HTTP COM NODE JS

Douglas Nassif Roma Junior

/douglasjunior

/in/douglasjunior

nassifrroma@gmail.com

Slides: https://github.com/douglasjunior/Bootcamp-DB1-2024

AGENDA

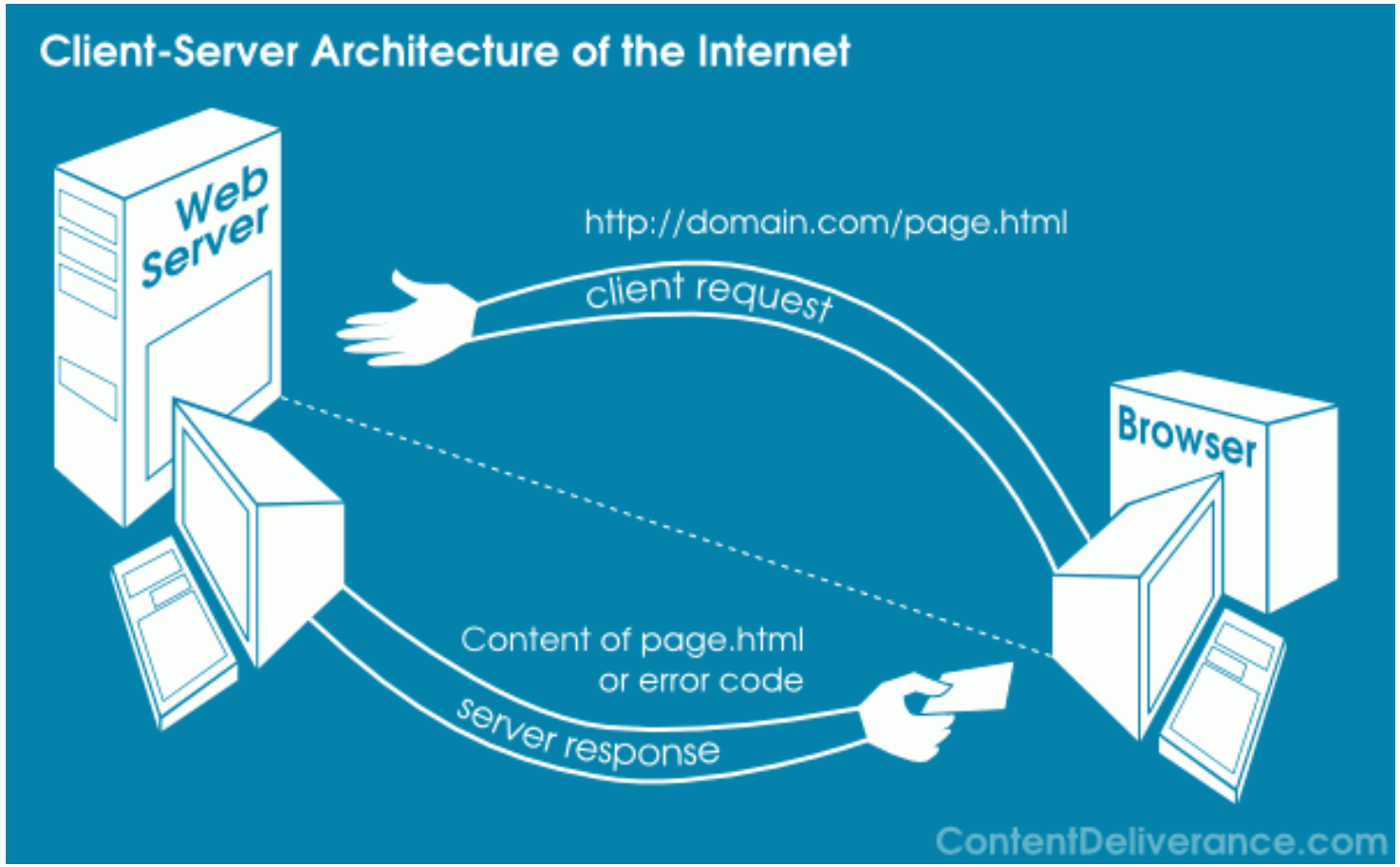
- Protocolo HTTP
- HTTP com Node JS
- Express JS
- Criação de rotas
- Middlewares
- Express Generator
- Jason Web Token
- Referências

PROTOCOLO HTTP

PROTOCOLO HTTP

http://arquivo.devmedia.com.br/REVISTAS/java/imagens/117/3/1.png

PROTOCOLO HTTP



http://contentdeliverance.com/cms-school/wp-content/uploads/2011/05/client-server-diagram-internet.png

HTTP COM NODE JS

HTTP COM NODE JS

- Crie um novo diretório chamado `MeuProjetoWeb`
- Repita os passos que você já aprender para iniciar um projeto com `npm`.
- Crie um arquivo `index.js` para escrever o código da aplicação.
- Inicie um servidor HTTP que seja capaz de escrever a frase "Olá Node JS!".

