

3. Website Building

Adrian Adiaconitei

LINKAcademy

Objective

- ✓ Obiectul XMLHttpRequest
- ✓ AJAX - JavaScript asincron și XML
- ✓ AJAX - JavaScript asincron și JSON
- ✓ **jQuery**
 - ✓ Introducere, integrare, scop
 - ✓ Utilizarea JQuery pentru gestionarea lui DOM
 - ✓ Evenimentele JQuery
- ✓ **Bootstrap + jQuery**

XML

CE ESTE XML (eXtensible Markup Language) ?

- ✓ este un set de reguli ce permit crearea unor limbaje de transport al datelor.
- ✓ poate fi citit cu ușurință de către oameni și interpretat de către calculatoare
- ✓ derivă din SGML (Standard Generalized Markup Language), o specificație complexă ce reprezintă părintele multor alte limbaje de acest fel (HTML, DocBook etc).
- ✓ XML folosește etichete (tag-uri) și attribute

JSON

JSON (JavaScript Object Notation) este un format folosit pe scară largă pentru reprezentarea informației structurată, derivat din modul de reprezentare a obiectelor în limbajul Javascript.

JSON prezintă următoarele caracteristici:

- ✓ este un format text, ușor de înțeles de către om și ușor de generat/parsat de către aplicații
- ✓ este un format "suplu" comparativ cu XML – cantitatea de informație suplimentară față de cea utilă este mult redusă
- ✓ ca și XML, este un format independent de platformă, ceea ce îl face potrivit pentru schimbul de date între aplicații

Servicii WEB

- ✓ **Serviciile web** sunt, în același mod, aplicații, însă diferența este că funcționalitatea pe care o expun este menită să fie consumată de către alte aplicații, scrise în limbaje de programare diverse și rulând pe platforme potențial diferite.
- ✓ *“A Web service is a software system designed to support interoperable **machine-to-machine interaction** over a network.”*

Servicii WEB

Tehnologii folosite în web services

- ✓ **RPC (Remote Procedure Call)** - este denumirea dată unei familii de tehnologii ce permit apelarea de la distanță a unor funcții expuse de către un server.
 - ✓ **XML-RPC** – folosește formatul XML pentru reprezentarea cererii și a răspunsului
 - ✓ **JSON-RPC** - analog dar folosind JSON
- ✓ **SOAP (Simple Object Acces Protocol)** - protocol complex, urmaș al lui XML-RPC, folosit pentru accesarea serviciilor expuse de o aplicație web:
 - ✓ **SOAP XML** - varianta SOAP clasică, în care XML este folosit pentru specificarea mesajelor și întregului protocol
 - ✓ **SOAPjr** - echivalentul mai recent, bazat pe JSON
- ✓ **REST** - nu este un protocol sau o tehnologie, ci un design pattern - o anumită perspectivă arhitecturală, dar care e folosită pe scară largă în aplicații

HTML vs XML

```
<html>
```

```
<body>
```

```
<h2>Adrian Adiaconitei</h2>
```

```
<p>Link Academy<br>
```

```
https://www.link-academy.com/<br>
```

```
aadiaconitei@gmail.com<br>
```

```
</p>
```

```
</body>
```

```
</html>
```

<https://developers.facebook.com/docs/jobs-xml/specification#validator>

https://help.apple.com/itc/podcasts_connect/#/itcbaf351599

<https://www.bnr.ro/nbrfxrates.xml>

```
<contact>
```

```
<name>Adrian Adiaconitei</name>
```

```
<address>Link Academy</address>
```

```
<web-page>https://www.link-  
academy.com/</web-page>
```

```
<email>aadiaconitei@gmail.com</email>
```

```
</contact>
```

LINK Academy

JSON vs XML

```
{  
  "heroes": [  
    {  
      "name": "Superman",  
      "power": 2000  
    },  
    {  
      "name": "Batman",  
      "power": 1000  
    },  
  ],  
}
```

```
<heroes>  
  <hero>  
    <name>Superman</name>  
    <power>2000</power>  
  </hero>  
  <hero>  
    <name>Batman</name>  
    <power>1000</power>  
  </hero>  
</heroes>
```


XML

- ✓ Un document/fișier XML are extensia .xml
- ✓ Un document/fișier XML este **Case sensitive**
- ✓ Un document XML este format din:
 - ✓ marcaje (tag-uri)
 - ✓ date caracter

Un marcaj (tag) este un șir de caractere delimitat de caracterele "<" și ">". Un fișier XML cuprinde următoarele secțiuni:

