

HTTP

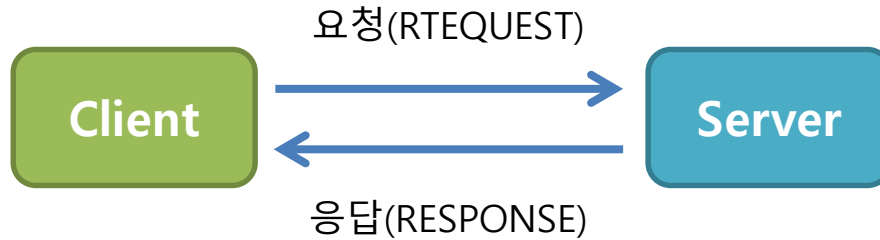
(HyperText Transfer Protocol)

1. 개요

- HyperText Transfer Protocol 또는 HyperText Protocol의 약자
HTTP는 서버와 클라이언트의 사이에서 어떻게 메시지(문서)를 교환할지를 정해놓은 규칙
- 서버와 클라이언트 간의 요청(Request)과 응답(Response)으로 구성
- HTTP/1.1 버전, HTTP/2 → TCP 기반의 통신

2. 구조

- 서버 클라이언트간의 요청과 응답으로 구성되어 클라이언트가 요청하면 응답하는 구조



- 클라이언트가 서버에 정보 요청을 하면 서버는 응답 코드와 내용을 전송하고, 클라이언트와의 연결을 종료

3. 요청 (Request)

- 요청 구조

```
POST /index HTTP/1.1
```

Start Line

```
Accept: application/json  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Content-Length: 83  
Content-Type: application/json  
Host: intropython.com  
User-Agent: HTTPie/0.9.3
```

Header

```
{  
    "uid": "1234567890",  
    "order_id": "order_id_8237352",  
    "status": "paid"  
}
```

Body

3. 요청 (Request)

- 요청 방식

- Start Line

- HTTP 요청의 시작줄
 - search 엔드포인트에 GET HTTP 요청을 보내면 해당 HTTP 요청의 start line은 아래와 같음

GET /index.html HTTP/1.1

- 구성 요소

- HTTP 메소드

- » HTTP 요청이 의도하는 액션을 정의하는 부분으로 GET, POST, PUT, DELETE 등

- Request target

- » 해당 HTTP 요청의 목표 주소, 위 요청에서는 요청의 request target은 /index.html

- HTTP version

- » 해당 요청의 HTTP 버전

- » 현재 1.0, 1.1, 2.0이 있음

- » 버전에 따라 요청 메세지 구조나 데이터가 다를 수 있기 때문에 버전을 명시

3. 요청 (Request)

- 요청 방식

- HTTP METHOD

- **GET**

- : 클라이언트가 서버에게 URL에 해당하는 자료의 전송을 요청한다.

- HEAD

- : GET 요청으로 반환될 데이터 중 헤더 부분에 해당하는 데이터만 요청한다.

- **POST**

- : 클라이언트가 서버에서 처리할 수 있는 자료를 body 영역에 포함시켜 보낸다.

- 예를 들어, 게시판에 글을 쓸 때 클라이언트의 문서가 서버로 전송되어야 한다.

- PATCH

- : 클라이언트가 서버에게 지정한 URL의 데이터를 부분적으로 수정할 것을 요청한다.

- **PUT**

- : 클라이언트가 서버에게 지정한 URL에 지정한 데이터를 저장할 것을 요청한다.

- **DELETE**

- : 클라이언트가 서버에게 지정한 URL의 정보를 제거할 것을 요청한다.

- TRACE

- : 클라이언트가 서버에게 송신한 요청의 내용을 반환해 줄 것을 요청한다.

- CONNECT

- : 클라이언트가 특정 종류의 프록시 서버에게 연결을 요청한다.

- OPTIONS

- : 해당 URL에서 지원하는 요청 메시지의 목록을 요청한다.

3. 요청 (Request)

- 요청 방식

- Header

- HTTP 요청 대한 정보를 담고 있음 : key와 value로 되어 있고, :로 연결됨

- 자주 사용되는 헤더

- Host

- 요청이 전송되는 target의 호스트 URL 주소 → Host: google.com

- User-Agent

- 요청을 보내는 클라이언트의 정보, 예를 들면 웹 브라우저 → User-Agent: Mozilla/5.0

- Accept

- 해당 요청이 받을 수 있는 response body 데이터 타입

- MIME(Multipurpose Internet Mail Extensions) type이 value로 지정

- » 예 : application/json, text/csv, text/html, image/jpen, application/xml

- Connection

- » 해당 요청이 끝난 후에 클라이언트와 서버가 계속 네트워크 연결을 유지할지, 끊을지

- » 매 요청마다 HTTP 연결을 새로 만드는 것은 시간이 걸리기 때문에 처음 만든 연결을 재사용하는 것을 선호 하기도 함

- Content-Type

- » 메시지 Body의 타입을 알려주며 Accept 헤더와 마찬가지로 MIME type 사용 → Content-Type: application/json

- Content-Length

- » 요청이 보내는 메시지 body의 총 사이즈 → Content-Length: 257

3. 요청 (Request)

- 요청 방식
 - Body
 - HTTP 요청이 전송하는 데이터를 담고 있는 부분
 - 전송하는 데이터가 없으면 Body는 비어있음

3. 응답 (Response)

- 응답 구조

```
HTTP/1.1 404 Not Found
```

Start Line

```
Connection: close  
Content-Length: 1573  
Content-Type: text/html; charset=UTF-8  
Date
```

Header

```
<DOCTYPE html>  
<html lang=ko>  
...
```

Body

3. 응답 (Response)

- 응답 구조
 - Status Line
 - HTTP 응답 메시지 상태를 간략히 요약해주는 부분
 - HTTP version, Status Code, Status Text로 나타남
 - Header
 - HTTP 요청의 헤더와 동일
 - 단, HTTP Response시에만 사용되는 헤더가 있음
 - HTTP Response엔 User-Agent 헤더 대신 Server 헤더가 사용
 - Body
 - HTTP 요청 메시지의 body와 동일
 - 전송하는 데이터가 없으면 body 부분은 비어있게 됨