

## Laboratorul 3. Programarea în rețea

### HTTP Client

Să se creeze o aplicație client HTTP

#### Pentru nota 7-8:

- Cererile HTTP să fie făcute prin intermediul unui proxy
- Să se utilizeze expresiile regulate

#### Pentru nota 9 și 10:

- Să se utilizeze firele de execuții și tehnici de sincronizare
- Clientul trebuie să se poată autentifica pe resursă utilizând cookies

#### Atenție:

- Pentru acest laborator utilizați librării HTTP deja existente, nu este necesar de a utiliza Socket API. Cine dorește poate să facă prin socket ca și la primul laborator.
- Clientul trebuie să facă cereri GET, POST, HEAD și OPTIONS
- Aplicația poate fi consolă sau GUI
- Nu sunteți limitați la funcțional, resursa(**pagina web**) la care clientul o să facă cereri HTTP este la alegere.
- Vă recomand să folosiți proxy private și nu free: <https://proxy-seller.com>
- Aplicația elaborată trebuie să posede o logică bine definită

#### Întrebări la apărarea laboratorului:

- Cum este formatat corpul unei cereri HTTP pentru o cerere HTTP de tip POST ?
- De unde știe un client HTTP ce tip de conținut trimite serverul HTTP ?
- Cum decide un client dacă ar trebui să aibă încredere în certificatul unui server ?
- Care este problema principală cu certificatele autosemnate ?
- Conexiunea persistentă HTTP – care sunt principalele beneficii ?
- Ce este negocierea conținutului în HTTP și cum are loc ?
- Care sunt tipurile de negociere a conținutului HTTP ?
- Ce este un ETag în HTTP și cum funcționează ?
- Diferența dintre protocoalele fără stare și cele cu stare. Căru tip îi aparține HTTP ?
- Avantajele cheie ale HTTP/2 în comparație cu HTTP/1.1
- Ce este un tip MIME, din ce constă și pentru ce se folosește ?
- Care este diferența dintre GET și POST ?
- Care este diferența dintre PUT și POST ?
- Care sunt metodele idempotente în HTTP și care sunt scopul lor.
- Cum sunt identificate resursele în protocolul HTTP ?
- Care sunt metodele sigure și nesigure în HTTP ?
- Pentru ce este nevoie de cURL ?

- Pentru ce este nevoie de HTTP Proxy?
- Diferența dintre autentificare și autorizare
- Care sunt metodele de autentificare HTTP ?
- Modalități de identificare a utilizatorilor în HTTP
- HTTP cookies – pentru ce se folosește ?

### Link-uri utile:

- <https://ec.haxx.se/http/http-post>
- <https://stackoverflow.com/questions/23714383/what-are-all-the-possible-values-for-http-content-type-header>
- <https://webmasters.stackexchange.com/questions/31212/difference-between-the-accept-and-content-type-http-headers>
- <https://www.globalsign.com/en/ssl-information-center/what-are-certification-authorities-trust-hierarchies>
- <https://www.sslshopper.com/article-when-are-self-signed-certificates-acceptable.html>
- <https://www.globalsign.com/en/ssl-information-center/dangers-self-signed-certificates>
- [https://developer.mozilla.org/en-US/docs/Web/HTTP/Connection\\_management\\_in\\_HTTP\\_1.x](https://developer.mozilla.org/en-US/docs/Web/HTTP/Connection_management_in_HTTP_1.x)
- [https://developer.mozilla.org/en-US/docs/Web/HTTP/Content\\_negotiation](https://developer.mozilla.org/en-US/docs/Web/HTTP/Content_negotiation)
- <https://www.logicbig.com/quick-info/web/etag-header.html>
- <https://www.quora.com/Why-was-HTTP-stateless-built-on-top-of-a-stateful-protocol-TCP>
- <https://stackoverflow.com/questions/5836881/stateless-protocol-and-stateful-protocol>
- <https://www.digitalocean.com/community/tutorials/http-1-1-vs-http-2-what-s-the-difference>
- [https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics\\_of\\_HTTP/MIME\\_types](https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types)
- <https://www.edureka.co/blog/get-and-post-method/>
- <https://www.keycdn.com/support/put-vs-post>
- <https://javarevisited.blogspot.com/2016/05/what-are-idempotent-and-safe-methods-of-HTTP-and-REST.html>
- <https://curl.haxx.se/docs/httpscripting.html>
- <https://stackoverflow.com/questions/7155529/how-does-http-proxy-work>
- <https://blog.risingstack.com/web-authentication-methods-explained/>
- <https://humanwhocodes.com/blog/2009/05/05/http-cookies-explained/>