

TEMA 1 DATC - REST API

În general, un **API(Application Programming Interface)** prevede o interfață între două sisteme. API-urile care utilizează protocolul HTTP sunt „servicii web”. Un „serviciu web” este o aplicație bazată pe web care oferă resurse într-un format consumabil de către alt computer. Serviciile Web includ diverse tipuri de API-uri inclusiv **REST API**(Representational State Transfer). Serviciile Web sunt în esență interacțiuni de solicitare și răspuns între clienți și servere(un computer face cererea pentru o resursă, iar serviciul web oferă răspunsul). Clientul sau server-ul pot avea la bază orice limbaj, dar **HTTP** este protocolul utilizat pentru a transporta mesajul. Acest model de solicitare-și-răspuns este modul fundamental în care acționează REST.

Un alt aspect unic al REST-ului este că API-urile **se concentrează pe resurse**(mai precis lucruri, nu acțiuni) și **modalități de accesare a resurselor**. Resursele sunt de obicei tipuri diferite de informații. Acestea sunt accesate prin adrese **URL**(Uniform Resource Locators). Adresele URL sunt însoțite de o metodă care specifică modul în care dorim să interacționăm cu resursa.

Metodele obișnuite includ: GET(read), POST(create), PUT(update) și DELETE(remove). Sfarșitul include, de obicei, parametrii de interogare care specifică mai multe detalii despre reprezentarea resursei pe care dorim să o vedem. De exemplu, putem specifica limitarea afișării a 5 instanțe ale resursei (Exemplu: <http://apiserver.com/homes?limit=5&format=json>).

Serverul nu stochează nicio stare despre sesiunea clientului . Această restricție se numește **statelessness**. Fiecare cerere de la client la server trebuie să conțină toate informațiile necesare pentru a înțelege cererea și nu poate profita de niciun context stocat pe server. Starea sesiunii este, prin urmare, păstrată în întregime pe client. Clientul este responsabil pentru **stocarea** și **manipularea** tuturor informațiilor legate de aplicație(pe partea de client). Statelessness înseamnă că fiecare cerere HTTP are loc în izolare completă. Atunci când clientul face o solicitare HTTP, acesta include toate informațiile necesare serverului pentru a îndeplini această solicitare. Serverul nu se bazează niciodată pe informații din cereri anterioare. Cu alte cuvinte, de fiecare dată când accesezi o resursă, API-ul oferă același răspuns și **nu există stări anterioare** pe care API-ul să le ia în considerare la fiecare cerere. Această proprietate ajută la scalarea API-urilor la milioane de utilizatori concurenți prin implementarea acestora pe mai multe servere. Orice server poate gestiona orice solicitare deoarece **nu există nicio dependență între sesiuni**.

De asemenea, răspunsurile pot fi stocate în **cache** pentru a crește performanța. Dacă memoria cache a browser-ului conține deja informațiile solicitate în request, browserul poate să returneze pur și simplu informațiile din memoria cache în loc să primească din nou resursa de pe server. Proprietatea de **caching** a REST API-urilor este similară cu caching-ul paginilor web. Browserul utilizează valoarea ultimei modificări din antetele HTTP pentru a determina dacă are nevoie să obțină din nou resursa. Dacă conținutul nu a fost modificat de la ultima dată de când a fost preluat, poate fi utilizată în schimb copia memorată în cache. Acest lucru crește viteza răspunsului. Deoarece caching-ul la nivel de rețea este utilizat în mare măsură pentru creșterea throughput-ului, este logic să ținem în cache răspunsuri chiar și pe perioade scurte de timp.