

HTTP(S)

Inleiding

- Client-Server protocol
- Transportprotocol: TCP op poort 80
- Aanvraag-antwoord protocol (request-response)

Structuur bericht

Eerste headerlijn

- Verschillend voor:
 - HTTP-aanvraag
 - Aanvraagmethode
 - GET -> enkel headerlijnen, geen corpus
 - HEAD -> vraag naar info, antwoord enkel headerlijnen
 - PUT -> bestand op server plaatsen, weinig ondersteund
 - POST -> gegevens aanpassen op server, vaak gebruikt met HTML-formulieren
 - Aanvraag-URI
 - Absolute URI -> vooral aanvragen aan proxy servers
 - Absoluut pad -> geen protocol en geen servernaam, naam van de server in een aparte Host-headers
 - HTTP-versie
 - HTTP-antwoord
 - Statuslijn
 - 1??: Informatief
 - 2??: Succes
 - 3??: Omleiding
 - 4??: Fout bij client
 - 5??: Fout bij server
 - Veel gebruikte:
 - 200 OK
 - 301 Moved Permanently
 - 302 Found / Moved Temporarily
 - 303 See Other
 - 401 Unauthorized
 - 403 Forbidden
 - 404 Not Found
 - 501 Not Implemented

Volgende headerlijnen

- Algemene vorm
naam: waarde CRLF
- Waarde over verschillende lijnen
 - Nieuwe lijn start met spatie of TAB
- Naam: geen onderscheid tussen hoofdletters en kleine letters

Lege lijn

Corpus (optioneel)

- Geen interne structuur
- Inhoud vaak gecodeerd met behulp van MIME
- Begin: blanco-lijn
- Einde:
 - Content-Length-header: lengte in bytes
 - Chunked transfer coding

Chunked transfer coding

- Headerlijn
 - Transfer-Encoding: chunked
- Corpus verdeeld in verschillende stukken (chunks)
- Chunk
 - Lengte chunk in bytes (hexadecimaal)
 - CRLF
 - Inhoud
- Laatste chunk heeft lengte 0, daarna volgt een lege lijn

Corpus bij formulier

```
<form action="toonWaarden" method="post">  
  Geef naam: <input type="text" name="naam"><br>  
  Geef adres: <input type="text" name="adreslijn"><br>  
  <input type="submit" value="Verstuur">  
</form>
```

Geef naam:

Geef adres:

Zal er in de corpus uitzien als "naam=Veerle+D%27Haese&adres=Kerkstraat+10"

Naam/Waarde-paren

- Codering
 - Speciale tekens: % + 2 hexadecimale cijfers (ASCII-code)
 - Spatie: bovenstaande methode of +

Cookies

- Algemeen
 - Klein stukje tekst
 - Bewaard door de client
 - Teruggestuurd bij elke aanvraag
 - In HTTP-headers
- Bij eerste aanvraag
 - Server stuurt cookie naar de client
- Bij volgende aanvraag
 - Browser voegt extra header toe

Wachtwoorden

- Twee wachtwoordenschema's
 - Basic: niet veilig, wachtwoorden niet gecodeerd, door bijna alle servers en browsers ondersteund
 - Digest: zie cursus beveiliging
- Principe
 - Client: aanvraag URI
 - Server: antwoord 401 Unauthorized
 - Client vraagt gebruikersnaam en paswoord aan gebruiker (indien eerste keer!)
 - Client: nieuwe aanvraag zelfde URI met wachtwoordgegevens
 - Gebruikersnaam en wachtwoord: samengevoegd en omgezet in base64 formaat
 - Client kan bij volgende aanvraag onmiddellijk Authorization-header toevoegen

Persistente verbindingen

- HTTP/1.1: meerdere aanvragen over dezelfde connectie
 - Bij elke aanvraag hoort een afzonderlijk antwoord
 - Verbinding wordt niet telkens afgesloten
 - Client kan verschillende aanvragen na elkaar sturen zonder op antwoord te wachten (pipelining)
- Connection-header
 - Connection: close
 - In aanvraag of antwoord

HTTPS

- HyperText Transfer Protocol Secure
- Maakt gebruik van SSL (Secure Sockets Layer)
- SSL:
 - Verzoek om een beveiligde verbinding via SSL te initialiseren
 - Weergave en verificatie van het certificaat
 - Geldigheid
 - Controle via een betrouwbare externe partij
 - Overdracht via een unieke encryptie key
 - Het decoderen van een encryptie key door de server, met het gebruik van een private key
 - Een beveiligde verbinding opzetten