



Aula – 11

Introdução ao ExpressJS

Disciplina: XDES03 – Programação Web

Prof: Phyllipe Lima Francisco
phyllipe@unifei.edu.br

Universidade Federal de Itajubá – UNIFEI
IMC – Instituto de Matemática e Computação

Agenda



- ☐ Introdução
- ☐ Criando Rotas
- ☐ Params
- ☐ Query Params
- ☐ Verbos HTTP
- ☐ Forms com Express

Verbos HTTP

☐ **GET**

- ☐ Usado para buscar informações
- ☐ Dados podem ser passados via Query Params / Params
- ☐ Dados ficam visíveis na URL

☐ **POST**

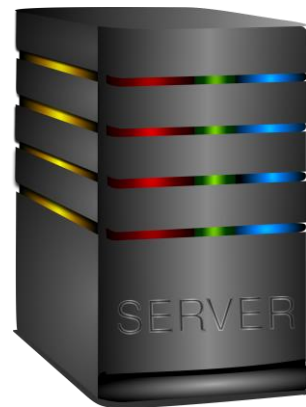
- ☐ Usado para enviar informações para o servidor
- ☐ Pode escrever/criar/atualizar
- ☐ Dados são enviados no corpo da requisição (*body request*)

Requisições GET



Requisição GET

<https://www.bing.com/search?q=playstation>



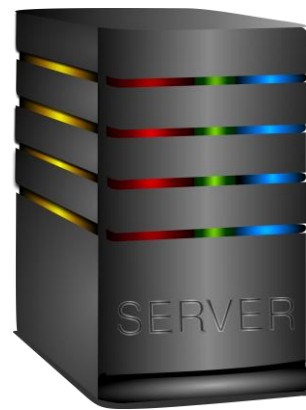
Endpoint: <https://www.bing.com/search>

Query Param: q=playstation

Requisições POST



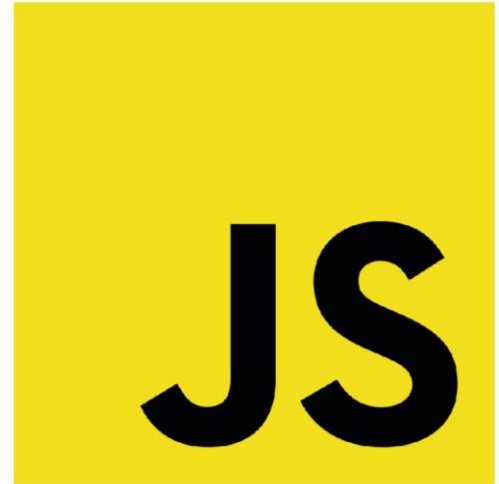
Requisição POST (hipotético)
`https://www.bing.com/search`



Endpoint: <https://www.bing.com/search>
Os parâmetros estariam no corpo da
requisição como parte do *payload*

Express JS

Express



Express JS - Introdução

- ❑ Framework (arcabouço) para criação de aplicações web do NodeJS.
- ❑ Apesar de ser possível criar puramente com NodeJS, o uso do ExpressJS torna o processo muito mais rápido.
- ❑ É o *padrão de facto* para criação de servidores NodeJS

Express JS - Introdução

❑ ExpressJS nos ajuda com:

- ❑ Criar um servidor e “ouvir” requisições.
- ❑ Analisar as requisições recebidas.
- ❑ Converter as requisições para objetos JavaScript
- ❑ Verificar qual rota (*endpoint*) irá tratar a requisição.
- ❑ Construir a resposta HTTP com o cabeçalho e conteúdo adequado.

Express JS - Instalação

- ❑ `:$ npm i express`
- ❑ Caso não tenha inicializado o npm, execute o comando na raiz do projeto
 - ❑ `:$ npm init`

Exemplo 1 – Criando um Servidor

- ❑ Inicializando um servidor Express JS
- ❑ Utilizar o “nodemon” para executar o servidor ao invés do “node” puro.
 - ❑ Instalar nodemon globalmente
 - ❑ `npm i -g nodemon`
- ❑ Nodemon reinicia o servidor automaticamente com mudanças no código.