- ✓ Prolog (optional , `<?xml version="1.0" encoding="UTF-8"?>`)
 - ✓ Definiția tipului de document (optionala, DTD - un set de reguli ce definește structura unui fișier XML)
 - ✓ Elementul rădăcină (Un document XML are un singur element rădăcină)
- Documente XML well-formed. Documente XML valide

XML

În XML etichetele(**marcaje / tag-uri**) sunt de 2 feluri:

- de început sau de deschidere (open tag): **<post>**, **<companie>**, **<nivel>**
- de final sau de închidere (end tag): **</post>**, **</companie>**, **</nivel>**

Conținutul de tip text al unui element **NU** poate conține caracterele **<** și **>**, aceste caractere trebuie encodate cu **<** respectiv **>**;

Entitate	Referinta la entitate
<	<
>	>
&	&
'	'
"	"e;

XML

- ✓ Sunt două modalități (din punct de vedere al structurii) de a scrie date într-un fisier XML

Exemplul: **Programator PHP Senior**

```
<posturi>
  <post>
    <titlu>Programator PHP</titlu>
    <nivel>Senior</nivel>
  </post>
</posturi>
```

sau

```
<posturi>
  <post
    titlu="Programator PHP"
    nivel="Senior"/>
</posturi>
```

XML

! Invalid

```
<root>  
  <comparison>1>0</comparison>  
</root>
```

✓ Valid

```
<root>  
  <comparison>1>0</comparison>  
</root>
```

XML

! Etichete invalide

- **<2_nume/>** (începe cu cifră)
- **<nume utilizator/>** (conține spațiu)

✓ Etichete valide

- **<nume_2/>**
- **<nume_utilizator/>**

XML

! Invalid

```
<limbaj>
  <nume>PHP</nume>
</limbaj>
<limbaj>
  <nume>Javascript</nume>
</limbaj>
<limbaj>
  <nume>Java</nume>
</limbaj>
```

✓ Valid

```
<limbaje>
  <limbaj>
    <nume>PHP</nume>
  </limbaj>
  <limbaj>
    <nume>Javascript</nume>
  </limbaj>
  <limbaj>
    <nume>Java</nume>
  </limbaj>
</limbaje>
```

JSON

JSON vs XML avantaje:

- ✓ e mai ușor de citit de către programator
- ✓ Conține mai puțină informație suplimentară față de cea utilă, ceea ce are următoarele implicații:
 - ✓ se transmite mai puțină informație prin rețea și eficiența transmisiunilor crește
 - ✓ necesită putere de procesare/memorie mai mică, ceea ce îl face mai potrivit pentru dispozitivele mobile care dispun de resurse limitate
- ✓ format nativ JavaScript - poate fi folosit ca atare de aplicațiile Javascript care rulează în browser și nu numai (vezi cazul AJAX)

JSON

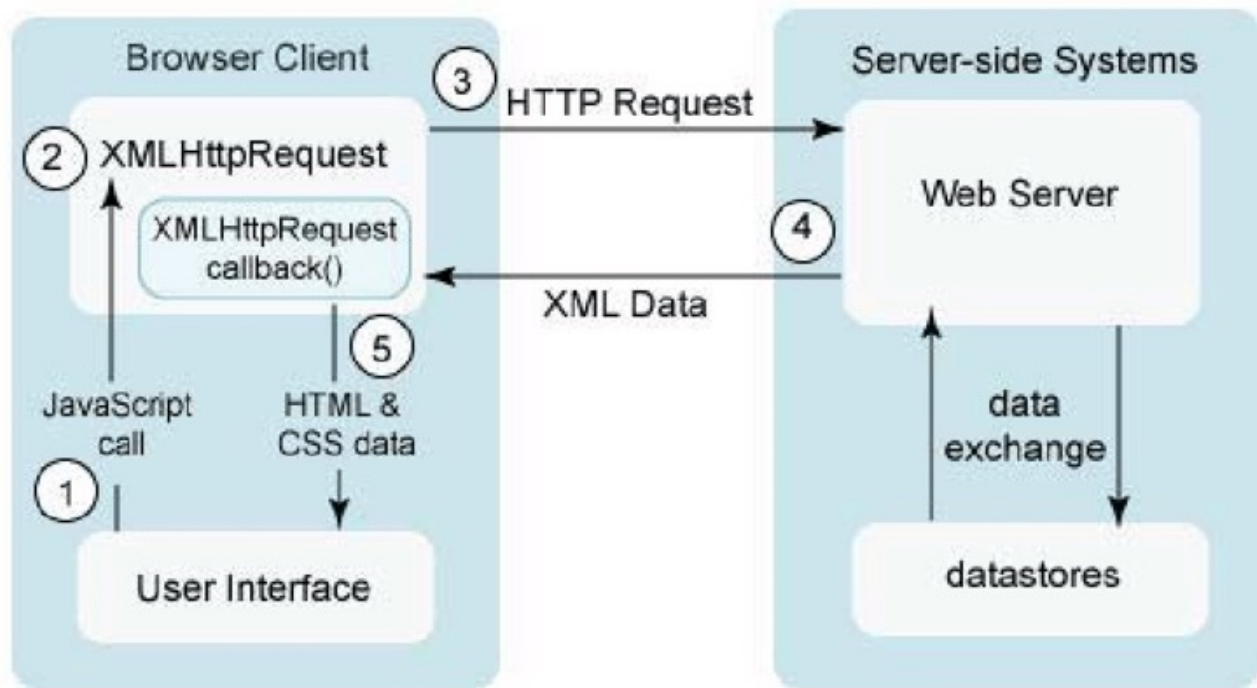
JSON vs XML dezavantaje:

