estgoh
Politécnico de Coimbra

PI II / DW II

2. SPA com *JavaScript vanilla*

Nuno Miguel Gil Fonseca
nuno.fonseca@estgoh.ipc.pt

SPA com *JavaScript vanilla*

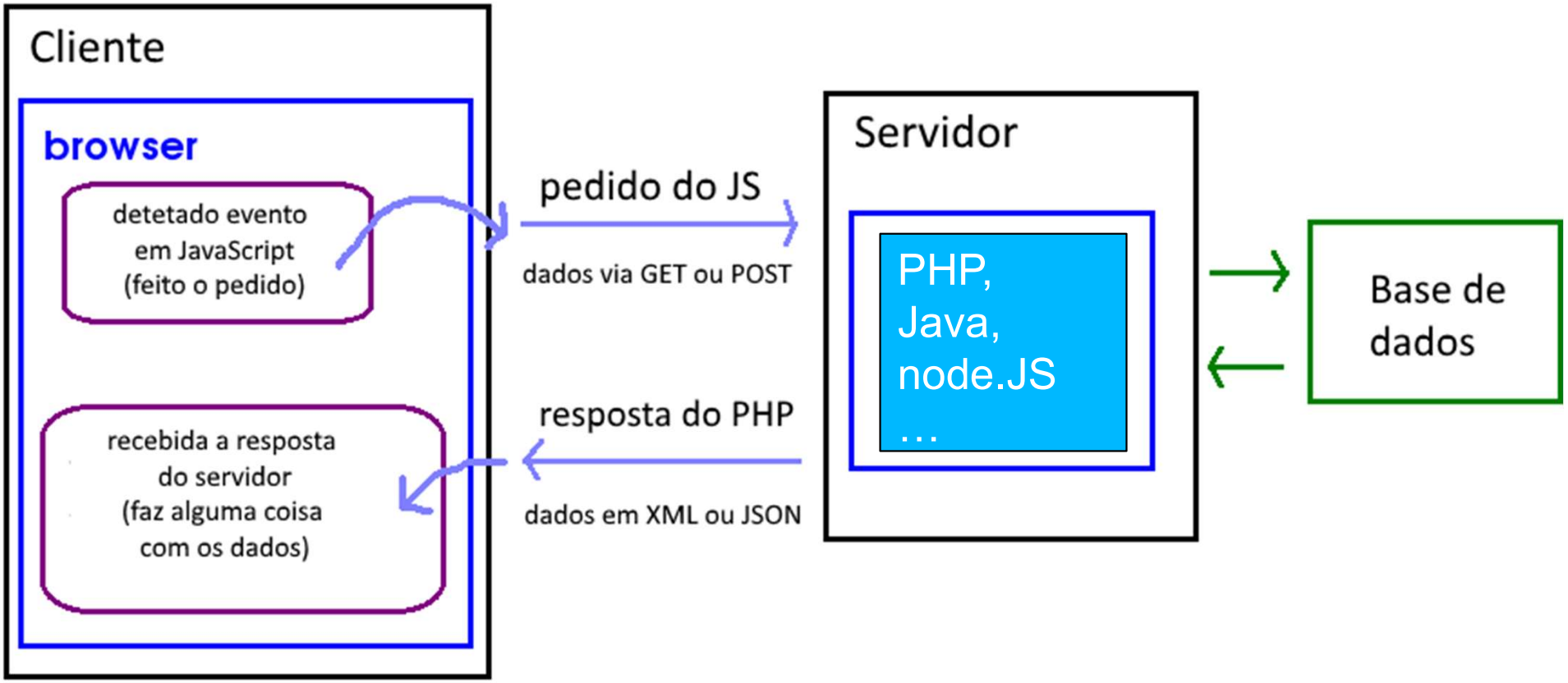
- Durante anos a linguagem *JavaScript* esteve limitada a manipular apenas os dados que possuía no lado do cliente
- Carregar novos dados para utilizar com *JavaScript* era apenas possível mediante o *reload* da página
- Este cenário mudou com o surgimento do objeto `XMLHttpRequest` que os *browsers* passaram a implementar e que potenciou o surgimento de:

SPA com *JavaScript vanilla*

- AJAX = *Asynchronous Javascript And Xml*
  - Ajax não é uma tecnologia! É um conceito!
  - Consiste no uso do objeto `XMLHttpRequest` para fazer pedidos a um servidor através de *JavaScript*.
  - É extremamente útil quando se pretende carregar apenas partes de uma página, validar formulários sem abandonar a página, etc.
  - Envolve sempre programação do lado do cliente (*JavaScript*) e do lado do servidor (PHP, por exemplo).

