AXIOS

FAPESC – DESENVOLVEDORES PARA TECNOLOGIA DA INFORMAÇÃO

HERCULANO DE BIASI
herculano.debiasi@unoesc.edu.br

Tópicos

- Axios
- Exemplo

Axios

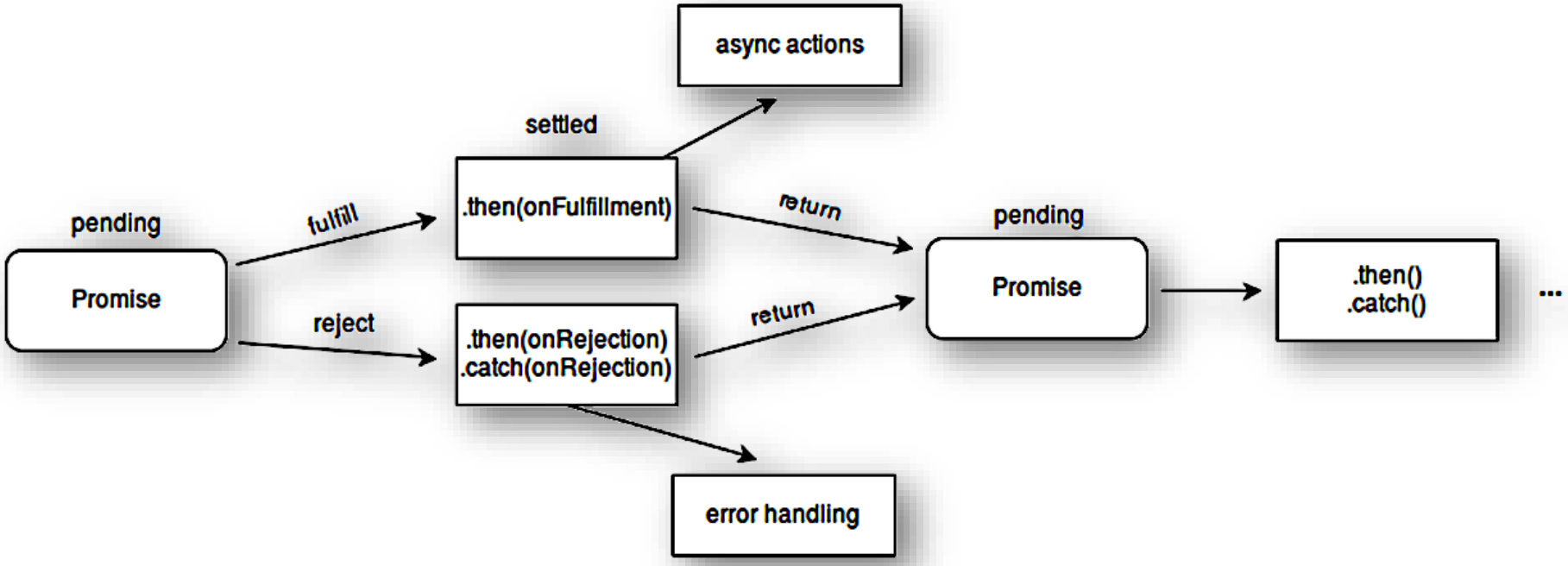
- Axios é um cliente HTTP baseado em *promises* para fazer requisições e que pode ser utilizado tanto no navegador quando no Node.js

Axios

- Funcionalidades
  - Por baixo dos panos faz requisições Ajax no navegador via XMLHttpRequests
  - Faz requisições HTTP no Node.js
  - Tratamento simples de erros
  - Suporte para navegadores mais antigos
  - Suporta a API de *promises*
  - Intercepta requisições e respostas (*request* & *response*)
  - Cancela requisições
  - Transforma os dados em JSON automaticamente
  - No lado cliente suporta a proteção contra XRSF
  - Transforma os dados da requisição e da resposta

Axios

- Funcionamento de uma promessa



Axios

- Para adicionar Axios em algum projeto pode-se usar `npm`, `yarn` ou `bower`
  - `npm install axios`
  - `bower install axios`
  - `yarn add axios`

- Também é possível utilizar um CDN `jsDelivr` ou `unpkg`
  - `<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>`
  - `<script src="https://unpkg.com/axios/dist/axios.min.js"></script>`

EXEMPLO

- Servidor (*back-end*)

```
const express = require('express');
const cors = require('cors');
const formData = require('express-form-data');

const app = express();
const porta = 8082;

// Configura o CORS
app.use(cors({ origin: '*' }));

// Configura o tratamento das requisições POST
app.use(express.urlencoded({extended: true}));
app.use(express.json());

// Faz o parsing no formato multipart
app.use(formData.parse());

// Instancia o servidor
app.listen(porta,
    () => console.log(`Servidor iniciado na porta: ${porta}`)
);

// Responde a requisição no endereço http://localhost:8080/
app.get('/', (request, response) => {
    response.status(200).send('<h2>Servidor node.js :-)</h2>');
});
```

UNOESC

EXEMPLO

- Servidor (*back-end*)

```
28  // Tratamento de requisição GET convencional
29  app.get('/axios', function(req, res) {
30      const n1 = parseInt(req.query['n1']);
31      const n2 = parseInt(req.query['n2']);
32
33      console.log('axios - get:', n1, n2);
34
35      res.send({ resultado: n1 + n2 });
36  });
37
38  // Versão alternativa de tratamento da requisição GET
39  app.get('/axiosParams', function(req, res) {
40      const n1 = parseInt(req.query.n1);
41      const n2 = parseInt(req.query.n2);
42
43      console.log('axiosParams - get:', n1, n2);
44
45      res.send({ resultado: n1 + n2 });
46  });
```

