input: select

```
<select name="favorite">
  <option value="">선택해주세요!</option>
  <option value="cse" selected>컴퓨터융합학부</option>
  <option value="chem">화학공학과</option>
  <option value="phys">물리학과</option>
  <option value="krlit">국어국문학과</option>
</select>
```

HTML Form Input: textarea

textarea



여러줄의 값을 입력하는
textarea 태그

- text 나 password와 비슷하지만 여러줄 입력 가능
- <textarea> 태그 사용
- 값은 <textarea>와 </textarea> 사이에 입력

HTML Form Input: submit

input type=submit

등록하기

- text 나 password와 비슷하지만 여러줄 입력 가능
- <textarea> 태그 사용
- 값은 <textarea>와 </textarea> 사이에 입력

HTTP GET

- 자원에 대한 "표현" 요청
- 주소에 데이터를 삽입함
`http://example.com/index.php?n=example`
- 여러 데이터의 경우 & 를 이용하여 연결
`http://example.com/index.php?n=example&n2=example2`
- 주로 검색창이나 글 읽기 등에 사용
- 구현이 간편하나 데이터가 주소에 그대로 표현되므로
보안이 중요하면 사용 금지

HTTP GET in PHP

- \$_GET 은 GET으로 전달받은 데이터를 보관하는 "배열"
- \$_GET["name"] 식으로 전달된 데이터를 사용할 수 있음

HTTP GET in PHP

← → ↻ ⓘ localhost:60000/sample_form.php

로그인(GET)

test ... 로그인

← → ↻ ⓘ localhost:60000/sample_form_proc_get.php?id=test&password=1234

```
array(2) { ["id"]=> string(4) "test" ["password"]=> string(4) "1234" }
```

HTTP POST

- \$_POST 는 GET으로 전달받은 데이터를 보관하는 "배열"
- 마찬가지로 \$_POST["name"] 식으로 전달된 데이터를 사용할 수 있음

Request: GET vs POST

```
GET http://localhost:60000/sample_form_proc_get.php?id=test&password=1234 HTTP/1.1
Host: localhost:60000
Connection: keep-alive
Cache-Control: max-age=0
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:60000/sample_form.php
Accept-Encoding: gzip, deflate, br
Accept-Language: ko,en;q=0.9,en-US;q=0.8
```

```
POST http://localhost:60000/sample_form_proc_post.php HTTP/1.1
Host: localhost:60000
Connection: keep-alive
Content-Length: 21
Cache-Control: max-age=0
Origin: http://localhost:60000
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:60000/sample_form.php
Accept-Encoding: gzip, deflate, br
Accept-Language: ko,en;q=0.9,en-US;q=0.8

id=test&password=1234
```

Header in Packet

```
POST http://localhost:60000/sample_form_proc_post.php HTTP/1.1
Host: localhost:60000
Connection: keep-alive
Content-Length: 21
Cache-Control: max-age=0
Origin: http://localhost:60000
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,i
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:60000/sample_form.php
Accept-Encoding: gzip, deflate, br
Accept-Language: ko,en;q=0.9,en-US;q=0.8

id=test&password=1234
```

Header in Packet

```
HTTP/1.1 200 OK
Server: nginx/1.4.6 (Ubuntu)
Date: Tue, 23 Mar 2021 05:05:38 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.5.9-1ubuntu4.29
Expires: Mon, 26 Jul 1997 05:00:00 GMT
Last-Modified: Tue, 23 Mar 2021 05:05:38 GMT
Cache-Control: no-store, no-cache, must-revalidate
Cache-Control: post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 43873

<!DOCTYPE html>
<html lang="ko">
<head>
<!-- META -->
<meta charset="utf-8">
<meta name="Generator" content="XpressEngine">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```


Request/Response: Content-Type

- 전송되는 데이터의 종류를 표현하는 헤더
- 요청/응답 양쪽에서 전송/활용할 수 있음
- 값은 MIME이라는 프로토콜을 따름

Request/Response: Content-Type - MIME

- Multipurpose Internet Mail Extensions
- 음악, 사진, 영상 등등의 데이터를 간결하게 표현하기 위한 형식
- 표준이므로 웬만한 브라우저 등에서 호환 가능
- <https://www.iana.org/assignments/media-types/media-types.xhtml>

실습: PHP Calculator

- 폼 POST 요청을 사용하여 간단한 계산기를 구현해보세요.
- 파일명은 calculator.php와 calculator_proc.php를 사용할 것
- 덧셈, 뺄셈, 나눗셈, 곱셈이 가능해야합니다!
- 기한: 다음주 활동시간까지
- 선택사항1: CSS 등을 사용해서 예쁘게 만들어보세요!
- 선택사항2: 굳이 calculator_proc.php를 사용할 필요가 있을까요?

질문?