- ✓ sintaxa este mai simplă, suportand doar câteva tipuri de date (pe când în XML se pot crea tipuri de date la nevoie)
- ✓ la decodarea unui obiect JSON nu se cunosc informații despre variabila de origine, pe baza careia a fost generat (array asociativ/ obiect) ,în XML avem numele de elemente ce indică semnificația informației
- ✓ JSON nu beneficiază încă, în mod standardizat, de diversele tehnologii conexe disponibile de mult timp în XML

JavaScript – XMLHttpRequest

- ✓ AJAX - Asynchronous [JavaScript](#) and [XML](#)
- ✓ Toate browserele moderne au un **obiect XMLHttpRequest** încorporat pentru a solicita date de la un server.
- ✓ Obiectul XMLHttpRequest poate fi utilizat pentru a solicita date de la un server web:
 - ✓ Actualizați o pagină web fără a reîncărca pagina
 - ✓ Solicitați date de la un server - după încărcarea paginii
 - ✓ Primiți date de la un server - după încărcarea paginii
 - ✓ Trimiteți date către un server - în fundal
- ✓ https://www.w3schools.com/xml/xml_http.asp

JavaScript – XMLHttpRequest



JavaScript – XMLHttpRequest

https://www.w3schools.com/xml/xml_http.asp

XML

Ap1.html - curs valutar

Ap2.html, ap3.html, ap4.html – colecție Cd-uri

https://www.w3schools.com/xml/ajax_applications.asp

JavaScript – XMLHttpRequest

https://www.w3schools.com/js/js_json_http.asp

JSON

Ap5.html – afișare o imagine

Ap6.html – afișare lista articole

Provocare: ap7.html Faceți o aplicație care afișează toți utilizatorii cu toate datele din acest API:

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

JavaScript – XMLHttpRequest

Aplicația1:

Web Service RESTfull((Representational State Transfer).

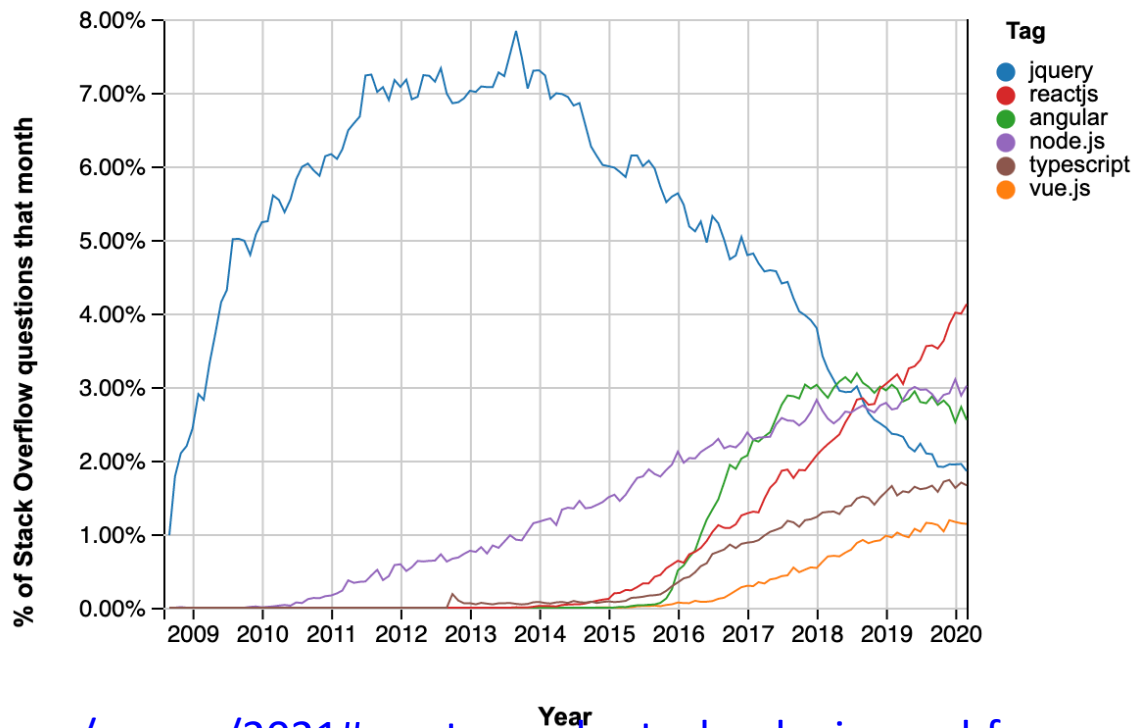
(GET, POST, PUT, DELETE)