SPA com *JavaScript vanilla*

- **AJAX**



SPA com *JavaScript vanilla*

- **AJAX**
  - No lado do cliente, começa-se por criar uma instância do objeto XMLHttpRequest (`http`).

```
var http = new XMLHttpRequest();
```

  - Todos os *browsers* modernos têm suporte para a utilização direta do objeto `XMLHttpRequest`.
  - Caso a nossa aplicação tenha que suportar *browsers* muito antigos, seria necessário mais algum código.

  - Muito embora, ainda seja possível receber as respostas do servidor na forma de XML, atualmente as respostas são regra geral recebidas na forma de JSON (http://json.org/)

SPA com *JavaScript vanilla*

- **AJAX**
  - **Para os próximos exemplos foi criado um servidor que permite editar a lista de prendas de Natal de um utilizador (uid)**
  - **GET (http://gilito.com.pt/natal/get_gifts.php?uid=1)**
    - Devolve um JSON contendo:
      - um valor **`err`** (0 - OK, 1 – Não OK)
      - um valor **`err_txt`** (vazio se OK, descrição do erro se não OK)
      - um valor **`gifts`** (contendo a lista dos presentes caso não ocorra nenhum erro)

```
{"err":0,"err_txt":"","gifts":[]}
```

```
{"err":0,"err_txt":"","gifts":[{"g_id":"1","u_id":"1","g_name":"iPhone 14 XPTO GTI","g_done":"0"}]}
```

```
{"err":1,"err_txt":"UID not found"}
```

SPA com *JavaScript vanilla*

- **AJAX**
  - **POST (http://gilito.com.pt/natal/add_gifts.php?uid=1)**
    - Devolve um JSON contendo:
      - um valor **`err`** (0 - OK, 1 – Não OK)
      - um valor **`err_txt`** (vazio se OK, descrição do erro se não OK)
  - **POST (http://gilito.com.pt/natal/add_gifts_json.php?uid=1)**
    - Devolve um JSON contendo:
      - um valor **`err`** (0 - OK, 1 – Não OK)
      - um valor **`err_txt`** (vazio se OK, descrição do erro se não OK)

```
{"err":1,"err_txt":"UID not found"}
```

```
{"err":0,"err_txt":"","gifts":[]}
```

Numa situação mais real, seria utilizada uma API REST.
Contudo, no essencial o modo de funcionamento é igual
Mais adiante iremos

SPA com *JavaScript vanilla*

- **AJAX**

Nota: Nestes exemplos o *output* surge na consola do *browser*

  - Cliente - GET

```
<input type="text" id="uid" placeholder="User ID">
<input type="button" id="button_get" value="GO">
<script>
    const http = new XMLHttpRequest();
    const baseURL = 'http://gilito.com.pt/natal/';
    document.getElementById('button_get').onclick = function(){
        var data = "uid=" + encodeURIComponent(document.getElementById('uid').value);
        //o pedido que será enviado
        http.open('GET', baseURL + 'get_gifts.php?' + data); //get_gifts.php?uid=1
        http.setRequestHeader('Accept', '*/*'); // CORS
        // o que fazer com a resposta
        http.onload = function() {
            if (http.readyState === 4 && http.status === 200) {
                let res = JSON.parse(http.responseText); // res é um array
                if(res.err == 0){
                        console.table(res);
                } else{
                        console.error(res.err_txt);
                }
            } else if (http.status !== 200) { // o que fazer em caso de erro
                console.error('Falhou! - ' + http.status);
          }
        };
       http.send(); // a enviar o pedido
    };
</script>
```

