Desenvolvimento de Componentes Distribuídos - RESTFull

Karen Christina Diz – 29/03/21

Descrição da Tarefa:

Em relação ao HTTP descreva e entenda os seguintes conceitos:

1. **Qual a diferença entre HTTP e HTTPS?**

Os dois protocolos são praticamente idênticos o que diferencia é que no HTTPS foi adicionado uma camada de 'S' de segurança, essa camada tem como princípios a confidencialidade, integridade e autenticação. E para isso ela utiliza alguns mecanismos como:

- SSL (Secure Sockets Layer) ou de TLS (Transfer Layer Security) que é a versão mais recente: Camadas de segurança que garantem a integridade e confidencialidade.
- Criptografia: Par de chaves sendo uma pública para o usuário, e uma privada para o servidor.
- MAC (Message Authentication Code): Código adicionado em cada mensagem SSL/TLS
- Certificados Digitais: As chaves são registradas garantindo a sua autenticidade.

Além disso podemos identificar outras diferenças entre os dois protocolos:

- A URL no HTTP inicia com http:// , e no HTTPS a URL inicia com https:// .
- HTTP usa a porta 80 para comunicação e o HTTPS usa a porta 443.
- HTTP é considerado inseguro, enquanto o HTTPS é considerado seguro.
- HTTP funciona na camada de aplicação, enquanto HTTPS funciona na camada de transporte.
- HTTP não necessita de certificados, já o HTTPS requer certificados SSL.
- HTTP não utiliza criptografia, enquanto no HTTPS os dados são criptografados antes de serem enviados.

2. **Qual o formato de uma requisição HTTP?**

O usuário faz uma requisição para um servidor e espera que o servidor consiga entender essa requisição e o responda, e para fazer essa requisição ele usa:

- Linha inicial contendo:

1 - Método HTTP – que descreve a ação a ser executada

2 - A URL ou o caminho do protocolo

3 - A versão HTTP

Exemplo: GET/ HTTP/1.1

- Cabeçalho (Request Header):

Utiliza uma cadeia de caracteres insensível à caixa seguida de dois pontos (':') e um valor cuja estrutura depende do cabeçalho. O cabeçalho inteiro, incluindo o valor, consiste em uma única linha, que pode ser bem grande.

- Corpo (Body Request):

É a parte final da requisição, nem toda requisição tem um corpo, ele aparece em alguns casos quando a requisição envia dados ao servidor para realizar atualizações. E esse corpo pode ser dividido em duas categorias, corpo de recursos simples, que contem apenas um arquivo e o corpo de recurso-múltiplo que tem várias partes contendo informações.

A requisição tem esse esquema:

Fonte: https://www.alura.com.br/artigos/desmistificando-o-protocolo-http-parte-1

3. Qual o formato de uma resposta HTTP?

O usuário faz uma requisição para um servidor e espera que o servidor consiga entender essa requisição e o responda, se o servidor entender ele responde da seguinte forma:

- Linha de Status:

1 - A versão HTTP

2 - Um código de status

3 - Um texto de status

Exemplo: HTTP/1.1 404 Not Found.

- Cabeçalho (Response Header):

Utiliza uma cadeia de caracteres insensível à caixa seguida de dois pontos (':') e um valor cuja estrutura depende do cabeçalho. O cabeçalho inteiro, incluindo o valor, consiste em uma única linha.

- Corpo (Body Response):

É a parte final da resposta, nem toda resposta tem um corpo, aquelas com código de status 201 ou 204 normalmente não tem. E esse corpo pode ser dividido em três categorias, corpo de recurso simples que consistem em um único arquivo de tamanho

conhecido, corpo de recurso simples que consistem em um único arquivo de tamanho desconhecido, codificado aos pedaços e corpo de recurso-múltiplo, que consiste em um corpo várias partes, cada uma contendo diferentes informações.

A resposta tem esse esquema:

Fonte: https://www.alura.com.br/artigos/desmistificando-o-protocolo-http-parte-1

4. **Quando um servidor não encontra um recurso, quais os principais códigos de status que existem? Por exemplo 404? O que significa?**

Os códigos de status das respostas HTTP indicam se uma requisição HTTP foi corretamente concluída.

Esses códigos são divididos em classes e intervalos nas faixas 100, 200, 300, 400 e 500.

Respostas de informação (100-199)
100 Continue
101 Switching Protocol
102 Processing (WebDAV)
103 Early Hints
Respostas de sucesso (200-299)
200 OK
201 Created
202 Accepted
203 Non-Authoritative Information
204 No Content
205 Reset Content
206 Partial Content
207 Multi-Status (WebDAV)
208 Multi-Status (WebDAV)
226 IM Used (HTTP Delta encoding)
Redirecionamentos (300-399)
300 Multiple Choices
301 Moved Permanently
302 Found
303 See Other
304 Not Modified
307 Temporary Redirect
308 Permanent Redirect
Erros do cliente (400-499)
400 Bad Request
401 Unauthorized
402 Payment Required
403 Forbidden

