{iT}Factory

Intalnirea 9

Testarea de API

{iT}Factory

Obiective Intalnire 9

- Sa aflam ce e un API si cum functioneaza
- Sa intelegem rolul testarii in calitatea unui API
- Sa diferentiem intre Json si XML
- Sa deosebim HTML si HTTP
- Sa stim care sunt codurile de raspuns HTTP
- Sa aflam ce este Postman si care sunt componentele lui

Ce inseamna API?

- **API este prescurtarea de la *Application Programming Interface*** si reprezinta un set de proceduri, funcții și alte elemente pe care un un sistem le pune la dispoziție pentru a facilita comunicarea cu un alt sistem.
- În general se poate decide implementarea unui API atunci cand avem nevoie sa transmitem un volum mai mare de date dintr-o aplicație web către un sistem extern, insa nu este necesara atunci cand vrem sa creăm spre exemplu doar un site de prezentare.

Exemplu

Sa presupunem ca mergem la un restaurant. Nu exista niciun chelner prin preajma, asa ca ne uitam peste meniu si apoi mergem la bucatarie sa cerem preparatul pe care il dorim. Dar daca acesta nu este disponibil va trebui sa ne intoarcem la masa si sa ne decidem asupra altui preparat si apoi sa reluam procesul.

Pe de alta parte, bucatarul va primi cereri nu doar de la noi, ci si de la ceilalti clienti care sunt in restaurant, si in scurt timp va fi coplesit de informatii si va ajunge in incapacitatea de a prepara materialele intr-un timp rezonabil.

De asemenea, daca intram intr-un restaurant in care bucatarul nu vorbeste o limba pe care sa o intelegem, ne lovim de bariera comunicarii.

In schimb, daca in restaurant exista un chelner, toate aceste probleme ar fi rezolvate (APROAPE intotdeauna)

Din punct de vedere tehnic, rolul unui API este similar cu rolul unui chelner.

Testare API

- Testarea de API este o testare care are loc la un nivel foarte timpuriu, pentru a beneficia de avantajul găsirii timpurii a defectelor, chiar înainte ca GUI-ul sa fie creat (principiul 3 al testarii, *Early Testing*)
- Un alt avantaj al testarii de API este ca putem executa anumite teste care pot fi imposibil de testat din cauza restrictiilor de GUI (ex: putem sa testam niste valori dintr-un dropdown care sunt imposibil de ales din GUI, in eventualitatea in care din cauza unei erori de sistem valoarea respectiva ar ajunge sa fie trimisa spre procesare)
- Exista mai multe aplicatii pe piata pentru testarea de API, printre care cele mai cunoscute sunt **Postman** si **SOAP UI**

API Testing vs Unit Testing

- Testarea de API se face de regula atunci cand avem un output final al sistemului, si atunci cand doua sisteme sunt deja integrate, pentru a testa comunicarea dintre ele, in schimb ce testarea unitara se poate face pe o bucata foarte mica dintr-o functionalitate (ex: functii, variabile)
- Testarea de API este directionata mai degraba catre verificarea comportamentului de business si a arhitecturii de sistem, in schimb ce testarea unitara verifica functionarea fiecarei functionalitati individuale, izolate de celelalte functionalitati/componente
- Testarea de API este in general de catre testeri, in schimb ce testarea unitara este de regula facuta de catre dezvoltatori

{iT}Factory

Abordarea Testarii de API

- Atunci cand facem testare de API, trebuie sa stim sa interpretam in primul rand erorile care pot aparea, care pot fi cauzate de erori umane (date incorecte introduse), erori generate de produs (bug-uri in aplicatie) sau erori generate de server (fisiere corupte, parametri de conectare incorecti, incapacitatea de procesare a unor date etc)
- Atunci cand o aplicatie este dezvoltata, este recomandat ca testarea sa fie desfasurata concomitent, astfel incat problemele sa pot sa fie descoperite in timp util
- La fel ca orice alt tip de testare, testarea de API va acoperi atat scenarii pozitive cat si scenarii negative, si se va baza pe toate tehnicile si tipurile de testare care au fost invatate pana acum

