

Tema DATC

În acest rezumat vom discuta în linii mari despre paradigma CRUD (Create, Read, Update, Delete). Acest model ne sugerează funcționalitatea de bază a unei aplicații, și anume posibilitatea de a crea, actualiza, a citi și șterge resurse dintr-o anumită locație prin request-uri către server. Un astfel de sistem trebuie să aibă capacitatea de a efectua aceste 4 operațiuni pentru a fi complet. Tiparul CRUD este foarte întâlnit în construirea de aplicații deoarece oferă o structură, o fundație sau un cardu pentru dezvoltatori.

Prin exemplu, explicația CRUD este aceasta: C (Create) – inserarea unei înregistrări într-o tabelă (cu diverse câmpuri) dintr-o bază de date. R (Read) – operațiunea de a obține toate sau o parte din înregistrările dintr-un tabel (se pot pune diverse condiții de filtrare a rezultatelor). U (Update) – procesul de schimbare/modificare a unor informații pentru anumite înregistrări existente deja. D (Delete) – posibilitatea de ștergere a unor înregistrări.

Mergând pe această idee, în tehnologiile web, se poate discuta despre o structură standard a unui request de tip HTTP (Hyper Text Transfer Protocol). Această comunicare în HTTP sugerează un ciclu de solicitare (request) ↔ răspuns (response). Pentru un request valid, un client trebuie să includă 4 lucruri în structura acestuia:

- URL (Uniform Resource Locator): adresele URL sunt o modalitate ușoară pentru client de a spune serverului cu ce resursă vrea să interacționeze.
- Method: indică serverului ce fel de acțiune dorește clientul să folosească. Cele mai întâlnite metode observate în API-uri sunt: Post (Create), Get (Read), Put (Update), Delete.
- Headers: oferă meta-informații despre request. Exemple: „Authorization”, „Accept-Charset”, „Content-Type”, „Cache-Control” etc.
- Body: conține datele pe care clientul dorește să le trimită serverului. Cele mai comune formate găsite în API-uri sunt JSON și XML. JSON este un format foarte simplu, care are două elemente: cheie și valoare. XML oferă câteva blocuri de construcție. Blocul principal se numește un nod. XML începe întotdeauna cu un nod rădăcină, în interiorul căruia sunt mai multe noduri "copil". Numele nodului ne spune atributul ordinului, iar datele din interior sunt detaliile reale.

Răspunsul dat de server are o structură asemănătoare cu cea a request-ului. Conține Status Code (clase de coduri, de la 1xx până la 5xx), Headers și Body.

Și pentru că am amintit de despre autorizare ca și element al header-urilor din structura requestului, este foarte important să protejăm diferite informații confidențiale, sau accesibile doar pentru anumiți utilizatori. Acest lucru se poate realiza printr-un token de autentificare. Autentificarea bazată pe token este o tehnică de securitate care autentifică utilizatorii care încearcă să obțină acces la un server. Serviciul validează token-ul de securitate și procesează solicitarea utilizatorului. În calitate de dezvoltator, se pot utiliza token-uri pentru operațiuni CRUD complete. După ce se generează un token, el poate fi folosit pentru a autoriza solicitările individuale făcute de utilizatori. Aplicația va valida token-ul trimis împreună cu fiecare solicitare. Pentru fiecare utilizator se generează un api-key și un api-secret care împreună formează un token. Api-key identifică utilizatorul, iar api-secret validează cererea (request-ul).