404 Not Found
405 Method Not Allowed
406 Not Acceptable
407 Proxy Authentication Required
408 Request Timeout
409 Conflict
410 Gone
411 Length Required
412 Precondition Failed
413 Payload Too Large
414 URI Too Long
415 Unsupported Media Type
416 Range Not Satisfiable
417 Expectation Failed
418 I'm a teapot
422 Unprocessable Entity
425 Too Early
426 Upgrade Required
428 Precondition Required
429 Too Many Requests
431 Request Header Fields Too Large
451 Unavailable For Legal Reasons
Erros do servidor (500-599)
500 Internal Server Error
501 Not Implemented
502 Bad Gateway
503 Service Unavailable
504 Gateway Timeout
505 HTTP Version Not Supported
506 Variant Also Negotiates
507 Insufficient Storage
508 Loop Detected
510 Not Extended
511 Network Authentication Required

O código 404 significa ‘Not Found.’ Que é um código de erro informando que o servidor não consegue encontrar o recurso solicitado pelo cliente.

5. Quais as principais diferenças do HTTP v1 para o HTTP v2?

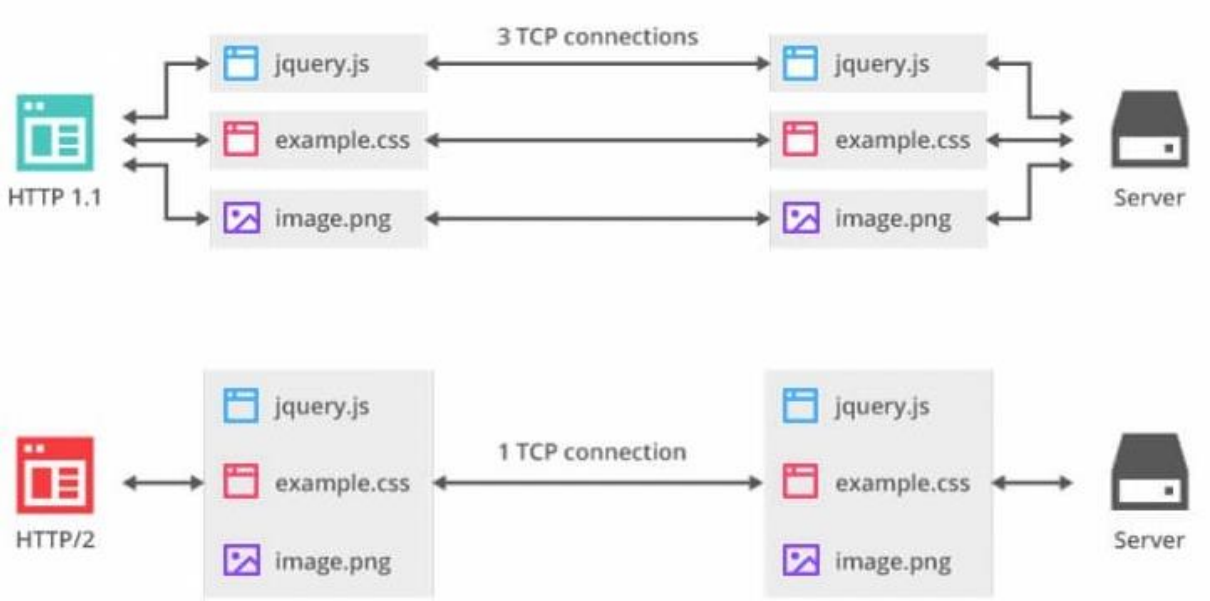
A primeira versão do protocolo HTTP foi lançada em 1991 que foi a HTTP/0.9 e só possuía o método GET, em 1996 foi feita uma nova versão a HTTP/1, que já conseguia

realizar transferências de arquivos mais ricos, não somente de textos como a primeira versão.

Em 1999 foi disponibilizada a terceira versão HTTP/1.1 que foi considera um marco e que define o padrão da internet. Essa versão resolveu vários problemas encontrados na versão anterior, e inseriu melhorias críticas de performance.

Até que em 2015, com os avanços da internet e suas complexidades, foi lançado uma nova versão do protocolo, o HTTP/2, que foi uma evolução do protocolo SPDY. O protocolo SPDY foi criado em 2010 com o objetivo de melhorar a forma com que o HTTP lidava com as requisições e respostas, seu principal objetivo era de reduzir a latência via TCP fazendo com que várias requisições HTTPs fossem enviadas em uma única conexão TCP, sem ter que ficar esperando pela resposta correspondente, como acontecia na versão anterior, onde só se processava uma solicitação de conexão TCP por vez, forçando os navegadores a utilizarem múltiplas conexões TCP para processar múltiplas requisições simultaneamente.

O HTTP/2 utiliza uma técnica chamada de multiplexação que possui uma conexão TCP que é persistente e precisa somente de uma por origem, onde as requisições e respostas paralelas poderão requisitar e receber todos os arquivos necessários. Assim fazendo com que haja uma redução no consumo de processamento e memória, uma redução no custo operacional das redes e uma maior capacidade de uso. O resultado é uma redução de latência na rede e redução de custo em hardware e software.



Fonte: https://blog.neomind.com.br/diferencas-entre-http1-1-e-http2/