jQuery

- ✓ **jQuery** vine de la JavaScript Query
- ✓ **jQuery** este o bibliotecă / librărie JavaScript, cu o istorie bogată, comunitate mare, rapidă, gratis și cu foarte multe caracteristici:
 - ✓ Manipulări DOM
 - ✓ Manipularea evenimentelor
 - ✓ Efecte și tranziții animate
 - ✓ Ajax
- ✓ Publicată în anul 2006 (John Resig)

jQuery

80% din
website-uri
folosesc jQuery



<https://insights.stackoverflow.com/survey/2021#most-popular-technologies-webframe>

<https://survey.stackoverflow.co/2022/#most-popular-technologies-language>

<https://w3techs.com/technologies/comparison/js-angularjs,js-jquery>

jQuery

Cum folosim jQuery?

- ✓ Descărcăm local un fișier cu extensia .js
<https://code.jquery.com/jquery-3.6.0.min.js>
- ✓ Sau includem din CDN (Content Delivery Network)

<https://releases.jquery.com/>

jQuery - Sintaxă

`$(selector).acțiune()`

`$(selector).event(function() {})`

- ✓ *`$` pentru a defini / accesa libraria jQuery*
 - ✓ *`(selector)` pentru a selecta elementul/elementele HTML (la fel ca în CSS)*
 - ✓ *`O acțiune jQuery ()` care trebuie executată pe elementul (elementele) selectat (selectate)*
-
- `$("p").hide()` - toate elementele `<p>` vor fi ascunse (`display:none`).
 - `$(".test").css("background-color", "#cccccc")` - elementele cu `class='test'` vor primi proprietatea `css background-color: "#cccccc"`
 - `$("#test").text()` - returneaza conținutul elementul cu `id="test"`.

jQuery - Sintaxă

ap8.html

Evenimentul **ready** nu este același lucru cu **onload**, deoarece se declanșează mult mai devreme decât onload, care așteaptă ca întregul conținut să se încarce pe pagină, inclusiv imaginile găsite pe pagină. Codul jQuery poate fi executat din momentul în care documentul este gata (ready).

```
$(document).ready( function () {  
    // codul jQuery aici  
});
```

ap9.html

jQuery - Selectori

<https://www.w3schools.com/jquery/tryselect.asp>

- ✓ Selectorii și filtrele sunt unele dintre principalele blocuri de construire a codului jQuery.
- ✓ Selectorii și filtrele se folosesc pentru extragerea informațiilor din pagina web, manipularea DOM-ului
- ✓ Selectorii în biblioteca jQuery folosesc aceeași sintaxă ca în CSS, care este una dintre primele idei pentru crearea unei astfel de biblioteci.
- ✓ Filtrele sunt folosite pentru o căutare suplimentară în rezultatul pe care îl returnează selectorii (p, .selector, #intro, p:first, h2:not (.selector)).

jQuery - Selectori

Selector	Rezultat
<code>\$("nume_tag")</code>	Selectează toate tagurile cu numele specificat.
<code>\$("#id_element")</code>	Selectează elementul cu id-ul specificat.
<code>\$(".nume_clasă")</code>	Selectează toate elementele clasei specificate.

