



Universidade Federal de Mato Grosso

Bacharelado em Ciência da Computação

Semestre: 2025/1

Programação 4

Atividade - Requisições Assíncronas

Objetivo

O objetivo deste exercício é aplicar os conhecimentos de JavaScript assíncrono (`async/await`, `fetch`) para interagir com uma API pública do mundo real. Você irá escolher uma API, fazer uma requisição baseada em uma entrada do usuário e exibir os dados recebidos de forma dinâmica na página.

Passo 1: A Missão de Exploração (Escolha sua API)

1. Acesse a lista de APIs públicas no seguinte endereço: [Public APIs no GitHub](#).
2. Explore as categorias e escolha uma API que lhe pareça interessante.
3. **Requisito:** Dê preferência a APIs que na coluna "**Auth**" estejam marcadas como "**No**", pois não exigem autenticação e são mais simples de usar para este exercício.

Sugestões de APIs simples e sem autenticação:

- **Cat Facts:** Busca fatos aleatórios sobre gatos.
- **Universities List:** Lista universidades por país.
- **Public Holidays:** Informa feriados públicos de um determinado país e ano.

Depois de escolher, identifique a URL base e qual parâmetro você usará para fazer a busca (por exemplo, o nome de um país, um ano, etc.).

Passo 2: A Estrutura (HTML)

Crie um arquivo `index.html`. A estrutura da página deve ser simples e conter os seguintes elementos essenciais:

- Um título, como `<h1>Meu Explorador de APIs</h1>`.
- Um campo de texto (`<input type="text">`) para o usuário digitar a busca. Dê a ele um `id`, como `search-input`.
- Um botão (`<button>`) para iniciar a busca. Dê a ele um `id`, como `search-button`.

- Uma div vazia que servirá como contêiner para os resultados. Dê a ela um `id`, como `results`.
- No final do `<body>`, inclua a tag `<script src="app.js"></script>` para carregar o seu código JavaScript.

Você pode adicionar qualquer estilo CSS que desejar, mas o foco do exercício é na lógica JavaScript.

Passo 3: A Lógica (JavaScript)

Crie um arquivo `app.js` e siga as instruções abaixo:

1. **Selecione os Elementos do DOM:** Obtenha as referências para o campo de input, o botão e a div de resultados.
2. **Crie uma Função Assíncrona:** Crie uma função `async` chamada `buscarDados`. Esta função será a responsável por todo o processo.
3. **Construa a URL:** Dentro de `buscarDados`, pegue o valor digitado pelo usuário e use-o para construir a URL completa da sua requisição.
4. **Faça a Requisição com fetch:**
 - Use um bloco `try...catch` para lidar com possíveis erros de rede.
 - Dentro do `try`, use `await fetch(suaUrl)` e `await response.json()`.
5. **Exiba os Dados:**
 - Crie uma função separada chamada `exibirResultados(dados)`.
 - Dentro desta função, limpe a div de resultados.
 - Inspeção o objeto recebido no console (`console.log(dados)`) para entender sua estrutura.
 - Percorra os resultados e crie novos elementos HTML para exibir as informações que achar interessantes, adicionando-os à div de resultados.
6. **Adicione o Event Listener:** Adicione um `addEventListener` ao botão de busca para que ele chame a sua função `buscarDados` (garanta também que isto aconteça se pressionar ENTER no campo).