SPA com *JavaScript vanilla*

- **AJAX**
  - **Cliente - POST**

Nota: Nestes exemplos o *output* surge na consola do *browser*

```
<input type="text" id="uid" placeholder="User ID">
<input type="text" id="gift_name" placeholder="Gift name">
<input type="button" id="button_add" value="Add Gift">
<script>
    const http = new XMLHttpRequest();
    const baseURL = 'http://gilito.com.pt/natal/';
    document.getElementById('button_add').onclick = function(){
        var data = "uid=" + encodeURIComponent(document.getElementById('uid').value);
        let name = document.getElementById("gift_name").value;
        let dataPOST = "gift_name=" + encodeURIComponent(name);
        http.open('POST', baseURL + 'post_gifts.php?' + data); //post_gifts.php?uid=1
        http.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
        http.setRequestHeader('Accept', '*/*'); // CORS
        http.onload = function() {// o que fazer com a resposta
            if (http.readyState === 4 && http.status === 200) {
                let res = JSON.parse(http.responseText); // res é um array
                if(res.err == 0){ console.table(res); }
                else{ console.error(res.err_txt); }
            } else if (http.status !== 200) { // o que fazer em caso de erro
                console.error('Falhou! - ' + http.status);
            }
        };
       http.send(dataPOST); // a enviar o pedido
    };
  </script>
```

SPA com *JavaScript vanilla*

- **AJAX**

Nota: Nestes exemplos o *output* surge na consola do *browser*

  - **Cliente - POST (JSON)**

```
<input type="text" id="uid" placeholder="User ID">
<input type="text" id="gift_name" placeholder="Gift name">
<input type="button" id="button_add" value="Add Gift">
<script>
    const http = new XMLHttpRequest();
    const baseURL = 'http://gilito.com.pt/natal/';
    document.getElementById('button_add').onclick = function(){
        var data = "uid=" + encodeURIComponent(document.getElementById('uid').value);
        let dados = {};
        dados.gift_name = document.getElementById("gift_name").value;
        var dadosJSON = JSON.stringify(dados);
        http.open('POST', baseURL + 'post_gifts_json.php?' + data); //post_gifts_json.php?uid=1
        http.setRequestHeader('Content-type','application/json; charset=utf-8')
        http.setRequestHeader('Accept', '*/*'); // CORS
        http.onload = function() {// o que fazer com a resposta
        if (http.readyState === 4 && http.status === 200) {
                let res = JSON.parse(http.responseText); // res é um array
                if(res.err == 0){ console.table(res); }
                else{ console.error(res.err_txt); }
            } else if (http.status !== 200) { // o que fazer em caso de erro
                console.error('Falhou! - ' + http.status);
            }
        };
       http.send(dadosJSON); // a enviar o pedido
    };
  </script>
```

SPA com *JavaScript vanilla*

- **AJAX**
  - **Cliente - Restantes métodos**
    - DELETE (http://gilito.com.pt/natal/delete_gifts.php?uid=1&gid=7)
      - Devolve um JSON contendo:
        - um valor `err` (0 - OK, 1 – Não OK)
        - um valor `err_txt` (vazio se OK, descrição do erro se não OK)
    - PATCH (http://gilito.com.pt/natal/patch_gifts.php?uid=1&gid=7)
      - Devolve um JSON contendo:
        - um valor `err` (0 - OK, 1 – Não OK)
        - um valor `err_txt` (vazio se OK, descrição do erro se não OK)

SPA com *JavaScript vanilla*

- **AJAX + Fetch API**

Nota: Nestes exemplos o *output* surge na consola do *browser*

  - **Cliente - GET**

```
<input type="text" id="uid" placeholder="User ID">
<input type="button" id="button_get" value="GO">
<script>
    const http = new XMLHttpRequest();
    const baseURL = 'http://gilito.com.pt/natal/';
    document.getElementById('button_get').onclick = function(){
      var url = new URL(baseURL + 'get_gifts.php');
      var params = {uid: document.getElementById('uid').value}
      url.search = new URLSearchParams(params).toString();
      fetch(url)
      .then((res) => res.json())
      .then(function (data){
          // os dados estão no array data
          if(data.err == 0){
            console.table(data);
          } else{
            console.error(data.err_txt);
          }
      })
      .catch((error) => console.log(error))
   };
  </script>
```