Responsabilitati si Cunostiinte Necesare ale unui tester de API

- Necesitatea cunoasterii notiunii de Web Service pentru diverse categorii cum ar fi *REST, SOAP* and *Micro Services* (Macar la nivel conceptual)
- Cunoasterea metodelor principale de HTTP, cum ar fi GET, POST, PUT, PATCH,DELETE
- Cunoasterea codurilor de raspuns HTTP pentru validarea raspunsului din punctul de vedere al codului, mesajului si al timpului de raspuns
- Cunoasterea conceptelor de XML si JSON pentru a putea defini corpul requesturilor de API
- Cunoasterea conceptuala a notiunii de OAuth
- Capacitatea de a intelege documentatia de API si de a scrie test case-uri plecand de la aceasta
- Capacitatea de a putea scrie instructiuni de baza de SQL pentru a putea verifica rezultatul request-urilor de API

{iT}Factory

XML vs JSON

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<root>
    <student>
        <id>01</id>
        <name>Tom</name>
        <lastname>Price</lastname>
    </student>
    <student>
        <id>02</id>
        <name>Nick</name>
        <lastname>Thameson</lastname>
    </student>
</root>
```

```json
{
    "root": {
        "student": [
            {
                "id": "01",
                "name": "Tom",
                "lastname": "Price"
            },
            {
                "id": "02",
                "name": "Nick",
                "lastname": "Thameson"
            }
        ]
    }
}
```

Mecanisme OAuth

Un **open protocol** permite autentificarea sigură a aplicațiilor de pe web, mobile sau desktop.

Open Authentication (OAuth) este un protocol de autorizare de tip open-standard care furnizeaza aplicațiilor capabilitatea de a oferi si cere acces în mod sigur.

De exemplu, o aplicație poate sa informeze aplicația Facebook că este ok ca o anumită aplicație sa acceseze profilul sau sa posteze actualizari pe timeline fără sa mai ceara parola, acest lucru minimizand riscul prin faptul ca, in cazul in care aplicatia este compromisa, accesul la parola va ramane restrictionat.

Mecanismul OAuth nu furnizeaza datele legate de parola, în schimb folosește tokenuri de autentificare pentru a putea dovedi identitatea dispozitivului de pe care se cere accesul (consumer) și furnizorul de servicii (service provider). OAuth este un protocol de autentificare care permite comunicarea dintre doua aplicații în numele utilizatorului fără ca acesta sa trebuiasca sa furnizeze parola.

Exista trei piloni principali în tranzacțiile de tipul OAuth: utilizatorul, consumatorul și furnizorul de servicii.

{iT}Factory

Coduri de răspuns HTTP

Codurile de răspuns HTTP sunt o serie de numere care definesc dacă request-ul care a fost facut de catre client (calculator, telefon, desktop, tableta etc) a fost efectuat cu succes sau nu.

Codurile de status furnizate de server sunt trimise în completarea mesajului de răspuns trimis la fiecare request al clientului.

Codurile de HTTP sunt împărțite în cinci categorii. Prima cifră a codului de răspuns HTTP definesc în care dintre cele cinci clase se încadrează răspunsul, iar ultimele doua cifre definesc semnificația răspunsului.

Conform standardelor IANA, următoarele sunt categoriile de coduri care pot fi returnate de catre protocolul HTTP:

- 1xx - informational
- 2xx - Success
- 3xx - Redirectare
- 4xx - Eroare de Client
- 5xx - Eroare de Server

POSTMAN - Introducere

Postman este un software independent care este folosit pentru crearea, design-ul, modificarea si testarea de API.