Care sunt funcțiile echivalente pentru cei 3 selectori în Java Script? (Vanilla.js)

ap10.html

jQuery - Evenimente

Numele evenimentului se folosește ca nume al metodei pentru setarea unui listener (ascultător, consumator) de eveniment asupra elementului. Parametru acestei metode al doilea este funcția callback care va fi apelată în momentul în care se va detecta evenimentul asupra elementului.

Mouse Events	Keyboard Events	Form Events	Document / Window Events
click	keypress	submit	onload
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	ready

jQuery – Metode / Funcții

✓ **Adaugă / șterge clasă**

(manipularea DOM)

```
$("#panel1").addClass("nume-class")
```

```
$("#panel1").removeClass("nume-class")
```

```
$("#panel1").toggleClass("nume-class")
```

✓ **Verifică dacă un element are class:**

```
$("#panel1").hasClass("nume-class"));
```

✓ **Adaugă stil**

```
$("#panel1").css({color:'red',fontSize:'16px'})
```

✓ **Modificarea atributelor**

```
attr()
```

```
removeAttr()
```

jQuery – Metode / Funcții

jQuery poate crea un conținut nou sau poate schimba proprietățile existente.

- ✓ Metodele pentru poziționarea unui conținut nou în elementul selectat:
 - ✓ **.append()**
 - ✓ **.prepend()**
 - ✓ **.before()**
 - ✓ **.after()**
- ✓ Metodele pentru adăugarea unui HTML sau conținut text unui element:
 - ✓ **.html()**
 - ✓ **.text()** – Conținutul transmis nu va fi interpretat ca HTML
 - ✓ **.val()**

jQuery – Metode / Funcții

- ✓ jQuery permite înlănțuirea funcțiilor (Statement Chaining).
- ✓ Înlănțuirea comenzilor înseamnă că asupra elementului selectat (sau asupra șirului de elemente) se pot apela mai multe funcții în aceeași linie.
- ✓ Funcțiile din lanț se execută pe rând (de la stânga la dreapta), una câte una.
- ✓ Sintaxa este următoarea: **`$(selector).fn1().fn2().fn3();`**

ap11.html

JS Method
Chaining



jQuery -Efecte / Animații

- ✓ .show()
- ✓ .hide()
- ✓ .toggle()
- ✓ .fadeIn()
- ✓ .fadeOut()
- ✓ .slideUp()
- ✓ .slideDown()
- ✓ .slideToggle()

jQuery - Efecte / Animații

ap12.html

```
$("#panel1").hide(500).show(1000).hide(500).show(1000);
```

```
$("#panel2").fadeIn(200).fadeOut(200).fadeIn(200).fadeOut(200).fadeIn(200).fadeOut(200).fadeIn(200).fadeOut(200);
```

jQuery - Efecte / Animații

- ap13.html

Pornind de la ap12.html: faceți 4 butoane și 4 div-uri.

Când se dă click pe butonul1 să dispară/apară div-ul 1

Când se dă click pe butonul2 să dispară/apară div-ul 2

Când se dă click pe butonul3 să dispară/apară div-ul 3

Când se dă click pe butonul4 să dispară/apară div-ul 4

.....

jQuery

Dacă dorim să nu scriem același cod de mai multe ori și avem mai multe butoate, cum optimizăm ap13.html?

ap14.html optimizare ap13.html

Aplicația2: **Lista de cumparaturi cu jQuery.**

Provocare: Lista de cumpărături să nu dispară la **F5**
Refresh

Bootstrap

Aplicația: Shop:

- ✓ Optimizare carousel – imagini încărcate în funcție de dispozitiv
- ✓ Modal
- ✓ Pe baza unui cookie să afișăm un mesaj de salut: Bine ai veni / revenit
- ✓ Darck mode
- ✓ Sectiune noua cu 6 produse: Ultimele produse
- ✓ Adăugare funcționalitate: Add to cart
- ✓ Load page
- ✓ Personalizam aplicația folosind .scss
- ✓ Google captcha <https://developers.google.com/recaptcha/docs/v3>

Resurse

<https://www.freecodecamp.org/news/how-javascript-implements-oop/>

<https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>

https://www.w3schools.com/js/js_ajax_intro.asp

<https://cdnjs.com/libraries>

<https://www.npmjs.com/>

<https://nodejs.org/en/docs/>

<https://www.typescriptlang.org/docs/home.html>

<https://v8.dev/>

<https://babeljs.io/docs/en/>

<https://webplatform.github.io/docs/apis/xhr/XMLHttpRequest/>

<https://www.w3schools.com/xml/>

<https://www.freeformatter.com/xsd-generator.html>

<https://xmltoolbox.appspot.com/>