

### Zadanie 3:

```
if ($r->method eq 'GET') {  
    my $headers = $r->headers_as_string;  
    $c->send_response(HTTP::Response->new(200, 'OK', ['Content-Type' =>  
'text/plain'], $headers));  
}
```

Zamiast

```
if ($r->method eq 'GET') {  
  
    $file_s = "./index.html";  
    $c->send_file_response($file_s);  
  
}
```

`my $headers = $r->headers_as_string;;`

Ta linia kodu pobiera nagłówki żądania HTTP jako jeden ciąg znaków. \$r to obiekt żądania, który reprezentuje żądanie HTTP przychodzące od klienta. Metoda headers\_as\_strin konwertuje wszystkie nagłówki żądania na tekst, umożliwiając łatwe ich wyświetlenie lub przetworzenie.

`$c->send_response(HTTP::Response->new(200, 'OK', ['Content-Type' => 'text/plain'], $headers));;`

Ta linia odpowiada za wysłanie odpowiedzi do klienta. Używa metody send\_response na obiekcie połączenia \$c, który reprezentuje połączenie z klientem.

HTTP::Response->new(200, 'OK', ['Content-Type' => 'text/plain'], \$headers) tworzy nowy obiekt odpowiedzi HTTP, gdzie:

200 to kod statusu HTTP, który oznacza, że żądanie zostało pomyślnie przetworzone.

'OK' to standardowy komunikat towarzyszący kodowi statusu 200.

['Content-Type' => 'text/plain'] to nagłówek odpowiedzi, który mówi, że typ zawartości odpowiedzi to zwykły tekst (text/plain).

\$headers to ciało odpowiedzi, które zawiera tekst wszystkich nagłówków żądania, jakie zostały przesłane przez klienta.

```
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: ru-PL,ru;q=0.9,en-US;q=0.8,en;q=0.7,ru-RU;q=0.6,pl;q=0.5
Host: 127.0.0.1:4321
If-Modified-Since: Sun, 09 Jun 2024 14:40:51 GMT
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.36
Sec-Ch-Ua: "Google Chrome";v="125", "Chromium";v="125", "Not.A/Brand";v="24"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
```

**Connection: keep-alive** - Ten nagłówek informuje, że połączenie z serwerem powinno pozostać aktywne po obsłużeniu tego żądania, aby można było wykonać kolejne żądania bez konieczności ponownego nawiązywania połączenia TCP, co zwiększa wydajność.

- **Accept** - Określa typy treści (MIME types), które klient może prawidłowo obsłużyć. Lista ta jest priorytetyzowana:  
text/html,application/xhtml+xml,application/xml;q=0.9 - Klient preferuje HTML i XHTML, ale zaakceptuje również XML z niższym priorytetem (q=0.9).  
image/avif,image/webp,image/apng,\*/\*;q=0.8 - Obsługa nowoczesnych formatów obrazów oraz dowolnych innych typów mediów, ale z niższym priorytetem (q=0.8).  
application/signed-exchange;v=b3;q=0.7 - Zaakceptowanie odpowiedzi w formacie Signed HTTP Exchange z jeszcze niższym priorytetem.
- **Accept-Encoding** - Wylicza metody kompresji danych, które klient jest w stanie obsłużyć: gzip, deflate, br, zstd - Obsługa różnych algorytmów kompresji, co pozwala na szybsze ładowanie treści.
- **Accept-Language** - Preferencje językowe klienta, podane w kolejności preferencji: ru-PL,ru;q=0.9,en-US;q=0.8,en;q=0.7,ru-RU;q=0.6,pl;q=0.5 - Preferencje wskazują na priorytet języka rosyjskiego, angielskiego, oraz polskiego z określonymi wagami (q).
- **Host** - Adres IP lub nazwa domeny serwera oraz numer portu, do którego klient wysyła żądanie. Tutaj 127.0.0.1:4321 oznacza lokalny adres serwera na porcie 4321.
- **If-Modified-Since** - Ten nagłówek jest używany, aby zapytać serwer, czy zasób został zmodyfikowany od podanej daty: Sun, 09 Jun 2024 14:40:51 GMT - Jeśli zasób nie był modyfikowany od tej daty, serwer może odpowiedzieć kodem 304 (Not Modified), co pozwala na oszczędność transferu.
- **User-Agent** - Informacje o przeglądarce i systemie operacyjnym klienta, które mogą być wykorzystywane przez serwer do dostosowania odpowiedzi:  
Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/125.0.0.0 Safari/537.36 - Wskazuje na przeglądarkę Chrome na Linuxie.
- **Sec-Ch-Ua, Sec-Ch-Ua-Mobile, Sec-Ch-Ua-Platform** - Te nagłówki są częścią nowego standardu Client Hints, które dostarczają dodatkowe informacje o klientach w bezpieczny sposób:

"Google Chrome";v="125", "Chromium";v="125", "Not.A/Brand";v="24" - Wersje klienta.

