Lab 10 - HTTP

HTTP

- application level protocol, used for sending data over the Internet
- stateless protocol
- TCP port 80 or 443 for HTTPS

Request	Response

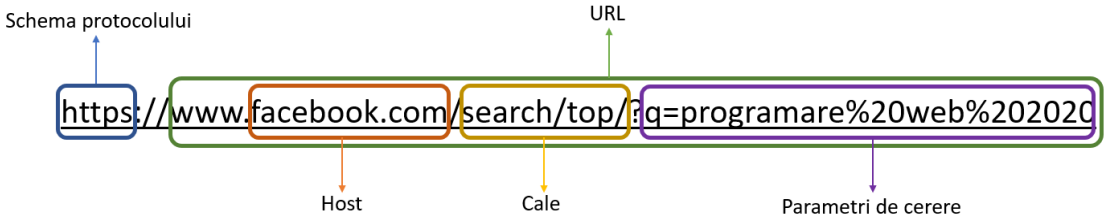
`METHOD PATH VERSION\r\n`
`Host: HOST\r\n`
`Header1: Value Header1\r\n`
`Header2: Value Header2\r\n`
`...`
`Cookie: key1=value1; key2=value2; ...;`
`keyN=valueN\r\n`
`\r\n`
`DATA`
	`VERSION STATUS_CODE TEXT_STATUS\r\n`
	`Header1: Value Header1\r\n`
	`Header2: Value Header2\r\n`
	`...`
	`HeaderN: Value HeaderN\r\n`
	`Set-Cookie: key1=value1\r\n`
	`Set-Cookie: key2=value2\r\n`
	`...`
	`Set-Cookie: keyN=valueN\r\n`
	`\r\n`
	`DATA`

Request		
Methods: • **GET** • **POST** • **PUT** • DELETE • HEAD • CONNECT • OPTIONS • TRACE • PATCH	Headers: • Host • Cookie • User-Agent • Connection • **Content-Type** • **Content-Length**	Content-Type: • text/plain • text/html • image/jpeg • audio/wav • video/ogg • multipart/form-data • application/x-www-form-urlencoded • application/json

X-WWW-FORM-URLENCODED ⇒ a=val1&b=val2
{a:val1, b:val2}

1xx - Information
2xx - Success
3xx - Redirect
4xx - Client errors
5xx - Server errors

Example




```
GET /search/top/?q=programare%20%web%202020 HTTPS\r\n
Host: facebook.com\r\n
User-Agent: Mozilla/5.0\r\n
Connection: keep-alive\r\n
Cookie: c_user=XXXXXXXXXX; presence=XXXXXXX\r\n          //// between cookies ';'
\r\n
```

ex. 3 example:

```
POST /api/v1/auth/login HTTP/1.1\r\n
Host: 34.241.4.235\r\n
Content-Length: 33\r\n
Content-Type: application/x-www-form-urlencoded\r\n
\r\n
username=student&password=student\r\n                    //// x-www-form-urlencoded
```

ex. 1:

GET /api/v1/dummy HTTP/1.1\r\n
Host: 34.241.4.235\r\n
\r\n

ex. 2:
POST /api/v1/dummy HTTP/1.1\r\n
Host: 34.241.4.235\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Content-Length: 10\r\n
\r\n
test&dummy   //// x-www-form-urlencoded

ex. 4 set cookies:

strcpy(cookies[0],
"connect.sid=s%3Aoy5V6CLDYPGv8SqYq6yzvXL-BpCKiCC1.BEK0tJsEWKyuecGp7ADt%2FNheOkMF7Ov3EcngkQmgHnE");
GET:
…
Cookie: ..

ex.5 URL parameters:
to find ip address of api.openweathermap.org , ping the server and get ip address

query_params = “lat=44.7398&lon=22.2767&appid=b912dd495585fbf756dc6d8f415a7649”
ex. 6 hint:
→ use **char *basic_extract_json_response(char *str)** from helpers.c to jsonify the response from the server

message = compute_post_request("34.241.4.235", "/api/v1/weather/44.7398/22.2767", "application/json", json_data, 1, cookies, 1);

Code Utils

```
sockfd = open_connection("34.241.4.235", 8080, AF_INET, SOCK_STREAM, 0);
message = compute_get_request("34.241.4.235", "/api/v1/dummy", NULL, NULL, 0);
puts(message);
send_to_server(sockfd, message);
response = receive_from_server(sockfd);
puts(response);
close_conection(sockfd);
```

```
$ telnet 34.241.4.235 8080
```

Link Utils

Lab OCW
RFC HTTP 1.1
Headere HTTP
Tipuri MIME
Stateless vs stateful