Ideea generală a proiectului

- Se va demonstra capacitatea de a transfera date între mai multe servere, în mai multe formate, folosind **mecanismul HTTP**
- Se va implementa o succesiune de transferuri demonstrabile printr-un front-end rudimentar (în browser)
- Se va lucra cu următoarele surse de date (vezi arhitectura de pe slideul următor):
    - **Web scraping** (de pe orice site)
    - **RDF4J**
    - La alegere între **JSON-Server** și **JSON-GraphQL-Server** (nu se vor implementa altele server NodeJS, se va folosi exact interfața oferită de aceste instrumente!)

Comportamentul paginii:

(0) Starea inițială: o pagină HTML cu butoane pentru declanșarea pe rând a transferurilor de date (vezi mockup pe slide-ul următor)

Pas (1) Butonul 1 de pe slideul următor

- declanșează Web scraping de pe un site oarecare ales de student
- afișează datele obținute într-un tabel (vezi conținutul datelor pe un slide următor)

Pas (2-3) Butonul 2 de pe slideul următor

- preia o înregistrare de la un formular HTML din pagină, o adaugă la cele obținute la pasul anterior
- inserează tot setul de date pe Server 1
- solicită tot ce conține Server 1 după inserare și afișează într-un tabel nou

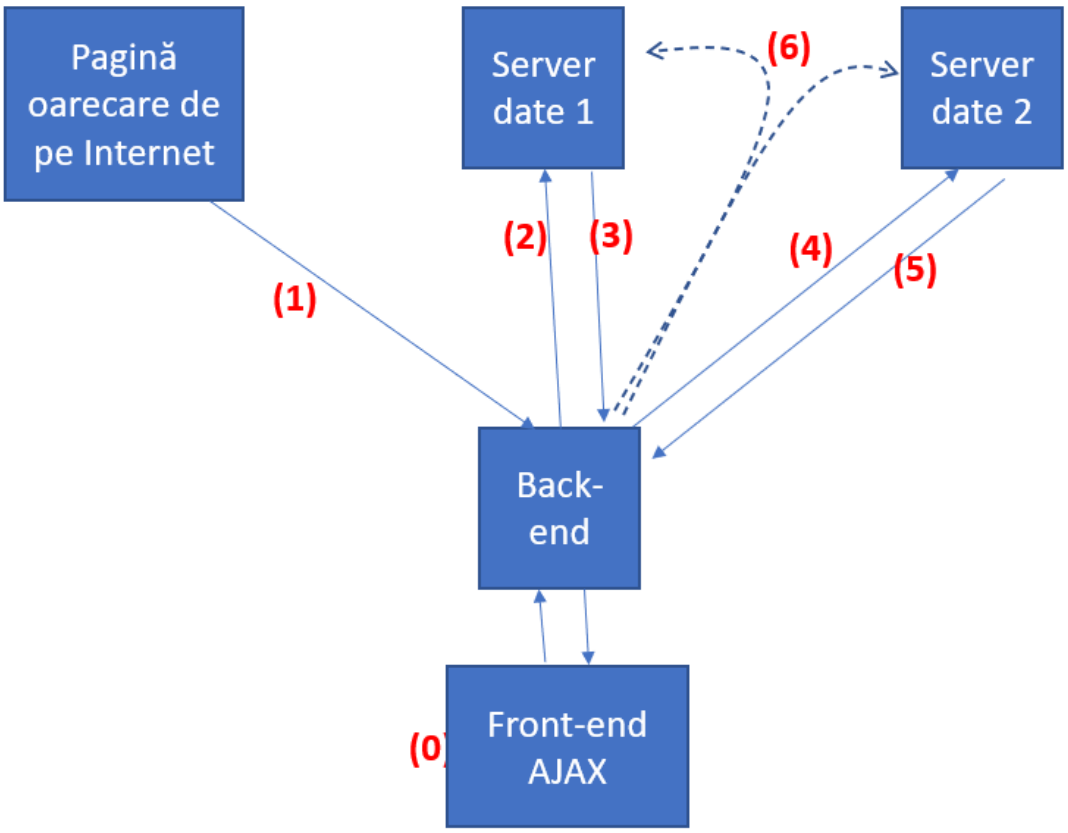
Pas (4-5) Butonul 3 de pe slideul următor

- preia datele afișate de pasul precedent
- le inserează mai departe pe Server 2
- solicită tot ce conține Server 2 după inserare și afișează într-un tabel nou