?0 w przypadku Sec-Ch-Ua-Mobile oznacza, że żądanie pochodzi z urządzenia stacjonarnego.

"Linux" jako platforma systemowa w Sec-Ch-Ua-Platform.

- **Sec-Fetch-Dest, Sec-Fetch-Mode, Sec-Fetch-Site, Sec-Fetch-User** - Te nagłówki dostarczają informacje związane z modelem żądania: document, navigate wskazują na typ i sposób żądania. none, ?1 wskazują na brak site i preferencję użytkownika do żądania strony.
- **Upgrade-Insecure-Requests: 1** - Żądanie od klienta, aby serwer preferował bezpieczne połączenia (HTTPS) nawet jeśli początkowe żądanie było wykonane przez HTTP.

#### Zadanie 4 i 5:

#### Traffic on usual interface:

#### Zapytanie HTTP GET:

```
Window: 502
[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0xe551 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
- Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  - TCP Option - No-Operation (NOP)
    Kind: No-Operation (1)
  - TCP Option - No-Operation (NOP)
    Kind: No-Operation (1)
  - TCP Option - Timestamps: TSval 4035976118, TSecr 517653164
    Kind: Time Stamp Option (8)
    Length: 10
    Timestamp value: 4035976118
    Timestamp echo reply: 517653164
- [Timestamps]
  [Time since first frame in this TCP stream: 0.023018440 seconds]
  [Time since previous frame in this TCP stream: 0.000253288 seconds]
- [SEQ/ACK analysis]
  [iRTT: 0.022765152 seconds]
  [Bytes in flight: 448]
  [Bytes sent since last PSH flag: 448]
  TCP payload (448 bytes)
- Hypertext Transfer Protocol
  - GET /gen204?nca=te_li&client=te_lib&logId=vTE_20240610 HTTP/1.1\r\n
    - [Expert Info (Chat/Sequence): GET /gen204?nca=te_li&client=te_lib&logId=vTE_20240610 HTTP/1.1\r\n]
      [GET /gen204?nca=te_li&client=te_lib&logId=vTE_20240610 HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
      Request Method: GET
    - Request URI: /gen204?nca=te_li&client=te_lib&logId=vTE_20240610
      Request URI Path: /gen204
    - Request URI Query: nca=te_li&client=te_lib&logId=vTE_20240610
      Request URI Query Parameter: nca=te_li
      Request URI Query Parameter: client=te_lib
      Request URI Query Parameter: logId=vTE_20240610
      Request Version: HTTP/1.1
    Host: translate.google.com\r\n
    Connection: keep-alive\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.36\r\n
    Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8\r\n
    Referer: http://127.0.0.1:4321/\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: ru-PL,ru;q=0.9,en-US;q=0.8,en;q=0.7,ru-RU;q=0.6,pl;q=0.5\r\n
    \r\n
    [Full request URI: http://translate.google.com/gen204?nca=te_li&client=te_lib&logId=vTE_20240610]
    HTTP request 1/1
```

**Request Method:** GET - typowa metoda używana do pobierania danych z serwera.

**Request URI:** /gen204?nca=te\_li&client=te\_lib&logId=vTE\_20240610 - URI używane przez Google do zbierania danych statystycznych lub monitorowania.

**Path:** /gen204 - endpoint na serwerze Google.

## Query Parameters:

- nca=te\_li
- client=te\_lib
- logld=vTE\_20240610 - parametry w zapytaniu, które mogą dotyczyć funkcji telemetrii lub logowania danych.