Exemplo 1 – Etapas

1. Inicializar um projeto Node com “npm init -y” (força os valores padrões)
2. Instalar o express com “npm i express”
3. Importar o express e criar uma instância (app)
4. Inicializar um servidor na porta 3000
`app.listen()`
5. Tratar todas as requisições
`app.use()`
6. Executar com “nodemon index.js”

Exemplo 1 – Código

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

```
const app = express();
```

```
app.listen(3000, () => {  
  console.log('Servidor ouvindo na porta 3000!!!!');  
});
```

```
app.use(() => {  
  console.log('Requisição Recebida!!!!');  
})
```

Exemplo 2 – Enviando uma Resposta

- ❑ Utilizando o método `send()`
- ❑ Utilizando os argumentos `(req,res)`.
 - ❑ `(req)` e `(res)` são criados pelo `express` para transformar a requisição HTTP em objetos JavaScript.
- ❑ Retornando JSON, HTML e Texto (plain text)

Exemplo 2 – Código

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

const app = express();

app.listen(3000, () => {
  console.log('Servidor ouvindo na porta 3000!!!!');
});

app.use((req,res) => {
  //res.status(200).send('Olá. Essa é uma resposta!');
  res.send('<h1>Resposta direto com HTML</h1>');
  //res.send({username : 'phillima'});

});
```

Exemplo 3 – Criando Rotas

- ❑ Utilizando o método "get" do express crie 4 rotas.
- ❑ Rota 1 : /ps5
 - ❑ Devolve a resposta "Você acessou Rota do PS5"
- ❑ Rota 2: /pc
 - ❑ Devolve a resposta "Você acessou a Rota do PC"
- ❑ Rota 3 :/ (raiz)
 - ❑ Devolve a resposta "Você acessou a Rota Game Dev Raiz"
- ❑ Rota 4 : * (genérica)
 - ❑ Devolve a resposta "Desconheço a Rota, você deve jogar no celular".

Exemplo 4 – Passando Parâmetros com Params

- ❑ Utilizando o arquivo “disciplinas.json”, escreva um programa que recebe requisições e retorna as informações das disciplinas.
- ❑ Endpoint: /disciplinas => retorna o nome de toda as disciplinas em formato de lista HTML.
- ❑ Endpoint: /disciplinas/:sigla => retorna as informações da disciplina com a sigla enviada (nome, ementa e equivalência)
 - ❑ Caso a sigla não seja válida, retorne uma mensagem informando o erro.

Exemplo 4 – Passando Parâmetros com Params

- ❑ Etapas
- ❑ Passo 1 – Código para Tratar Requisições GET
- ❑ Passo 2 – Verificar se o parâmetro chega corretamente pela URL
- ❑ Passo 3 – Código para manipular o arquivo “disciplinas.json”
- ❑ Passo 4 – Código para construir a resposta HTML
- ❑ OBS: Em casos profissionais, usualmente, devolveríamos apenas os dados e não o HTML. Cabe ao frontend construir a apresentação. Faremos isso com React nas próximas aulas.

Exemplo 5 – Passando Parâmetros com Query Params

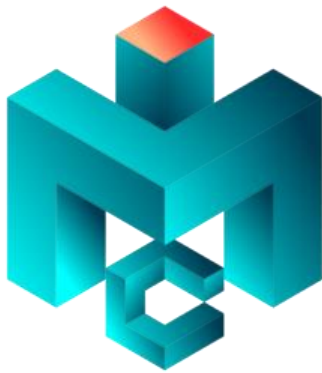
- ❑ Repita o Exemplo 4 utilizando Query Parameters
- ❑ Query Parameters são passados na URL no formato chave-valor.
- ❑ Se iniciam logo após o endpoint com o caractere "?"
- ❑ Diversos parâmetros podem ser passados e são separados por "&".
- ❑ Exemplo de URL que podemos usar
- ❑ Localhost:3000/disciplinas?sigla=VALOR
 - ❑ Parâmetro é "sigla" com o valor "VALOR"

Exemplo 6 – Forms + Express

- ❑ Crie uma aplicação web com Forms submetendo requisições POST e GET.
- ❑ O Forms deverá conter apenas o campo “sigla” para.
- ❑ As requisições POST e GET são tratadas e locais distintos, e devem apenas enviar uma resposta “recebido” com a sigla passada como parâmetros.
- ❑ No caso de requisições POST, o parâmetros são passados via *body request*

Exemplo 7 – Forms + Express Parte 2

- ❑ Combinando os Exemplos anteriores, crie uma aplicação que permita inserir uma nova sigla de disciplina no arquivos “disciplinas.json”.
- ❑ Os dados devem ser inseridos via Forms com requisição POST
- ❑ Um ID incremental deve ser gerado pela aplicação.



Aula – 11

Introdução ao ExpressJS

Disciplina: XDES03 – Programação Web

Prof: Phyllipe Lima Francisco
phyllipe@unifei.edu.br

Universidade Federal de Itajubá – UNIFEI
IMC – Instituto de Matemática e Computação