

Nume: Tamas Horvath

An: 4 IS

Grupa: 3.1

Tema 1 DATC: Richardson maturity model

Modelul de maturitate Richardson este o modalitate de a imparti pe nivele un API in functie de constrangerile REST. Cu cat API-ul evaluat respecta mai bine cerintele modelului, cu atat scorul este mai mare. Modelul lui Richardson este folosit frecvent in industrie, fiind un model ce respecta cu strictete constrangerile REST pentru a evalua API-ul din cat mai multe puncte de vedere. Cateva dintre aceste constrangeri sunt: nu trebuie să depindă de niciun protocol de comunicare, nu trebuie să definească nume de resurse fixe sau ierarhii sau lipsa cunostiintelor anterioare exceptand URI-ul initial.

In cadrul acestui model exista 4 nivele; atingerea nivelului 3 aduce API-ul intr-un punct in care poate fi considerat cu adevarat RESTful. Cele 4 nivele sunt:

- nivelul 0 – The Swamp of POX

Acesta este punctul de pornire al modelului, care este caracterizat prin utilizarea HTTP ca sistem de transport pentru interactiunea la distanta, fara utilizarea mecanismelor avansate web. Cu alte cuvinte la acest nivel HTTP este folosit ca un mecanism de “tunneling” cu un singur punct de intrare(URI) in combinatie cu o singura o metoda HTTP(de exemplu POST sau GET).

Un exemplu de request ar putea fi: GET <http://UPT.ro/facultati>, prin care ne vor fi returnati toti studentii din facultate.

- nivelul 1 – Resources

La acest nivel, comunicarea ajunge sa fie intre elemente, ajungand ca API-ul tau sa distinga resursele unele de altele. Acest nivel ajunge sa foloseasca mai multe URI-uri, astfel fiecare devenind un punct de intrare separat pentru fiecare resursa.

Astfel, in loc sa parcurgem datele prin <http://UPT.ro/facultati>, avem acces la fiecare element in parte (ex: <http://UPT.ro/facultati/AC>, <http://UPT.ro/facultati/ETC>) acestea fiind considerate resurse de sine statatoare.

- nivelul 2 – HTTP verbs

La nivelul 2, nu mai folosim verbele HTTP pentru simplul proces de “tunneling”, ajungand sa le utilizam similar cu felul in care sunt ele folosite in HTTP. In acest moment, intelegem fiecare verb HTTP(GET, PUT, POST, DELETE) si utilitatea acestora, astfel avand posibilitatea de a le folosi in functie de nevoi. In acelasi timp, la acest nivel trebuie sa ne folosim si de coduri astfel incat sa returnam valorile respective in conformitate cu evenimentul petrecut in aplicatie(ex: nu returnam 200(OK), in momentul in care nu gasim datele cautate, in loc de 404(Not Found)).

- nivelul 3 – Hypermedia controls

Ultimul nivel se refera la introducerea conceptului de HATEOAS(Hypertext As The Engine Of Application State) API-ul ajungand sa se foloseasca de toate serviciile puse la dispozitie de protocolul HTTP. In acest moment, API-ul ar trebui sa ajute utilizatorul prin descoperirea tuturor resurselor asociate datelor din request, facand acest lucru prin intermediul legaturilor disponibile din cadrul datelor.

Astfel, ideile abordate de fiecare nivel ar fi:

- nivelul 1: manipularea complexității prin utilizarea divide et impera, impartind o singura resursa masiva in mai multe resurse mai mici.
- nivelul 2: introduce un set standard de verbe, astfel încât să rezolvăm situații similare în același mod.
- nivelul 3: ofera o modalitate unui protocol de a se auto-documentat.