UNOESC

EXEMPLO

- Servidor (*back-end*)

```
48  // Tratamento de requisição POST convencional
49  app.post('/axios', function(req, res) {
50      const n1 = parseInt(req.body.n1);
51      const n2 = parseInt(req.body.n2);
52
53      console.log('axios - post:', n1, n2);
54
55      res.send({ resultado: n1 + n2 });
56  });
57
58  // Tratamento de POST no formato multipart/form-data
59  app.post('/axiosFD', function(req, res) {
60      const n1 = parseInt(req.body.n1);
61      const n2 = parseInt(req.body.n2);
62
63      console.log('axiosFD - post:', n1, n2);
64
65      res.send({ resultado: n1 + n2 });
66  });
```

UNOESC

EXEMPLO

- Servidor (*back-end*)

```
68  // Tratamento de requisição PUT
69  app.put('/axios', function(req, res) {
70      const operacao = req.body.operacao;
71
72      console.log('axios - put:', operacao);
73
74      res.send({ resultado: operacao + ' efetuada com sucesso!' });
75  });
76
77  // Tratamento de requisição DELETE
78  app.delete('/axios', function(req, res) {
79      const operacao = req.body.operacao;
80
81      console.log('axios - delete:', operacao);
82
83      res.send({ resultado: operacao + ' efetuada com sucesso!' });
84  });
```

EXEMPLO

- Cliente (*front-end*)

```html
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/all.min.css"
    crossorigin="anonymous">

    <title>Axios</title>
</head>
```

EXEMPLO

- Cliente (*front-end*)

```
<body>
    <div class="container">
        <form method="post" action="#" class="was-validated" id="formulario">
            <h2 class="alert alert-info mt-2">Calculando com Axios</em></h2>

            <div class="form-group mb-3">
                <label for="n1">Número 1:</label>
                <input type="number" name="n1" id="n1" value="40" class="form-control" placeholder="Entre um número" required>
                <small class="form-text text-muted">Digite um número</small>
            </div>

            <div class="form-group mb-3">
                <label for="n2">Número 2:</label>
                <input type="number" name="n2" id="n2" value="2" class="form-control" placeholder="Entre um número" required>
                <small class="form-text text-muted">Digite um número</small>
            </div>

            <div class="form-group mb-3 text-center">
                <button type="submit" class="btn btn-primary">
                    <i class="fa-solid fa-right-to-bracket"></i>
                    Entrar
                </button>
            </div>
        </form>

        <div id="get"></div>
        <div id="post"></div>
        <div id="put"></div>
        <div id="delete"></div>
    </div>
```

EXEMPLO

- Cliente (*front-end*)

```
48      <script>
49          const formulario = document.getElementById('formulario');
50
51          function exibirResultado(id, dados) {
52              const resultado = JSON.stringify(dados);
53              document.getElementById(id).innerHTML = id.toUpperCase() + ': ' + resultado;
54          }
55
56          // Intrercepta o evento de submissão do formulário
57          formulario.addEventListener('submit', async e => {
58              // Evita o comportamento padrão de submissão de formulário
59              e.preventDefault();
60
61              // Cria um objeto do tipo FormData a partir dos dados do formulário
62              const dadosForm = new FormData(formulario);
63              console.log(dadosForm);
64
65              // Recupera os campos de formulário de nome n1 e n2
66              const n1 = dadosForm.get('n1');
67              const n2 = dadosForm.get('n2');
68              console.log(n1, n2);
```

EXEMPLO

- Cliente (*front-end*)

```
70          // Padrão GET passando os dados através da URL
71          axios(`http://localhost:8082/axios?n1=${n1}&n2=${n2}`)
72              .then(res => exibirResultado('get', res.data.resultado) );
73
74          // GET com parâmetros
75          const res = await axios.get('http://localhost:8082/axiosParams', {
76                                      params: { n1, n2 }
77                                  });
78          exibirResultado('get', res.data.resultado);
79
80          // POST
81          // axios.post('http://localhost:8082/axios', { n1, n2 })
82          //      .then(res => exibirResultado('post', res.data.resultado) );
```

EXEMPLO

- Cliente (*front-end*)

```
80          // POST
81          axios.post('http://localhost:8082/axios', { n1, n2 })
82              .then(res => exibirResultado('post', res.data.resultado) );
83
84          // POST - FormData
85          const ret = await axios.post('http://localhost:8082/axiosFD',
86                      dadosForm,
87                      // {
88                      //     headers: {
89                      //         'Content-Type': 'multipart/form-data'
90                      //     }
91                      // }
92                  )
93          exibirResultado('post', ret.data.resultado);
```

EXEMPLO

- Cliente (*front-end*)

```
95              // PUT
96              axios.put('http://localhost:8082/axios', {
97                  operacao: 'Alteração'
98              })
99              .then(res => exibirResultado('put', res.data.resultado));
100
101             // DELETE
102             axios.delete('http://localhost:8082/axios', {
103                 data: { operacao: 'Exclusão' }
104             })
105             .then(res => exibirResultado('delete', res.data.resultado));
106         });
107     </script>
108
109     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" crossorigin="anonymous">
110     </script>
111 </body>
112 </html>
```