

URI와 웹 브라우저 요청 흐름

- URI

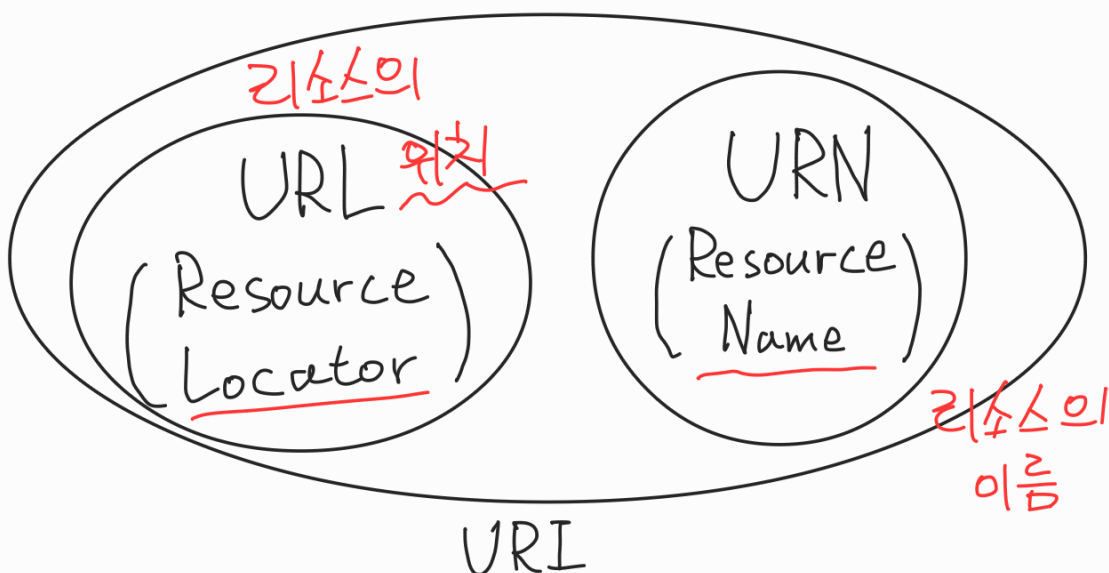
- 웹 브라우저 요청 흐름

URI (Uniform Resource Identifier)
(리소스를 통합하는 식별된 방법)

“URI는 locator, name

또는

둘 다 (both) 추가로 분류될 수 있다”



URL

foo://example.com:8042/over/there?name=ferret#nose
scheme authority path query fragment
urn:example:animal:ferret:nose

URI

URI

- Uniform : 리소스 식별하는 통일된 방식
- Resource : 자원 // URI로 식별 가능한 모든 것 (제한 X)
- Identifier : 다른 항목과 구분하는 데 필요한 정보

URL, URN

- URL : 리소스 있는 위치를 지정 가변
- URN : 리소스에 이름 부여 불변

ex) 어떤 책의 ISBN URN

⇒ urn:isbn:(number ~)

→ 실제 resource 찾는 방법이 표준화 X

URL 전체 문법

- $\text{scheme} : // [\text{userinfo}@] \text{host}[:\text{port}][/\text{path}][?\text{query}][\#\text{fragment}]$

protocol

host
name

port

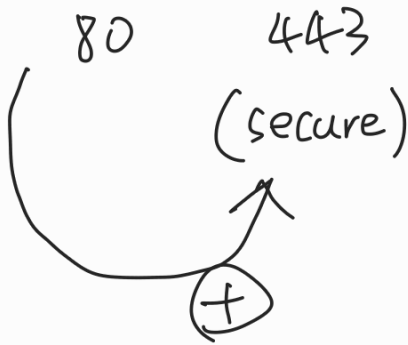
path

query
parameter

scheme

- 주로 protocol 사용

- ex) http, https, ftp, etc ...



userinfo

사용자 정보 포함하여 인증 / 거의 사용 X

host

호스트명 / domain name or IP address 사용 가능.

PORT

접속 포트

path

리소스 경로 / 계층적 구조

query

key = value 형태 / '?'로 시작, '&'로 추가

query parameter, query string 등으로 불림
(웹 서버에 제공해서) (다 문자라서)

