

# Programare web

## Laboratorul 07

### 7. Comunicarea asincronă

#### Cuprins

7. Comunicarea asincronă .....	1
7.1. Concepte și tehnologii .....	1
7.2. Structura proiectului.....	1
7.3. Resurse utile .....	2
7.4. Teme .....	2
7.4.1. Tema 1 .....	2
7.4.2. Tema 2 .....	3
7.4.3. Tema 3 .....	4

#### 7.1. Concepte și tehnologii

În acest laborator veți utiliza și extinde serverul web din laboratorul precedent și veți comunica asincron cu serverul prin intermediul AJAX.. De asemenea, veți lucra în continuare cu sistemul de versionare git și cu aplicația Visual Studio Code.

#### 7.2. Structura proiectului

Structura proiectului `proiect1-numeUtilizatorGitHub` de până la acest laborator (inclusiv):

- `[server_web]`
  - `lanseaza_server.bat` (Windows) / `lanseaza_server.sh` (Linux) - lab 06
  - `server_web.py` (Python) / `ServerWeb.java` (Java) - lab 06, actualizat **lab 07**
- `[continut]`
  - `[css]`
    - `stil.css` - lab 04
  - `[imagini]`
    - `logo.png` - lab 03
  - `[js]`
    - `script.js` - lab 05
    - `persoane.js` - **lab 07**
  - `[resurse]`
    - `persoane.dtd` și/sau `persoane.xsd` - lab 01
    - `persoane.xml` - lab 01
    - `utilizatori.json` - **lab 07**
  - `acasa.html` - **lab 07**
  - `desen.html` - lab 03 tema 3, actualizat lab 04, lab 05, **lab07**
  - `despre.html` - lab 02, actualizat lab 03, lab 04, lab 05, **lab07**
  - `favicon.ico` - lab 06

- `index.html` - lab 03, actualizat lab 04, lab 05, **lab07**
- `inregistreaza.html` - lab 03, actualizat lab 04, lab 05, **lab07**
- `invat.html` - lab 05, **lab07**
- `persoane.html` - **lab07**
- `verifică.html` - **lab07**
- `video.html` - lab 03, actualizat lab 04, lab 05, **lab07**

### 7.3. Resurse utile

Pentru detalii privind limbajul CSS puteți să citiți cursurile 4, 5 și 6, sau să consultați documentația următoare:

- Tutorial JavaScript - <http://www.w3schools.com/js/default.asp>
- JavaScript | MDN - <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Tutorial AJAX - [https://www.w3schools.com/xml/ajax\\_intro.asp](https://www.w3schools.com/xml/ajax_intro.asp)

Serverul web din laboratorul precedent:

- Server web Python 2 - <https://edu.tuiasi.ro/mod/resource/view.php?id=32608>
- Server web Python 3 - <https://edu.tuiasi.ro/mod/resource/view.php?id=32614>
- Server web Java - <https://edu.tuiasi.ro/mod/resource/view.php?id=32611>

### 7.4. Teme

#### 7.4.1. Tema 1

Modificați proiectul de pe GitHub astfel încât site-ul să devină un Single Page Application<sup>1</sup>.

Accesul la paginile web ale site-ului se va realiza prin intermediul paginii `index.html`, care va conține scheletul site-ului (tot mai puțin conținutul elementului `<main>`).

Prin intermediul AJAX se va încărca în mod dinamic conținutul paginilor *Acasă*, *Desen*, *Despre HTML*, *Înregistrează*, *Învăță JS*, *Persoane*, *Verifică* și *Video*.

Pentru a realiza acest lucru, continuați proiectul de pe GitHub și efectuați următorii pași (după fiecare pas, dați `git add` și `commit`):

- creați în directorul `continut` un fișier cu numele `acasa.html` în care copiați tot conținutul fișierului `index.html`,
- în fișierul `index.html`, ștergeți tot conținutul elementului `<main>` și adăugați-i atributul `id="continut"`,
- modificați conținutul celorlalte fișiere html astfel încât să rămână în fișierele respective doar conținutul elementului `<main>` (ștergeți tot codul de deasupra tag-ului `<main>` și de după tag-ul `</main>`, inclusiv cele două tag-uri),
- în fișierul `index.html`, adăugați în elementul `<head>` un element `<script>` care să aibă în el o funcție cu numele `schimbaContinut` care primește ca argument un șir de caractere cu numele `resursa` (în funcția respectivă se va folosi AJAX pentru a se face cerere pentru resursa `resursa + '.html'`, iar răspunsul va fi setat ca și conținut al elementului cu id-ul `continut`),
- în fișierul `index.html`, modificați link-urile interne prin înlăturarea atributului `href` și adăugarea atributului `onclick` (valoarea acestui atribut va fi apelul funcției `schimbaContinut`, iar ca argument va fi numele resursei cerute; e.g., `<a onclick="schimbaContinut('acasa') ">Acasă</a>`).