SPA com *JavaScript vanilla*

- **AJAX + Fetch API**
  - **Cliente - POST**

```
<input type="text" id="uid" placeholder="User ID">
<input type="text" id="gift_name" placeholder="Gift name">
<input type="button" id="button_add" value="Add Gift">
<script>
    const baseURL = 'http://gilito.com.pt/natal/';
    document.getElementById('button_add').onclick = function(){
        var url = new URL(baseURL + 'post_gifts.php');
        var params = {uid: document.getElementById('uid').value}
        url.search = new URLSearchParams(params).toString();
        fetch(url, {
                method : 'POST',
                mode:    'cors',
                headers: {
                    'Accept':'*/*', 'Content-Type': 'application/x-www-form-urlencoded'
                },
                body: "gift_name=" + encodeURIComponent(document.getElementById("gift_name").value)
            })
            .then((res) => res.json())
            .then(function (data){
                // os dados estão no array data
                if(data.err == 0){ console.table(data); }
                else{ console.error(data.err_txt); }
            })
            .catch((error) => console.log(error));
    };
  </script>
```

SPA com *JavaScript vanilla*

- **AJAX + Fetch API**
  - **Cliente - POST (JSON)**

```
<input type="text" id="uid" placeholder="User ID">
<input type="text" id="gift_name" placeholder="Gift name">
<input type="button" id="button_add" value="Add Gift">
<script>
    const baseURL = 'http://gilito.com.pt/natal/';
    document.getElementById('button_add').onclick = function(){
        var url = new URL(baseURL + 'post_gifts_json.php');
        var params = {uid: document.getElementById('uid').value}
        url.search = new URLSearchParams(params).toString();
        let dados = {};
        dados.gift_name = document.getElementById("gift_name").value;
        fetch(url, {
                method : 'POST',
                mode:    'cors',
                headers: {
                    'Accept':'*/*',
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify(dados),
            })
            .then((res) => res.json())
            .then(function (data){
                // os dados estão no array data
                if(data.err == 0){  console.table(data); }
                else{ console.error(data.err_txt); }
            })
            .catch((error) => console.log(error));
    };
  </script>
```

- **AJAX + Fetch API**
  - **Cliente - Restantes métodos**
    - DELETE (http://gilito.com.pt/natal/delete_gifts.php?uid=1&gid=7)
      - Devolve um JSON contendo:
        - um valor `err` (0 - OK, 1 – Não OK)
        - um valor `err_txt` (vazio se OK, descrição do erro se não OK)
    - PATCH (http://gilito.com.pt/natal/patch_gifts.php?uid=1&gid=7)
      - Devolve um JSON contendo:
        - um valor `err` (0 - OK, 1 – Não OK)
        - um valor `err_txt` (vazio se OK, descrição do erro se não OK)

SPA com *JavaScript vanilla*

- **AJAX + Fetch API**
  - **Cliente - Autenticação**
    - POST (http://gilito.com.pt/natal/v2/auth_users.php)
      - Deverão ser enviados no corpo do pedido os valores do *username* (variável `user`) e da *password* (variável `pass`).
      - Devolve um JSON contendo:
        - um valor **`err`** (0 - OK, 1 – Não OK)
        - um valor **`err_txt`** (vazio se OK, descrição do erro se não OK)
        - um valor **`user`** (contendo os detalhes do utilizador que se autenticou corretamente, incluindo o token)
      - Todos os pedidos seguintes deverão conter informação sobre o *token* na forma do cabeçalho HTTP Authorization

```
//( ... )
http.open('GET', baseURL + 'v2/get_gifts.php');
http.setRequestHeader("Authorization", localStorage.getItem("token"));
http.setRequestHeader('Accept', '*/*'); // CORS
// ( ... )
```

SPA com *JavaScript vanilla*

- **AJAX + Fetch API**
  - **Cliente - Terminar a sessão**
    - PATCH (http://gilito.com.pt/natal/v2/logout_users.php)
      - Devolve um JSON contendo:
        - um valor **`err`** (0 - OK, 1 – Não OK)
        - um valor **`err_txt`** (vazio se OK, descrição do erro se não OK)

        - Este pedido também tem que conter informação sobre o *token* na forma do cabeçalho HTTP Authorization