

## **TEMA 1 DATC**

### **REST**

REST( REpresentational State Transfer) e un stil architectural ce furnizează standarde între calculatoare, făcând comunicarea între acestea mai ușoară. Sistemele REST-compliant sunt adesea numite sisteme RESTful și sunt caracterizate prin faptul că sunt stateless și clientul și server-ul nu depinde unul de celălalt.

În stilul architectural REST, implementarea client-ului și server-ului poate fi făcută independent fără ca unul să știe de celălalt. În felul acesta, codul din client poate fi schimbat oricând fără a afecta operațiile server-ului și invers. Atât timp cât fiecare parte știe ce format al mesajului să transmită, poate fi păstrată modulară și separată.

Separând interfața grafică de stocarea datelor, putem îmbunătăți flexibilitatea pe toate platformele și scalabilitatea simplificând componentele server-ului.

Un sistem stateless înseamnă că server-ul nu trebuie să știe starea client-ului și vice versa. În acest fel, atât server-ul cât și client-ul pot înțelege orice mesaj primit, chiar fără să analizeze mesajele anterioare. Constrângerea stateless e realizată prin folosirea resurselor. Resursele sunt substantivele din WEB, ele pot descrie orice poate fi salvat sau transmis către alte servicii.

Comunicarea dintre client și server se realizează prin transmiterea de request-uri. Client-ul trimite un request pentru a primi sau modifica o resursă, iar server-ul răspunde la acest request. Un request este alcătuit din:

- Un verb HTTP care definește tipul acțiunii ce urmează a fi efectuată
- Un header care îi permite client-ului să transmită informații despre request.
- O cale către resursă
- Un mesaj( opțional) ce conține date

Cele 4 verbe ce respectă paradigma CRUD sunt:

- GET – folosit pentru a cere o resursă sau o colecție de resurse
- POST – creează o nouă resursă
- PUT – face update unei resurse
- DELETE – șterge o resursă identificată printr-un id

În cazul în care server-ul transmite date către client, el trebuie să includă și tipul conținutului în header-ul răspunsului. Tipul conținutului avertizează client-ul asupra conținutului din corpul mesajului transmis. Tipul transmis trebuie să fie unul din tipurile specificate de client în request. Răspunsul trimis de către server conține un cod status pentru a informa client-ul despre starea operației. Aceste coduri pot fi:

- 1XX: request-ul a fost primit, procesarea continuă
- 2XX: request-ul a fost primit, înțeles și acceptat
- 3XX: sunt necesare acțiuni următoare pentru a putea termina request-ul
- 4XX: request-ul conține erori sau nu s-a putut îndeplini
- 5XX: server-ul nu a putut îndeplini request-ul cerut.