HTTP COM NODE JS

- Importe o módulo `http` e inicie o serviço na porta desejada.

```
const http = require('http');
const porta = 3000;

const server = http.createServer((request, response) => {
  response.setHeader('content-type', 'text/html; charset=utf-8');
  response.writeHead(200);
  response.write('Olá Node JS!');
  response.end();
});

server.listen(porta, () => {
  console.log('Servidor iniciado na porta:', porta);
});
```

- Abra no navegador `http://localhost:3000`

HTTP COM EXPRESS

HTTP COM EXPRESS

- **Instale a dependência** `express`:

```
$ npm install express
```

- **Modifique o exemplo anterior para utilizar** `express` **no lugar do pacote nativo** `http`.

- **Defina rotas GET e POST para criação e listagem de usuários.**

HTTP COM EXPRESS

- Modificando o exemplo anterior

```
const express = require('express');
const porta = 3000;

const server = express();

server.use('/', (request, response) => {
  response.setHeader('content-type', 'text/html; charset=utf-8');
  response.status(200);
  response.send('Olá Node JS!');
});

server.listen(porta, () => {
  console.log('Servidor iniciado na porta:', porta);
});
```

- Abra no navegador `http://localhost:3000`

CRIAÇÃO DE ROTAS

- As rotas são as URLs que a aplicação será capaz de responder.

```
const express = require('express');

const server = express();

const porta = 3000;

server.use(express.json());

server.get('/usuarios/:usuarioId/', (request, response) => {
  const usuarioId = request.params.usuarioId;
  response.status(200);
  response.send("Listando usuários: " + usuarioId);
});

server.post('/usuarios/', (request, response) => {
  const usuario = request.body;
  console.log('Usuários recebido:', usuario);
  response.status(201).send();
});

server.listen(porta, () => {
  console.log('Servidor iniciado na porta:', porta);
});
```

MIDDLEWARES

- Middlewares atuam como mediadores que interceptam as requisições para realizar algum tipo de tarefa.
  - Por exemplo, validação, autenticação e etc.

- Usando:

```
server.use((request, response, next) => {
  const token = request.headers.token;

  if (token == 'meu-token') {
    next();
  } else {
    response.status(403);
    response.send();
  }
});
```

EXPRESS GENERATOR

EXPRESS GENERATOR

- Facilita a criação e organização inicial do projeto.

- Para criar um novo projeto:

```
$ npx express-generator MeuProjetoExpress
```

- Entre na pasta criada pelo gerador, e então instale as dependências:

```
$ cd MeuProjetoExpress
$ npm install
```

JASON WEB TOKEN (JWT)

JWT

- JSON Web Token (JWT) é um padrão industrial aberto (RFC 7519) para representar reivindicações de forma segura entre duas partes.