**Request Version:** HTTP/1.1 - wersja protokołu HTTP używana do transmisji.

**Referer:** http://127.0.0.1:4321/: URL, który wywołał żądanie. Może być używany do śledzenia, skąd pochodzą żądania.

## Odpowiedź:

```
Destination Port: 41936
[Stream index: 7]
[Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 1302]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 1936858884
[Next Sequence Number: 1303 (relative sequence number)]
Acknowledgment Number: 449 (relative ack number)
Acknowledgment number (raw): 2485656434
1000 ... = Header Length: 32 bytes (8)
+ Flags: 0x010 (PSH, ACK)
Window: 261
[Calculated window size: 66810]
[Window size scaling factor: 256]
Checksum: 0x000 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
+ Options: [12 bytes], No-Operation (NOP), No-Operation (NOP), Timestamps
+ [Timestamps]
+ [SEQ/ACK analysis]
+ TCP payload (1302 bytes)
+ Hypertext Transfer Protocol
- HTTP/1.1 204 No Content\r\n
+ [Expert Info (Chat/Sequence): HTTP/1.1 204 No Content\r\n]
Response Version: HTTP/1.1
Status Code: 204
[Status Code Description: No Content]
Response Phrase: No Content
Content-Type: image/gif; charset=utf-8\r\n
Cache-Control: no-cache, no-store, max-age=0, must-revalidate\r\n
Pragma: no-cache\r\n
Expires: Mon, 01 Jan 1990 00:00:00 GMT\r\n
Date: Wed, 12 Jun 2024 17:12:56 GMT\r\n
Cross-Origin-Resource-Policy: cross-origin\r\n
Accept-CH: Sec-CH-UA-Arch, Sec-CH-UA-Bitness, Sec-CH-UA-Full-Version, Sec-CH-UA-Full-Version-List, Sec-CH-UA-Model, Sec-CH-UA-WoW64, Sec-CH-UA-Form-Factor, Sec-CH-UA-Platform, Sec-CH-UA-Platform-Version\r\n
Permissions-Policy: ch-ua-arch=, ch-ua-bitness=, ch-ua-full-version=, ch-ua-full-version-list=, ch-ua-model=, ch-ua-wow64=, ch-ua-form-factor=, ch-ua-platform=, ch-ua-platform-version=\r\n
Cross-Origin-Opener-Policy: same-origin\r\n
Content-Security-Policy: script-src 'report-sample' 'nonce-Po8tKfJm8kYNLYE2KD4fMw' 'unsafe-inline';object-src 'none';base-uri 'self';report-uri //TranslateApiHttp/cspreport;worker-src 'self'\r\n
Content-Security-Policy: require-trusted-types-for 'script';report-uri //TranslateApiHttp/cspreport\r\n
reporting-endpoints: default="//TranslateApiHttp/web-reports?context=e3zjktD1kmJw0gD19BmsAUC8J01164HE16xCPByz113fzCbwoWf-0kYlpaT8wviS0sS84p2EktT11KKyIKJ4IwMJEwMzQwM9A_P4AgMALLcyZA"\r\n
Server: ESF\r\n
- Content-Length: 0\r\n
[Content length: 0]
X-XSS-Protection: 0\r\n
X-Frame-Options: SAMEORIGIN\r\n
X-Content-Type-Options: nosniff\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.045381345 seconds]
[Request in frame: 336]
[Request URI: http://translate.google.com/gen204?nca=te_li&client=te_lib&logld=vTE_20240610]
```

- **IPv4:**
  - **Adres źródłowy:** 142.251.143.78 (zewnętrzny serwer Google)
  - **Adres docelowy:** 192.168.0.106 (lokalny adres IP)
  - Pakiet ten pochodzi z zewnętrznego adresu IP, co wskazuje, że odpowiedź nie pochodzi z lokalnie uruchomionego serwera, a z serwera Google.
- **TCP:**
  - **Port źródłowy:** 80 (standardowy port HTTP)
  - **Port docelowy:** 41936 (port na lokalnym komputerze)
  - Ustawienia flag TCP (PSH, ACK) oraz sequence i acknowledgment numbers wskazują na to, jak dane są synchronizowane i potwierdzane między klientem a serwerem.
- **HTTP/1.1 204 No Content**
  - Odpowiedź z kodem 204, co oznacza, że serwer przetworzył żądanie pomyślnie, ale nie zwraca żadnej treści.