---

<sup>1</sup> Single Page Application - <https://developer.mozilla.org/en-US/docs/Glossary/SPA>

- f. de acum, toate fișierele html exceptând `index.html` vor fi încărcate folosind AJAX și nu accesate direct din browser; utilizatorul nu părăsește pagina `index.html`, ci doar conținutul elementului `<main>` se schimbă,
- g. pentru ca să se încarce conținutul paginii **Acasă** în momentul în care utilizatorul accesează site-ul, trebuie adăugat atributul `onload` la elementul `<body>` astfel: `<body onload="schimbaContinut('acasa')">`,
- h. de asemenea, adăugați la server suport pentru fișiere xml și json prin adăugarea logicii corespunzătoare la header-ul `Content-Type: json → application/json`, `xml → application/xml`.
- i. pentru ca pagina **Învăț JS** să funcționeze cum trebuie, fișierul `script.js` trebuie adăugat după cum este explicat mai jos la Tema 2.

#### 7.4.2. Tema 2

Extindeți proiectul prin adăugarea fișierului `persoane.html` în rădăcina proiectului. Pe această pagină se va permite vizualizarea sub forma unui tabel a persoanelor din fișierul `persoane.xml` creat în primul laborator.

- a. Pentru început, mutați fișierul `persoane.xml` și fișierele `persoane.dtd` / `persoane.xsd` (dacă există), într-un director cu numele resurse din directorul `continut`.
- b. Prin AJAX, la apăsarea butonului `Persoane`, din `index.html` se cere resursa `persoane.html`. După ce s-a primit răspunsul pentru `persoane.html` ar trebui, să se execute un script care să ceară fișierul `persoane.xml`. Problema este că nu se execută niciun cod JavaScript din interiorul fișierului `persoane.html`.
- c. Va trebui să modificați funcția `schimbaContinut` ca să primească încă două argumente (`jsFisier` - reprezintă numele fișierului JavaScript care se va încărca după încărcarea resursei cerute, `jsFuncție` - reprezintă numele funcției care se va apela după ce se încarcă scriptul din `jsFisier`) și să adăugați când se primește răspunsul, după linia

```
document.getElementById("continut").innerHTML = this.responseText;
```

, următorul cod:

```
if (jsFisier) {  
    var elementScript = document.createElement('script');  
    elementScript.onload = function () {  
        console.log("hello");  
        if (jsFuncție) {  
            window[jsFuncție]();  
        }  
    };  
    elementScript.src = jsFisier;  
    document.head.appendChild(elementScript);  
} else {  
    if (jsFuncție) {  
        window[jsFuncție]();  
    }  
}
```

În bara de navigare codul pentru butonul `Persoane` este de forma:

```
<a onclick="schimbaContinut('persoane', 'js/persoane.js', 'incarcaPersoane')">Persoane</a>
```

- d. Conținutul fișierului `persoane.html` va fi doar textul *Se încarcă. Vă rugăm așteptați....*

- e. În fișierul `persoane.js` va exista funcția `incarcaPersoane` care conține logica prin care se face o cerere la server folosind AJAX pentru fișierul `persoane.xml`.
- f. În momentul primirii fișierului XML se vor folosi funcții specifice DOM pentru a parcurge XML-ul (XML DOM<sup>2</sup>) și a genera un tabel HTML.
- g. Tabelul va fi afișat în secțiune în locul mesajului de încărcare și va avea stiluri CSS.

### 7.4.3. Tema 3

Extindeți proiectul prin adăugarea fișierului `verifica.html` în rădăcina proiectului. Pe această pagină se va permite verificarea dacă un nume de utilizator și parola acestuia sunt introduse corect.

- a. Pentru început, creați un fișier `utilizatori.json` în directorul resurse care va conține codul:  

```
[{"utilizator": "test", "parola": "test"}]
```
- b. În fișierul `verifica.html` trebuie să creați de două căsuțe text, un buton și un element în care să se afișeze dacă utilizatorul și parola sunt corecte, și o funcție JavaScript în care să se facă validarea.
- c. La apăsarea butonului, prin intermediul AJAX, se va lua de la server un fișier JSON cu numele `utilizatori.json`, iar prin intermediul JavaScript se va verifica dacă numele utilizatorului se află în lista din JSON și dacă corespunde parola (JavaScript JSON<sup>3</sup>). Folosiți metoda `JSON.parse(text)` pentru a converti un text care reprezintă un JSON la obiect.
- d. Extindeți funcționalitatea formularului din fișierul `inregistreaza.html` creat în laboratorul 3, astfel încât, la apăsarea butonului `Înregistrează`, să se facă o cerere prin metoda HTTP POST pentru resursa `/api/utilizatori`. La server, când se primește o astfel de cerere se interpretează corpul mesajului, informațiile de acolo sunt transformate în obiect JSON, și apoi este actualizat corespunzător fișierul `resurse/utilizatori.json` de la server.  
Atenție! Nu trebuie să existe vreun fișier `/api/utilizatori` la server, ci doar se verifică dacă resursa cerută de client este `/api/utilizatori` și apoi se face procesarea respectivă.

---

<sup>2</sup> XML DOM - [https://www.w3schools.com/xml/xml\\_dom.asp](https://www.w3schools.com/xml/xml_dom.asp)

<sup>3</sup> JavaScript DOM - [https://www.w3schools.com/js/js\\_json.asp](https://www.w3schools.com/js/js_json.asp)