

Aula V - Introdução à HTTP

Amanda Luna, Anderson Nóbrega e João Soares

Incodde, Imersão Incodde

O que é HTTP?

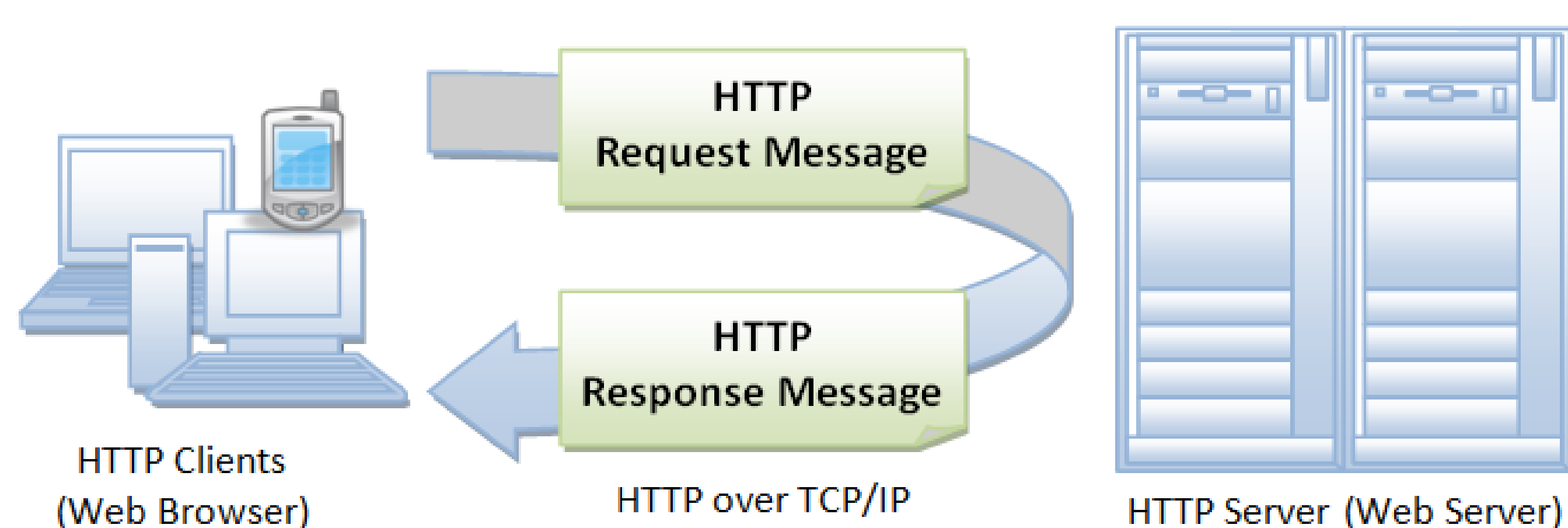
HTTP é um *protocolo de transferência* o qual proporciona que as pessoas que inserem a URL de um site possam ver os conteúdos e dados que nele existem. A sigla vem do inglês Hypertext Transfer Protocol.

Qualquer servidor que você escolha para hospedar um site tem um sistema/*API* para receber solicitações HTTP. Portanto, o navegador que você usa é um *cliente HTTP* que envia solicitações constantemente ao seu servidor.

Assim, quando um usuário acessa ou digita a URL do seu site, o navegador cria uma solicitação HTTP e a envia ao servidor, que a recebe e retorna os dados associados.

Desafio 06

Descrição: Utilizando como referência a lista feita no desafio 01 ou criando uma nova, faça um objeto JSON(Javascript Object Notation), em seguida crie um servidor http que o retorne.



Primeiro servidor HTTP

Para criar o primeiro servidor *http*, usaremos uma biblioteca padrão do javascript chamada também de *http*. Para importar uma biblioteca em nosso código utilizamos a função *require*, em que iremos requisitar e instanciar o servidor.

Para que a conexão se estabeleça corretamente, necessitaremos declarar o *host* e a *porta*, em que o host se refere ao endereço de ip utilizado para acessarmos o nosso servidor e porta a qual porta abriremos para que haja a comunicação. A porta padrão para *http* é a 80 e para *https* é 443. Geralmente no desenvolvimento nós alteramos estas portas, para que não haja conflito entre nossos serviços(api,front,worker e similares).

• Criando o servidor

Para que o servidor seja efetivamente criado, utilizaremos a função do *http* chamada *createServer*, na qual passaremos como parâmetro uma função anônima, cujos argumentos serão *req*(request) e *res*(response). Como apenas iniciaremos o servidor, não precisamos nos preocupar com a request por enquanto, apenas com a response. Ou seja, a resposta que enviaremos quando recebermos uma requisição(alguém acessar nossa URL).

Podemos retornar várias coisas como resposta às requisições, porém temos três principais: o status da resposta(se deu certo ou não), o header(tipo de conteúdo que vai ser retornado) e o end, que nada mais é do que o conteúdo em si.

Após isto, chamaremos a função *listen*, passando o *host* e a porta, que fica "escutando" nesse endereço, para que quando haja uma requisição, ele avise ao nosso servidor e retorne o conteúdo referente àquela página, naquela porta.

Observando a imagem abaixo, poderemos ver um exemplo de um servidor *http* criado manualmente.

```
const http = require('http');
const host = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end("Estamos acessando o servidor!");
})

server.listen(port, host, () => {
  console.log("Server listening on port " + port);
})
```