- Występowanie odpowiedzi 204 jest typowe dla żądań, które nie wymagają, aby serwer zwracał jakiekolwiek dane, na przykład po wykonaniu operacji, która nie generuje danych do wyświetlenia.
- **Content-Type:** image/gif; charset=utf-8 - mimo że serwer twierdzi, że nie zwróci treści, nadal określa typ zawartości, który byłby zwrócony.
- **Cache-Control, Pragma, Expires:** Instrukcje dla przeglądarek, aby nie przechowywać odpowiedzi w pamięci podręcznej.
- **Cross-Origin-Resource-Policy, Cross-Origin-Opener-Policy:** Zabezpieczenia związane z zasadami samego pochodzenia, które kontrolują, jak zasoby z jednego źródła mogą być wykorzystywane przez dokumenty z innego źródła.
- **Content-Security-Policy:** Polityka bezpieczeństwa określająca, które skrypty mogą być uruchomione.
- **Server:** Nazwa serwera, co wskazuje, że odpowiedź pochodzi z serwera Google.
- **X-XSS-Protection, X-Frame-Options, X-Content-Type-Options:** Nagłówki związane z zabezpieczeniami, które pomagają w ochronie przed różnymi atakami internetowymi.

Analiza ramki nr 358 ukazuje, jak odpowiedzi HTTP i powiązane z nimi nagłówki przyczyniają się do zabezpieczeń, zarządzania pamięcią podręczną oraz ogólnej komunikacji sieciowej między serwerem a klientem.

### Traffic on the loopback interface:

No.	Time	Source	Destination	Protocol	Length	Info
9	3.369775868	127.0.0.1	127.0.0.1	HTTP	760	GET / HTTP/1.1
23	3.371749400	127.0.0.1	127.0.0.1	HTTP	182	HTTP/1.1 200 OK (text/html)
28	4.711337848	127.0.0.1	127.0.0.1	HTTP	811	GET /second.html HTTP/1.1
39	4.712539498	127.0.0.1	127.0.0.1	HTTP	181	HTTP/1.1 200 OK (text/html)
43	5.196285380	127.0.0.1	127.0.0.1	HTTP	821	GET /third.html HTTP/1.1
54	5.197457131	127.0.0.1	127.0.0.1	HTTP	182	HTTP/1.1 200 OK (text/html)
56	5.639233622	127.0.0.1	127.0.0.1	HTTP	821	GET /second.html HTTP/1.1
67	5.640542754	127.0.0.1	127.0.0.1	HTTP	181	HTTP/1.1 200 OK (text/html)
69	6.060185357	127.0.0.1	127.0.0.1	HTTP	821	GET /third.html HTTP/1.1
80	6.061641182	127.0.0.1	127.0.0.1	HTTP	182	HTTP/1.1 200 OK (text/html)
82	6.489790473	127.0.0.1	127.0.0.1	HTTP	821	GET /second.html HTTP/1.1
93	6.491256139	127.0.0.1	127.0.0.1	HTTP	181	HTTP/1.1 200 OK (text/html)
95	6.925395806	127.0.0.1	127.0.0.1	HTTP	821	GET /third.html HTTP/1.1
106	6.926527134	127.0.0.1	127.0.0.1	HTTP	182	HTTP/1.1 200 OK (text/html)

Serwer obsługuje wiele żądań GET dla różnych plików (/first.html, /second.html, /third.html).

Dla każdego żądania serwer pomyślnie znajduje i zwraca żądany plik z odpowiedzią 200 OK oraz typem zawartości text/html.

Cały ruch do ruchu HTTP przez interfejs pętli zwrotnej (loopback) (127.0.0.1 do 127.0.0.1), co jest zgodne z serwerem działającym lokalnie i obsługującym żądania lokalne.

Domyślny plik zwracany dla żądań do / to first.html, zgodnie z konfiguracją serwera.

Serwer prawidłowo odczytuje i serwuje pliki z katalogu ./www.

Wszystkie odpowiedzi to 200 OK, co wskazuje, że wszystkie żądane pliki istnieją i są serwowane bez żadnych problemów

### Zapytanie od second.html:

```

> Frame 28: 811 bytes on wire (6488 bits), 811 bytes captured (6488 bits) on interface lo, id 0
> Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 51546, Dst Port: 4301, Seq: 695, Ack: 267, Len: 745
< Hypertext Transfer Protocol
  < GET /second.html HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /second.html HTTP/1.1\r\n]
    [GET /second.html HTTP/1.1\r\n]
    [Severity level: Chat]
    [Group: Sequence]
    Request Method: GET
    Request URI: /second.html
    Request Version: HTTP/1.1
    Host: 127.0.0.1:4301\r\n
    Connection: keep-alive\r\n
    sec-ch-ua: "Google Chrome";v="125", "Chromium";v="125", "Not.A/Brand";v="24"\r\n
    sec-ch-ua-mobile: ?0\r\n
    sec-ch-ua-platform: "Linux"\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.0\r\n
    Sec-Fetch-Site: same-origin\r\n
    Sec-Fetch-Mode: navigate\r\n
    Sec-Fetch-User: ?1\r\n
    Sec-Fetch-Dest: document\r\n
    Referer: http://127.0.0.1:4301/\r\n
    Accept-Encoding: gzip, deflate, br, zstd\r\n
    Accept-Language: ru-PL,ru;q=0.9,en-US;q=0.8,en;q=0.7,ru-RU;q=0.6,pl;q=0.5\r\n
    \r\n
    [Full request URI: http://127.0.0.1:4301/second.html]
    [HTTP request 2/7]
    [Prev request in frame: 9]
    [Response in frame: 39]
    [Next request in frame: 43]
```