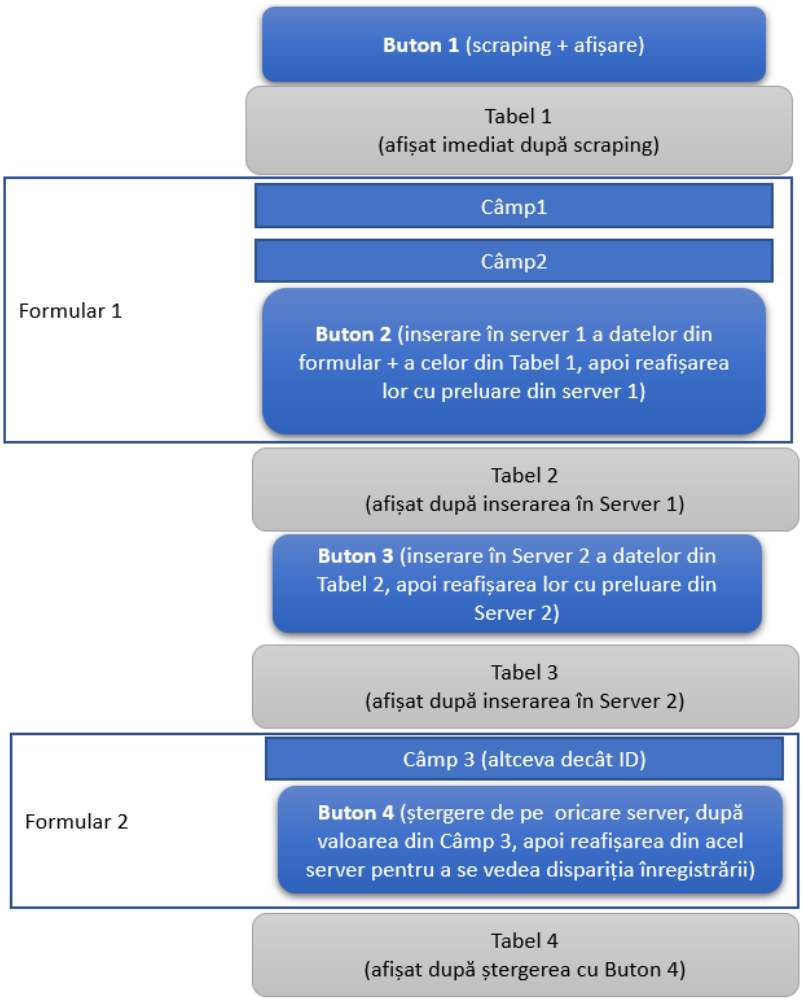
Pas (6) Butonul 4 de pe slideul următor

- șterge o înregistrare din oricare server (1 sau 2), ștergerea se va face după un câmp preluat din pagină și care să nu fie ID

Arhitectura



Mockup


Buton 1 (scraping + afișare)
Tabel 1
(afișat imediat după scraping)
Formular 1
Câmp1
Câmp2
Buton 2 (inserare în server 1 a datelor din formular + a celor din Tabel 1, apoi reafișarea lor cu preluare din server 1)
Tabel 2
(afișat după inserarea în Server 1)
Buton 3 (inserare în Server 2 a datelor din Tabel 2, apoi reafișarea lor cu preluare din Server 2)
Tabel 3
(afișat după inserarea în Server 2)
Formular 2
Câmp 3 (altceva decât ID)
Buton 4 (ștergere de pe oricare server, după valoarea din Câmp 3, apoi reafișarea din acel server pentru a se vedea dispariția înregistrării)
Tabel 4
(afișat după ștergerea cu Buton 4)

Restricții tehnologice

- Pasul de scraping se va face **prin Xpath, folosind alt criteriu decât ID-uri**
    - un scraping mai rudimentar se acceptă dar cu penalizare (ex. bazat pe ID sau CSS cu o librărie de scraping HTML)
- Unul din cele două servere va fi **RDF4J**, al doilea va fi un server JSON (**la alegere între JSON-Server și JSON-GraphQL-Server**)
    - e irelevantă ordinea între servere (ex. RDF4J poate fi Server 1 sau Server 2)
    - pasul de citire din RDF4J va insera rezultatele nu doar în corpul HTML ci și ca bloc JSON-LD
    - serverul JSON nu va putea fi pornit fără o înregistrare de inițializare (va fi creată manuală și preluată în afișările ulterioare)
- Limbaje de programare folosite
    - Implementarea back-end se poate face în **orice limbaj de programare**
    - Implementarea front-end va fi **JavaScript**, putându-se folosi și frameworkuri JavaScript/TypeScript (React, Angular, Vue etc.)

Conținutul setului de date

Setul de date culese prin Web scraping = echivalentul unui tabel cu

- 2 câmpuri de tipuri diferite (altele decât ID, care va fi gestionat intern de scripturi/server)
- 3 înregistrări

Formular 1 (de după Tabel 1) va conține casete corespunzătoare celor 2 câmpuri, deci va adăuga o a 4-a înregistrare

Formular 2 (de după Tabel 3) va conține o casetă corespunzătoare câmpului după care se face ștergerea (să nu fie ID)

La momentul accesării serverului JSON (indiferent că e JSON-Server sau JSON-GraphQL-Server) va mai apărea măcar încă o înregistrare căci acele servere trebuie pornite cu un dataset de inițializare (se va crea manual măcar o înregistrare .json pentru a putea porni serverul)

Bonusuri

- Dacă pentru cererile HTTP se folosesc alte librării/metode decât cele din seminarii
    - deci altele decât xhr, fetch, jquery, curl, guzzle, file_get_contents, easyRDF Http Client
- Dacă în faza de scraping se aduc și date găsite undeva în format JSON-LD
    - acestea vor fi afișate ca tabel distinct lângă Tabel 1 și nu vor interfera cu restul proiectului

nu trebuie să provină de pe același site pe care se face și Xpath

Setări de respectat!!!

- Porturi de folosit:
    - JSON-Server = localhost:4000
    - JSON-GraphQL-Server = localhost:3000

    - RDF4J = localhost: 8080
        - baza de grafuri să se numească grafexamen
- Codul sursă să nu conțină comentarii!