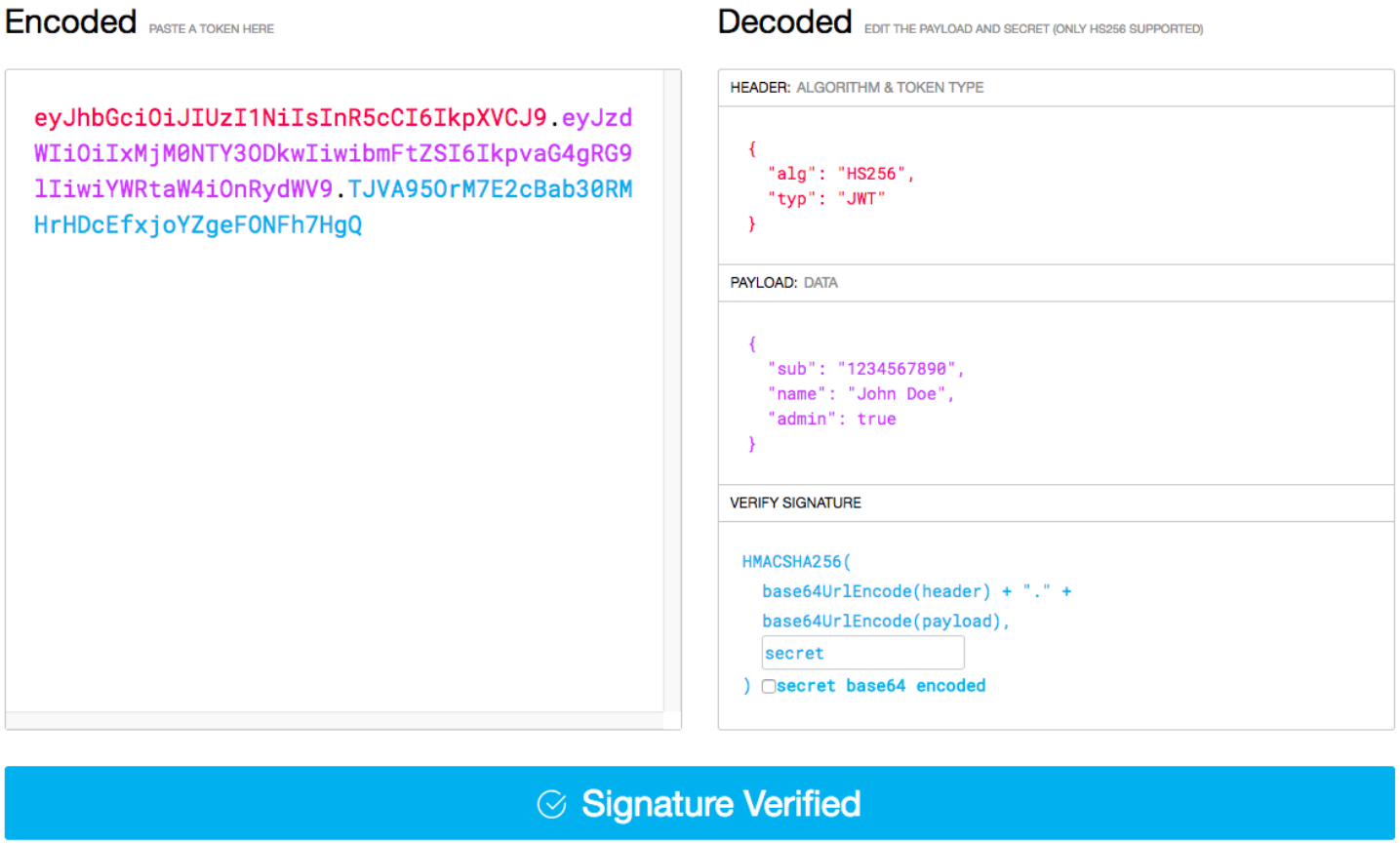
- O JWT é formado de três partes: **Header**, **Payload** e **Signature**.

- Exemplo:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI
6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONF
h7HgQ
```

JWT

- É possível testar e validar o seu JWT diretamente no site https://jwt.io



JWT COM NODE JS

- Para trabalhar com JWT no Node, basta instalar a dependência:

```
$ npm install jsonwebtoken
```

- Para codificar um JWT:

```
const jwt = require("jsonwebtoken");

const secret = "minha-senha-super-secreta";

const payload = {
  usuario: {
    id: 1,
    nome: 'Douglas',
  },
  tipo: 'administrador',
};

const token = jwt.sign(payload, secret);

console.log('Meu Token: ', token);
```

JWT COM NODE JS

- Para decodificar:

```
const jwt = require("jsonwebtoken");

const secret = "minha-senha-super-secreta";

const token =
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3VhcmlvIjp7ImlkIjo
xLCJub21lIjoiRG91Z2xhcyJ9LCJ0aXBvIjoiYWRtaW5pc3RyYWRvciIsIml
dCI6MTY4NjUwNzUxMn0.ki_-
SmcWaIVkuM8GL4EMkTccZnrthVTmVLNCgIA7m7E";

const payload = jwt.verify(token, secret);

console.log('Meu Payload: ', payload);
```

REFERÊNCIAS

- DevMedia: Como funcionam as aplicações Web - http://www.devmedia.com.br/como-funcionam-as-aplicacoes-web/25888
- Node JS HTTP - https://nodejs.org/api/http.html#http_class_http_server
- Express JS - http://expressjs.com/pt-br/
- Express Generator - https://expressjs.com/en/starter/generator.html
- Json Web Token - https://jwt.io/
- Node Json Web Token - https://github.com/auth0/node-jsonwebtoken

DÚVIDAS?

Douglas Nassif Roma Junior

/douglasjunior

/in/douglasjunior

nassifrroma@gmail.com

Slides: https://github.com/douglasjunior/Bootcamp-DB1-2024