

WebDev - Módulo 3

Roteiro de aula

Aula 2 - Hard: Burocracia



Tópicos da aula:

- Revisão de conceitos de chamadas de APIs e de protocolos da Web (HTTP)
- Fetch api
- XMLHttpRequest vs Fetch
- node-fetch



Objetivos da aula:

1. Relembrar conceitos de protocolos da web (HTTP)
2. Revisar conceitos de requisições
3. Reconhecer a vantagem de se empregar Fetch, no lugar de XMLHttpRequest
4. Empregar Fetch para realizar requisições web



Atividades da Aula

→ Atividade 1: Pegando Pokémons

◆ Usando o node:

- Instale o "node-fetch" na versão 2;
- Importe o "node-fetch" ("require");
- Realize um fetch para a [PokéAPI](#);
- Utilize a rota para pegar todos os pokémons;
- Acaso tenha sucesso;
- Transforme o resultado em JSON;
- Imprima o nome do pokémon no console.



Momento Aprendizagem em Pares

Esse momento é dedicado para vocês desenvolverem suas demandas e entregas para o curso.

- Utilize esse tempo da maneira que preferir, mas atente-se às aulas que você deve realizar as entregas.
- ◆ **Dica:** Nas Propostas dos projetos, vocês encontram uma sugestão de organização para a realização das atividades.
 - ◆ **Lembre-se:** O momento de Aprendizagem em Pares é justamente para fazer trocas e aprender em comunidade, aproveite seus colegas e se desenvolvam juntos!
- **Entregas:**
- ◆ Projeto individual: aula 5 - HARD
 - ◆ Projeto em grupo: aula 10 - HARD
 - ◆ Apresentação do projeto: aula 10 - HARD



Revisão da aula

- **HTTP** é um **protocolo de comunicação via web** que tornou-se o padrão de comunicação para enviar e receber dados entre computadores pela internet. Ou seja, **toda interação que você realizar com um sistema web pelo seu navegador de internet, utiliza uma requisição HTTP por "debaixo dos panos"**. Então sim, quando você entra em algum site de Ecommerce, por exemplo, para ver os produtos o seu navegador está utilizando uma chamada HTTP para obter todas as informações que você está visualizando agora. E toda vez que você adiciona o produto em um carrinho virtual, o seu navegador também está enviando requisições HTTP para um servidor do Ecommerce.
- Mas **o que significa "HTTP"**? As siglas se referem a **Hypertext Transfer Protocol** e possuem um método, um caminho e uma versão de protocolo. Os métodos são conhecidos como **Get** (buscar ou pegar), **Post** (inserir enviar), **Put-Patch** (atualizar) e **Delete** (deletar) e ajudam a identificar qual será o tipo da requisição e qual sua finalidade. Essas requisições percorrem um **caminho** (path) para a conexão com o servidor. Já a versão preferida dos desenvolvedores é a HTTP 1.1.

- A **Fetch API** é considerada a versão mais simples de usar da XMLHttpRequest, quando consideramos consumir recursos de modo assíncrono. A **Fetch trabalha com promises em vez de callbacks**. A promise nós vimos na primeira aula dessa unidade. Aos poucos os desenvolvedores vem fugindo de usar callbacks e mergulhando de cabeça nas promises, acreditando na sua maior facilidade em escrever solicitações assíncronas.
- Antes da hegemonia do formato JSON, o formato principal de troca de dados era o XML. A **XMLRequest()** é uma função possível obter dados das APIs que retornavam dados em XML. A XMLHttpRequest **nos deu a opção de buscar dados em XML do back-end sem recarregar toda a página**. Essa função cresceu desde seus primeiros dias em que trabalhava apenas com dados em XML.



```
1 fetch("url")
2   .then((result) => {
3     console.log(result);
4   })
5   .catch((erro) => {
6     console.log(erro);
7   });
```

- Para utilizarmos o **fetch no node.js** precisamos instalar a biblioteca node-fetch e importar. Depois que instalamos o "node-fetch" e fizemos o "require", vamos realizar um fetch para acessar a **PokéAPI**, que basicamente é um banco de dados com todos os pokémons e nos é disponibilizado à API. Você vai utilizar a rota(path) para pegar todos os pokémons e no final vai imprimir o nome do pokémon no console.



Para ajudar



Links interessantes:

- [PUT vs PATCH \(What's the Difference?\) | RapidAPI](#)
- [HTTP headers - HTTP | MDN \(mozilla.org\)](#)
- [Evolution of HTTP — HTTP/0.9, HTTP/1.0, HTTP/1.1 Medium](#)
- [HTTP | MDN \(mozilla.org\)](#)