

Web

Prática 20

fetch

GitHub

<https://github.com/wdiasmaciел/js-fetch-API>

Analise e execute os exemplos a seguir.

1) GET /posts/:id

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>fetch</title>
  <style>
    html,
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      font-size: 16px;
    }

    main {
      width: 51rem;
      margin: 1rem auto;
    }

    input[type="number"],
    button {
      text-align: right;
      padding: 0.5rem;
    }

    input[type="number"] {
      width: 10rem;
      margin-bottom: 0.5rem;
    }
  </style>
</head>

<body>
  <main>
    <div>
      <input type="number" value="1" />
      <button>GET</button>
    </div>
  </main>
</body>
</html>
```

```

button {
  margin-bottom: 0.5rem;
}

#resp {
  padding: 0.5rem;
  background-color: orangered;
  color: white;
  width: 50rem;
}

.campo {
  color: yellow;
  font-weight: bold;
}
</style>

<script>
function imprime(objeto) {
  var mensagem = `
  <br>
  {
    <br>
    "<span class='campo'>id</span>": ${objeto.id},
    <br>
    "<span class='campo'>userId</span>": ${objeto.userId},
    <br>
    "<span class='campo'>title</span>": ${objeto.title},
    <br>
    "<span class='campo'>body</span>": ${objeto.body}
    <br>
  }
  <br>`

  return mensagem;
}

async function getPostagem(id) {
  await fetch(`https://jsonplaceholder.typicode.com/posts/${id}`)
    .then((response) => response.json())
    .then((json) => {
      console.log(json);
      respjson.innerHTML = imprime(json);
    });
}
</script>
</head>

```

```

<body>
  <main>
    <h1>Postagens</h1>

    <input id="numero" type="number" min="1" max="100" value="1">
    <button onclick="getPostagem(numero.value)">Buscar</button>

    <div id="resp">
      <p class="campo">Resposta:</p>
      <p id="respjson"></p>
    </div>
  </main>
</body>

</html>

```

2) GET /posts

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>fetch</title>
  <style>
    html,
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      font-size: 16px;
    }

    main {
      width: 51rem;
      margin: 1rem auto;
    }

    button {
      text-align: right;
      padding: 0.5rem;
      margin-bottom: 0.5rem;
    }

    #resp {
      padding: 0.5rem;
      background-color: orangered;
      color: white;
    }
  </style>

```

```

        width: 50rem;
    }

    .campo {
        color: yellow;
        font-weight: bold;
    }
</style>

<script>
    function imprime(objeto) {
        var mensagem;

        objeto.forEach(function (obj) {
            mensagem += `
            <br>
            {
                <br>
                "<span class='campo'>id</span>": ${obj.id},
                <br>
                "<span class='campo'>userId</span>": ${obj.userId},
                <br>
                "<span class='campo'>title</span>": ${obj.title},
                <br>
                "<span class='campo'>body</span>": ${obj.body}
                <br>
            },
            <br>`
        });

        return mensagem;
    }

    async function getPostagens() {
        await fetch(`https://jsonplaceholder.typicode.com/posts`)
            .then((response) => response.json())
            .then((json) => {
                console.log(json);
                respjson.innerHTML = imprime(json);
            });
    }
</script>
</head>

<body>
    <main>
        <h1>Postagens</h1>
    
```

```

<button onclick="getPostagens()">Buscar</button>

<div id="resp">
  <p class="campo">Resposta:</p>
  <p id="respjson"></p>
</div>
</main>
</body>

</html>

```

3) POST /posts

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>fetch</title>
  <style>
    html,
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      font-size: 16px;
    }

    main {
      width: 51rem;
      margin: 1rem auto;
    }

    button {
      text-align: right;
      padding: 0.5rem;
      margin-bottom: 0.5rem;
    }

    #resp {
      padding: 0.5rem;
      background-color: orangered;
      color: white;
      width: 50rem;
    }

    .campo {
      color: yellow;
      font-weight: bold;
    }
  </style>

```

```

    }
</style>

<script>
  function imprime(objeto) {
    var mensagem = `
    <br>
    {
      <br>
      "<span class='campo'>id</span>": ${objeto.id},
      <br>
      "<span class='campo'>userId</span>": ${objeto.userId},
      <br>
      "<span class='campo'>title</span>": ${objeto.title},
      <br>
      "<span class='campo'>body</span>": ${objeto.body}
      <br>
    }
    <br>`

    return mensagem;
  }

  async function postPostagens() {
    fetch('https://jsonplaceholder.typicode.com/posts', {
      method: 'POST',
      headers: {
        'Content-type': 'application/json; charset=UTF-8',
      },
      body: JSON.stringify({
        title: 'Nova Postagem',
        body: 'Essa é a minha mais nova postagem! :D',
        userId: 1,
      }),
    })
      .then((response) => response.json())
      .then((json) => {
        console.log(json);
        respjson.innerHTML = imprime(json);
      });
  }
</script>
</head>