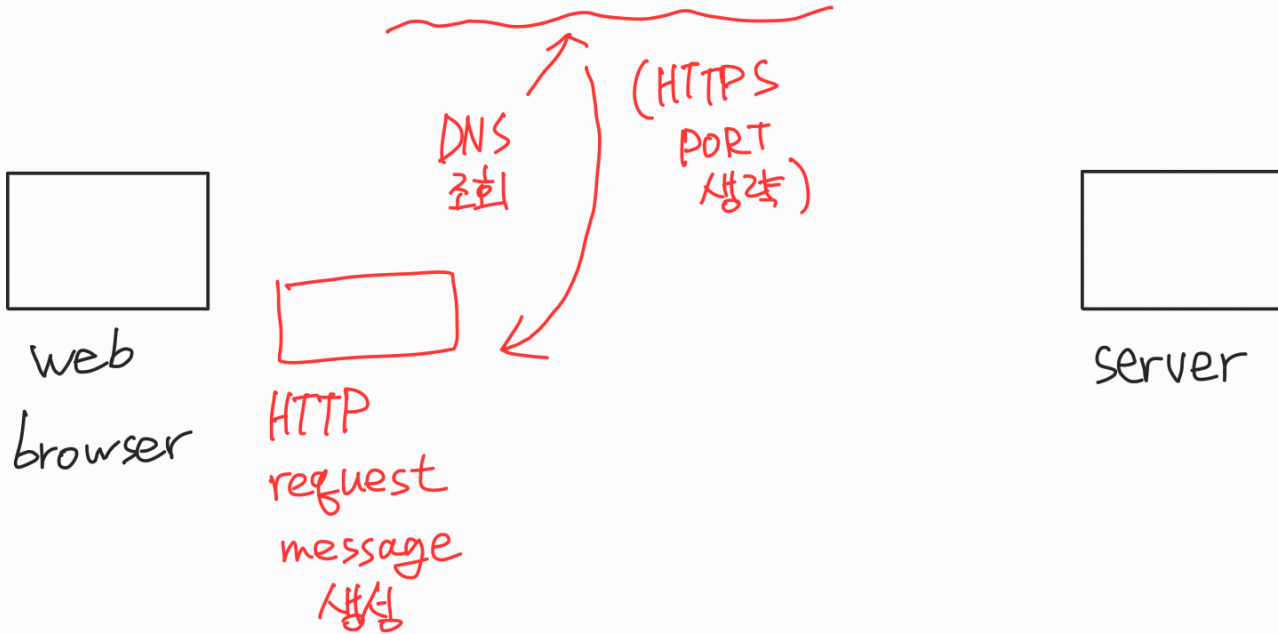
fragment

(# fragment)

- html 등 내부 북마크 등에 사용
- 서버에 전송하는 정보 X

웹 브라우저의 요청 흐름

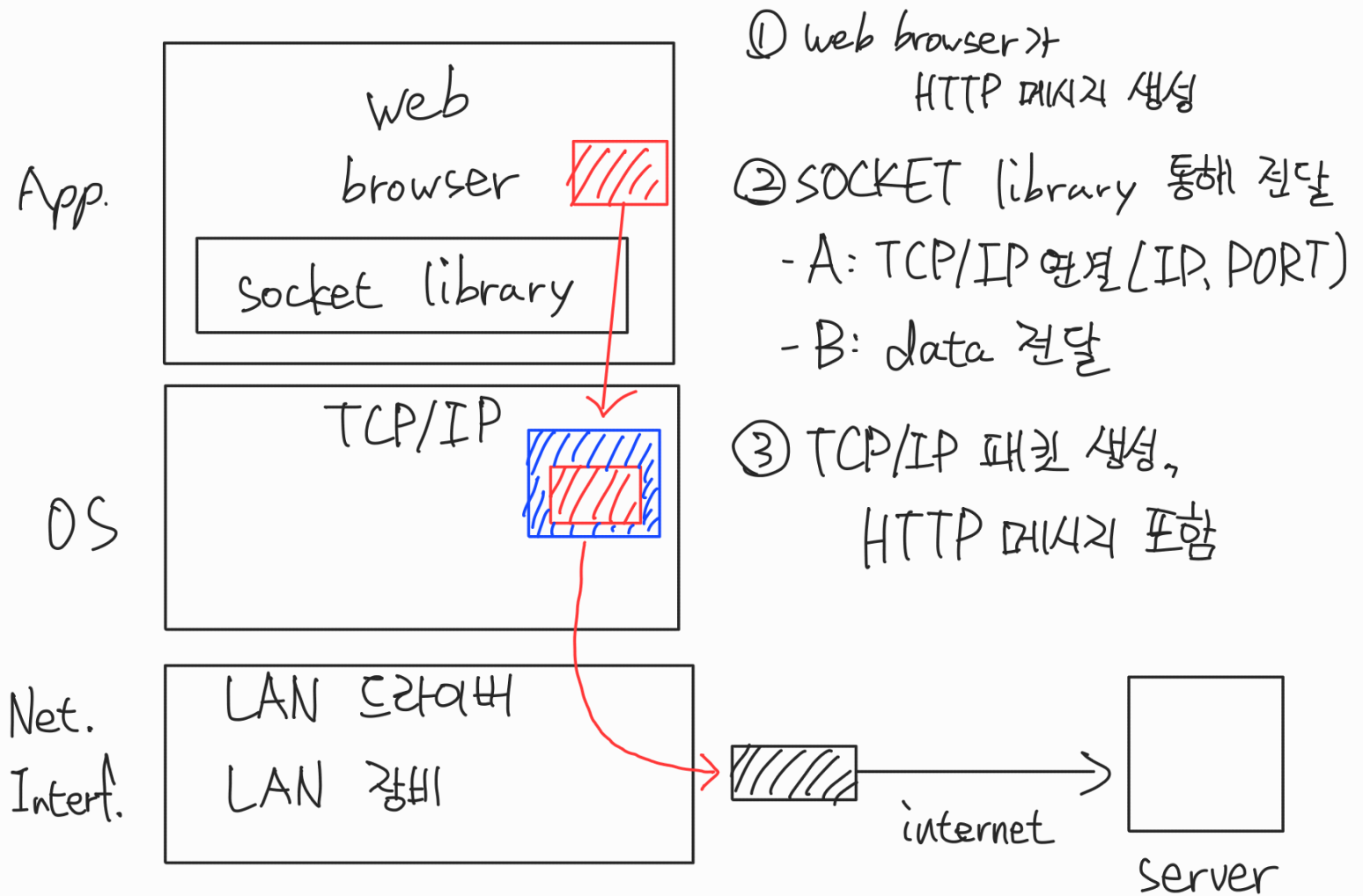
ex) ~ //www.google.com/search?q=hello&hl=ko



HTTP Request Message

ex) GET [] /search?~&~ [] HTTP/1.1
Host: ~.google.~

HTTP 메시지 전송



✱ TCP/IP 패킷은

받은 시에 버림

HTTP msg만!



HTTP Response Message

HTTP/1.1 200 OK

Content-Type: text/html; charset=UTF-8

Content-Length: ~~~~~

<html>

~~~~~

</~>