

Tema DATC

REST

REST sau Representational State Transfer este pe scurt un stil arhitectural folosit in crearea sau construirea de servicii Web. Acest stil este foarte popular datorita simplității dar si pentru ca se bazează pe sisteme si tehnologii deja caracteristice protocolul HTTP. Site-uri populare care sunt construite pe modelul REST sunt: Yahoo, Facebook, LinkedIn dar si multe altele . REST se bazează pe un set reguli numite si constrângeri. REST ca si principiu poate fi cel mai ușor înțeles prin intermediul constrângerilor pe care le folosește.

Setul de constrângeri pe care se bazează sunt:

Client Server – In spatele acestei constrângeri sta principiul separării preocupărilor. Constrângerea Client server înseamnă de fapt separarea preocupărilor interfeței utilizatorului de preocupările modului în care se stochează datele. Deci clientul si serverul reprezintă doua entități total diferite din punct de vedere a responsabilităților fiecăruia.

Stateless – Aceasta constrângere spune ca fiecare request de la client către server trebuie sa conțină toate informațiile necesare pentru ca serverul sa reușească sa înțeleagă in totalitate request-ul fără sa acceseze vreo posibila informație stocata deja in server care ar putea avea legătură cu request-ul clientului.

Cache – Constrângerea Cache spune ca informațiile dintr-un răspuns provocat de un request trebuie sa fie etichetata in mod implicit sau explicit ca cacheable sau non-cacheable. Daca un răspuns este notat ca fiind cacheable atunci acesta are dreptul de refolosi informația venita cu acel răspuns, rezultând astfel request-uri echivalente.

Uniform Interface – Aceasta constrângere reprezintă caracteristica care face diferența între Rest și alte stiluri arhitecturale. Uniform Interface se rezuma la faptul ca REST accentuează faptul ca trebuie sa exista o interfață uniforma între componente. De asemenea prin aplicarea principiului generalității interfeței componentelor se simplifica arhitectura sistemului dar si interacțiunile din sistem reies mai vizibil.

Layered System – Constrângerea Layered System lasă ca sistemul sa fie alcătuit din straturi ierarhice, după cum ii spune si numele, care sunt limitate la faptul ca nu pot accesa sau „vedea” dincolo de propria existenta. Deci fiecare strat are independenta proprie.

Code-On-Demand – Este singura constrângere ce poate fi considerata opțională. Aceasta permite funcționalității sistemului sa fie extinsa prin script-uri sau applet-uri. Astfel se îmbunătățește extensibilitatea sistemului pentru ca acesta poate descarcă si executa alt cod chiar si după ce a fost implementat.