<body>
  <main>
    <h1>Postagens</h1>
  
```

```

<button onclick="postPostagens()">Enviar</button>

<div id="resp">
  <p class="campo">Resposta:</p>
  <p id="respjson"></p>
</div>
</main>
</body>

</html>

```

4) PUT /posts/:id

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>fetch</title>
  <style>
    html,
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      font-size: 16px;
    }

    main {
      width: 51rem;
      margin: 1rem auto;
    }

    button {
      text-align: right;
      padding: 0.5rem;
      margin-bottom: 0.5rem;
    }

    #resp {
      padding: 0.5rem;
      background-color: orangered;
      color: white;
      width: 50rem;
    }

    .campo {
      color: yellow;
    }
  </style>

```



```

        font-weight: bold;
    }
</style>

<script>
    function imprime(objeto) {
        var mensagem = `
        <br>
        {
            <br>
            "<span class='campo'>id</span>": ${objeto.id},
            <br>
            "<span class='campo'>userId</span>": ${objeto.userId},
            <br>
            "<span class='campo'>title</span>": ${objeto.title},
            <br>
            "<span class='campo'>body</span>": ${objeto.body}
            <br>
        }
        <br>`

        return mensagem;
    }

    async function atualizarPostagens() {
        fetch('https://jsonplaceholder.typicode.com/posts/1', {
            method: 'PUT',
            headers: {
                'Content-type': 'application/json; charset=UTF-8',
            },
            body: JSON.stringify({
                id: 1,
                title: 'Minha Mais Nova Postagem',
                body: 'Essa é a atualização da minha mais nova postagem! :D',
                userId: 1,
            }),
        })
        .then((response) => response.json())
        .then((json) => {
            console.log(json);
            respjson.innerHTML = imprime(json);
        });
    }
</script>
</head>

<body>
    <main>

```

```

<h1>Postagens</h1>

<button onclick="atualizarPostagens()">Atualizar</button>

<div id="resp">
  <p class="campo">Resposta:</p>
  <p id="respjson"></p>
</div>
</main>
</body>

</html>

```

5) PATCH /posts/:id

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>fetch</title>
  <style>
    html,
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      font-size: 16px;
    }

    main {
      width: 51rem;
      margin: 1rem auto;
    }

    input[type="number"],
    button {
      text-align: right;
      padding: 0.5rem;
    }

    input[type="number"] {
      width: 10rem;
      margin-bottom: 0.5rem;
    }

    button {
      margin-bottom: 0.5rem;
    }
  </style>

```

```

#resp {
  padding: 0.5rem;
  background-color: orangered;
  color: white;
  width: 50rem;
}

.campo {
  color: yellow;
  font-weight: bold;
}
</style>

<script>
function imprime(objeto) {
  var mensagem = `
  <br>
  {
    <br>
    "<span class='campo'>id</span>": ${objeto.id},
    <br>
    "<span class='campo'>userId</span>": ${objeto.userId},
    <br>
    "<span class='campo'>title</span>": ${objeto.title},
    <br>
    "<span class='campo'>body</span>": ${objeto.body}
    <br>
  }
  <br>`

  return mensagem;
}

async function modificarPostagens(id) {
  fetch(`https://jsonplaceholder.typicode.com/posts/${id}`, {
    method: 'PATCH',
    headers: {
      'Content-type': 'application/json; charset=UTF-8',
    },
    body: JSON.stringify({
      title: 'Essa é a modificação da minha mais nova postagem! :D',
    }),
  })
  .then((response) => response.json())
  .then((json) => {
    console.log(json);
    respjson.innerHTML = imprime(json);
  })
}

```

```

    });
  }
</script>
</head>

<body>
  <main>
    <h1>Postagens</h1>

    <input id="numero" type="number" min="1" max="100" value="1">
    <button onclick="modificarPostagens(numero.value)">Modificar</button>

    <div id="resp">
      <p class="campo">Resposta:</p>
      <p id="respjson"></p>
    </div>
  </main>
</body>

</html>

```

6) DELETE /posts/:id

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>fetch</title>
  <style>
    html,
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      font-size: 16px;
    }

    main {
      width: 51rem;
      margin: 1rem auto;
    }

    input[type="number"],
    button {
      text-align: right;
      padding: 0.5rem;
    }
  </style>