Zapytanie pochodzi od klienta działającego na adresie 127.0.0.1 i jest skierowane do serwera nasłuchującego na porcie 4301 tego samego adresu. Jest to typowe dla środowiska lokalnego, w którym serwer i klient działają na tej samej maszynie, wykorzystując interfejs loopback.

Linia żądania składa się z trzech części: metody żądania, która w tym przypadku jest GET, co oznacza, że klient chce pobrać zasób; URI żądania, które jest /second.html, wskazując na konkretny plik na serwerze; oraz wersji HTTP, która wynosi HTTP/1.1, określając używany standard protokołu.

User-Agent identyfikuje oprogramowanie klienta, dostarczając serwerowi informacji o przeglądarce i systemie operacyjnym, co może być użyteczne dla serwera do optymalizacji treści. Accept określa typy mediów, które klient jest w stanie zaakceptować w odpowiedzi, w tym różne typy dokumentów HTML, obrazów i inne. Referer wskazuje adres URL strony, z której pochodzi żądanie, co może być użyteczne do śledzenia nawigacji użytkownika.

Nagłówek Accept-Encoding informuje serwer o tym, jakie schematy kompresji klient może obsługiwać, co może pomóc w zmniejszeniu rozmiaru odpowiedzi. Accept-Language wskazuje preferencje językowe klienta, co może być użyte do dostosowania treści odpowiedzi do preferowanego języka.

## Odpowiedź:

```
Frame 39: 181 bytes on wire (1448 bits), 181 bytes captured (1448 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 4301, Dst Port: 51546, Seq: 417, Ack: 1440, Len: 115
[6 Reassembled TCP Segments (265 bytes): #29(17), #31(37), #33(33), #35(61), #37(2), #39(115)]
```

### Hypertext Transfer Protocol

```
HTTP/1.1 200 OK\r\n
```

```
  [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
```

```
    [HTTP/1.1 200 OK\r\n]
```

```
    [Severity level: Chat]
```

```
    [Group: Sequence]
```

```
    Response Version: HTTP/1.1
```

```
    Status Code: 200
```

```
    [Status Code Description: OK]
```

```
    Response Phrase: OK
```

```
    Date: Wed, 12 Jun 2024 22:40:24 GMT\r\n
```

```
    Server: libwww-perl-daemon/6.16\r\n
```

```
Content-Length: 115\r\n
```

```
  [Content length: 115]
```

```
Content-Type: text/html; charset=utf-8\r\n
```

```
\r\n
```

```
  [HTTP response 2/7]
```

```
  [Time since request: 0.001201650 seconds]
```

```
  [Prev request in frame: 9]
```

```
  [Prev response in frame: 23]
```

```
  [Request in frame: 28]
```

```
  [Next request in frame: 43]
```

```
  [Next response in frame: 54]
```

```
  [Request URI: http://127.0.0.1:4301/second.html]
```

```
File Data: 115 bytes
```

```
Line-based text data: text/html (8 lines)
```

```
<html>\n
```

```
<body>\n
```

```
  <h1>SECOND</h1>\n
```

```
  <a href="first.html">first</a>\n
```

```
  <a href="third.html">third</a>\n
```

```
</body>\n
```

```
</html>\n
```

```
\n
```

Odpowiedź zaczyna się od linii statusu: HTTP/1.1 200 OK. Wersja protokołu (HTTP/1.1) informuje o tym, że odpowiedź korzysta z wersji 1.1 protokołu HTTP, co jest standardem w dzisiejszym internecie. Kod statusu 200 oraz fraza statusu OK wskazują, że żądanie zostało pomyślnie przetworzone przez serwer i że żądany zasób (second.html) został znaleziony i jest zwracany do klienta.

Treść odpowiedzi zawiera kod HTML strony second.html, który zostanie wyświetlony przez przeglądarkę.