

HTTP COM NODE JS



Douglas Nassif Roma Junior

 /douglasjunior

 /in/douglasjunior

 douglasjunior.me

 nassifrroma@gmail.com

Slides: <https://git.io/vbU3N>

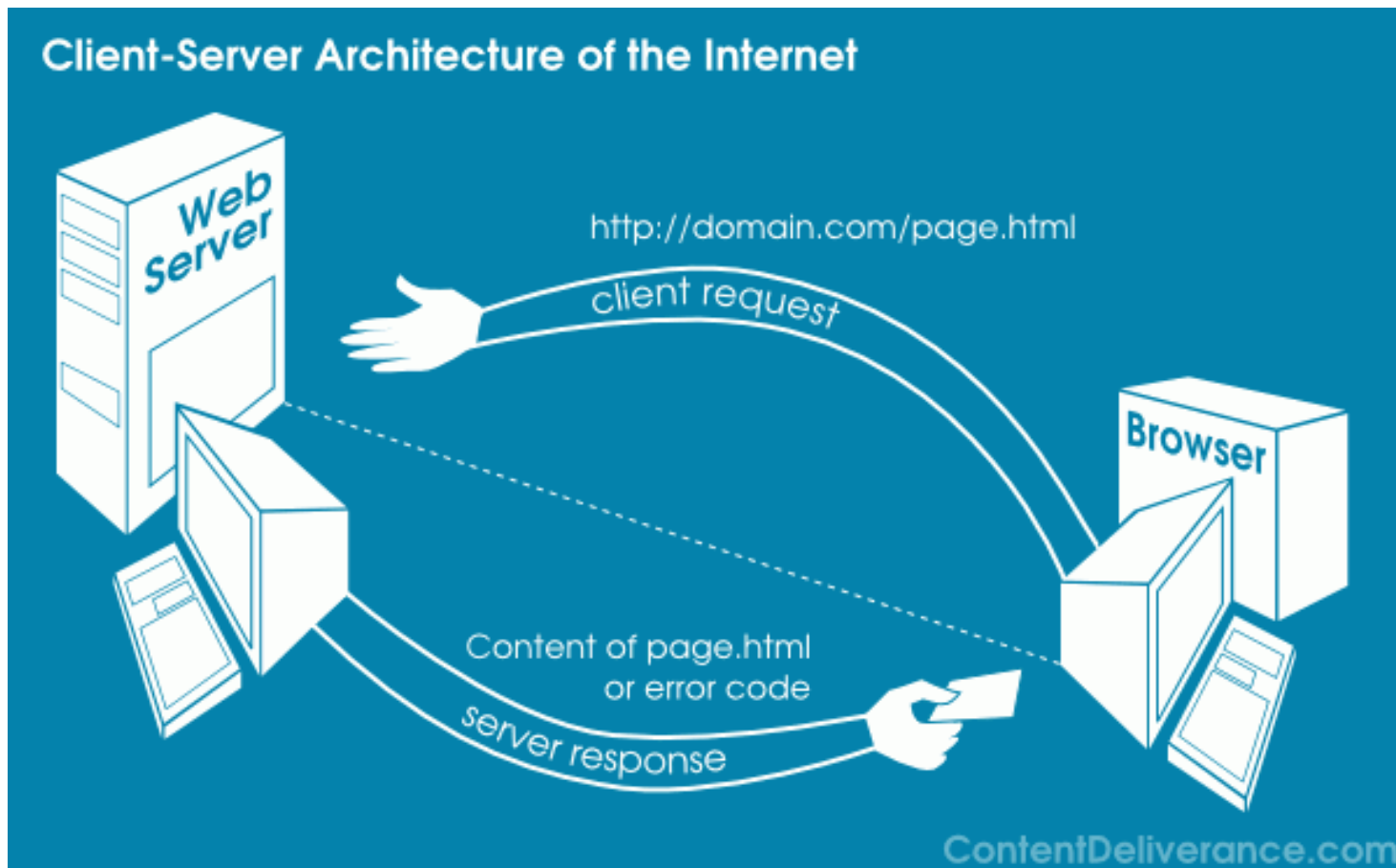
AGENDA

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

PROTOCOLO HTTP



PROTOCOLLO HTTP



HTTP COM NODE JS

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

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

http.createServer((request, response) => {

    response.setHeader('content-type', 'text/html; charset=utf-8');
    response.writeHead(200);
    response.write('Olá Node JS!');
    response.end();

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

server.post('/usuarios/', (request, response) => {
  let usuario = "";
  request.on('data', (chunk) => { usuario += chunk });
  request.on('end', () => {
    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, body-parser e etc.

- Registrando o middleware:

```
const bodyParser = require('body-parser');
server.use(bodyParser.json());
server.use(bodyParser.urlencoded({ extended: false }));
```

- Usando:

```
server.use((request, response, next) => {
  if (usuarioAutenticado) {
    next();
  } else {
    response.status(403).send();
  }
});
```

EXPRESS VALIDATOR

- Express Validator é um middleware que facilita a validação do corpo da requisição e parâmetros da URL.

```
server.post('/usuarios/', (request, response, next) => {
  request.check({
    nome: {
      in: "body",
      notEmpty: true,
      errorMessage: "Informe o nome do usuário."
    }
  });
  request.getValidationResult()
    .then((result) => {
      if (result.isEmpty()) {
        next();
      } else {
        response.status(400).json(result.array());
      }
    })
    .catch(next);
}, (request, response) => {
  let usuario = request.body;
  console.log('Usuários recebido:', usuario);
  response.status(201).send();
});
```


EXPRESS GENERATOR

- Facilita a criação e organização inicial do projeto. Instale a dependência globalmente:

```
$ npm install express-generator -g
```

- Para criar um novo projeto:

```
$ express meu-projeto
```

- Instale as dependências do novo projeto:

```
$ cd meu-projeto
```

```
$ npm install
```

- eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoiYWRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

JWT

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

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoiYWRtaW4iOnRydWV9.TJVA950rM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ
```

Decoded

EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

HEADER: ALGORITHM & TOKEN TYPE

```
{  "alg": "HS256",  "typ": "JWT"}
```

PAYLOAD: DATA

```
{  "sub": "1234567890",  "name": "John Doe",  "admin": true}
```

VERIFY SIGNATURE

```
HMACSHA256(  base64UrlEncode(header) + "." +  base64UrlEncode(payload),  secret)
```

secret

☐secret base64 encoded

JWT

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

```
$ npm install --save jsonwebtoken
```

- Para codificar um JWT:

```
const jwt = require("jsonwebtoken");  
  
const secret = "minha-senha-super-secreta";  
const token = jwt.sign(payload, secret);
```

- Para decodificar:

```
const jwt = require("jsonwebtoken");  
  
const secret = "minha-senha-super-secreta";  
const payload = jwt.verify(token, secret);
```

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 Validator - <https://github.com/ctavan/express-validator>
- Express Generator - <http://expressjs.com/pt-br/starter/generator.html>
- Json Web Token - <https://jwt.io/>
- Node JsonWebToken - <https://github.com/auth0/node-jsonwebtoken>

DÚVIDAS?



Douglas Nassif Roma Junior

 /douglasjunior

 /in/douglasjunior

 douglasjunior.me

 nassifrroma@gmail.com

Slides: <https://git.io/vbU3N>