```

```

input[type="number"] {
  width: 10rem;
  margin-bottom: 0.5rem;
}

button {
  margin-bottom: 0.5rem;
}

#resp {
  padding: 0.5rem;
  background-color: orangered;
  color: white;
  width: 50rem;
}

.campo {
  color: yellow;
  font-weight: bold;
}
</style>

<script>
function imprime(objeto) {
  var mensagem = `
  <br>
  {
    <br>
    "<span class='campo'>id</span>": ${objeto.id},
    <br>
    "<span class='campo'>userId</span>": ${objeto.userId},
    <br>
    "<span class='campo'>title</span>": ${objeto.title},
    <br>
    "<span class='campo'>body</span>": ${objeto.body}
    <br>
  }
  <br>`

  return mensagem;
}

async function removerPostagem(id) {
  fetch(`https://jsonplaceholder.typicode.com/posts/${id}`, {
    method: 'DELETE',
  });
}
</script>

```

```

</head>

<body>
  <main>
    <h1>Postagens</h1>

    <input id="numero" type="number" min="1" max="100" value="1">
    <button onclick="removerPostagem(numero.value)">Remover</button>

    <div id="resp">
      <p class="campo">Resposta:</p>
      <p id="respjson"></p>
    </div>
  </main>
</body>

</html>

```

7) Filtrar

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>fetch</title>
  <style>
    html,
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      font-size: 16px;
    }

    main {
      width: 51rem;
      margin: 1rem auto;
    }

    input[type="number"],
    button {
      text-align: right;
      padding: 0.5rem;
    }

    input[type="number"] {
      width: 10rem;
    }
  </style>

```

```

        margin-bottom: 0.5rem;
    }

    button {
        margin-bottom: 0.5rem;
    }

    #resp {
        padding: 0.5rem;
        background-color: orangered;
        color: white;
        width: 50rem;
    }

    .campo {
        color: yellow;
        font-weight: bold;
    }
</style>

<script>
    function imprime(objeto) {
        var mensagem;

        objeto.forEach(function (obj) {
            mensagem += `
                <br>
                {
                    <br>
                    "<span class='campo'>id</span>": ${obj.id},
                    <br>
                    "<span class='campo'>userId</span>": ${obj.userId},
                    <br>
                    "<span class='campo'>title</span>": ${obj.title},
                    <br>
                    "<span class='campo'>body</span>": ${obj.body}
                    <br>
                },
                <br>`
        });

        return mensagem;
    }

    async function filtrarPostagens(id) {
        fetch(`https://jsonplaceholder.typicode.com/posts?userId=${id}`)
            .then((response) => response.json())
            .then((json) => {

```

```

        console.log(json);
        respjson.innerHTML = imprime(json);
    });
}
</script>
</head>

<body>
    <main>
        <h1>Postagens</h1>

        <input id="usuario" type="number" min="1" max="10" value="1">
        <button onclick="filtrarPostagens(usuario.value)">Filtrar</button>

        <div id="resp">
            <p class="campo">Resposta:</p>
            <p id="respjson"></p>
        </div>
    </main>
</body>

</html>

```

Exercícios

Usando HTML, CSS, JS e Bootstrap:

- 1) A partir dos exemplos apresentados, crie **uma única** página Web que permita o usuário ler, postar, atualizar, modificar, remover e filtrar postagens. **OBS:** usar o Bootstrap.
- 2) Usando a API do ViaCep (<https://viacep.com.br/>), crie uma página que permita o usuário realizar consultas por CEP. **OBS:** usar o Bootstrap.
- 3) Usando a API DummyJSON (<https://dummyjson.com/>), crie uma página Web que permita o usuário realizar consultas. **OBS:** usar o Bootstrap.
- 4) Usando "The Star Wars API" (SWAPI) (<https://swapi.dev/>), crie uma página Web que permita o usuário realizar consultas. **OBS:** usar o Bootstrap.
- 5) Pesquise e Escreva sobre o cliente HTTP Axios:
<https://axios-http.com/docs/intro>

Referências:

https://developer.mozilla.org/pt-BR/docs/Web/API/Fetch_API/Using_Fetch

<https://jsonplaceholder.typicode.com/>

<https://jsonplaceholder.typicode.com/guide/>

<https://viacep.com.br/>

<https://dummyjson.com/>

<https://swapi.dev/>

<https://pipedream.com/apps/swapi>

<https://axios-http.com/docs/intro>