Este un GUI simplu folosit pentru trimiterea si vizualizarea requesturilor si raspunsurilor HTTP.

Prin intermediul aplicatiei Postman putem sa trimitem un request de API catre server si sa vizualizam raspunsul detaliat cu scopul analizei si evaluarii acestuia.

Este folosit foarte mult de catre testeri si developeri pentru o acoperire mai buna a aplicatiei.

Descarcarea si instalarea aplicației POSTMAN

1. Acceseaza site-ul https://www.postman.com/downloads/ si alege sistemul de operare al laptopului/desktop-ului pe care lucrezi. Descarcarea va porni automat
2. Dupa ce descarcarea s-a finalizat, deschideti si rulati aplicatia pentru pornirea instalarii
3. In fereastra care se deschide, apasati pe ***Signup for a Postman Account*** si incepeti instalarea
4. La final dati click pe ***Save My Preferences***. Veți vedea ecranul de Startup

Componente ale aplicației POSTMAN

•**New** - Opțiune folosită pentru a crea un nou request, colecție sau mediu de testare (sau alte elemente utile pentru dezvoltare)

•**Import** – Opțiune folosită pentru importarea colecțiilor din exterior

•**My Workspace** – Un concept similar cu cel de proiect, în care se vor stoca toate requesturile din cadrul organizației sau echipei

•**Invite** – Opțiune folosită pentru a invita alți oameni sa colaboreze la proiectul nostru.

•**History** – Contine toate request-urile trimise anterior in workspace-ul curent

•**Collections** – Contine o serie de requesturi care sunt grupate in functie de diverse obiective. O colectie poate contine subfoldere. Subfolderele si requesturile pot fi dublate (desi nu se recomanda)

•**Request tab** – Arata numele request-urilor pe care le ai deschise

•**HTTP Request** – Contine un dropdown cu mai multe metode de HTTP cum ar fi GET, POST, COPY, DELETE, etc. In testarea de API, cele mai folosite metode sunt GET si POST.

…Componente ale aplicației POSTMAN

•**Request URL** – Mai poarta numele de endpoint, si reprezintaun link pe care API-ul il va folosi pentru comunicare

•**Save** – Optiune pentru a salva noul request sau pentru a actualiza un request creat anterior in urma unor schimbari

•**Params** – Stocheaza parameterii necesari pentru filtrarea unui request sub forma unei perechi cheie-valoare

•**Authorization** - Loc in care sunt stocate datele de autentificare pentru a putea fi autorizati sa executam request-ul. Aici vom pune token-ul pentru Oauth daca este necesar.

•**Headers** – Headers este locul in care vom defini informatii legate de tipul request-ului cum ar fi content type JSON. In practica, tot aici se face si autorizarea

•**Body** – Informatia care va fi pasata API-ului intr-un request de POST, PUT sau PATCH .

•**Pre-request Script** – Bucati de cod care vor fi executate automat inainte de request. Cele mai des folosite scripturi sunt cele pentru setarea mediului

•**Tests** – Bucati de cod executate automat dupa executarea request-ului cu scopul de a verifica daca raspunsul returnat in timpul executarii testului este cel asteptat (mesaj, cod, timp de executie, informatii etc)

{iT}Factory

Intrebari de interviu:

- Ce inseamna API?
- Ce este Postman?
- Care este diferenta intre HTML si HTTP?
- De la ce vine ”s”-ul de la HTTPS?

Intrebari & Curiozitati?

{iT}Factory

Intalnirea 9

Testarea de API

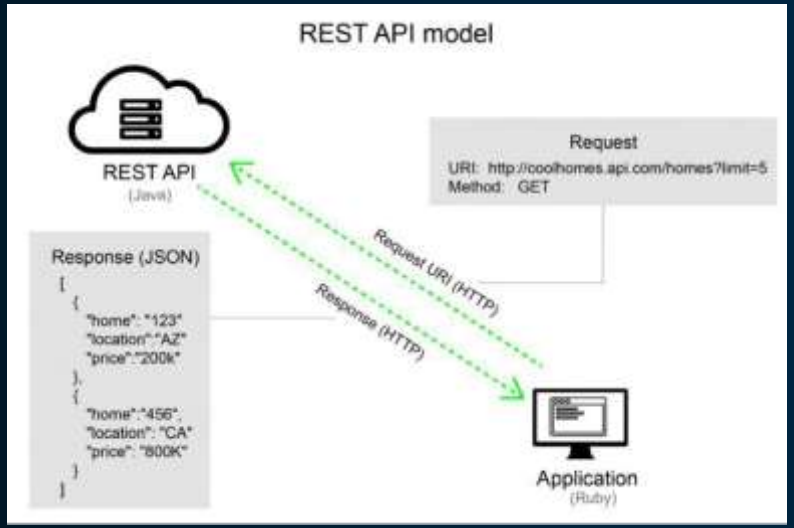
- Studiu Optional -

{iT}Factory

Rest API vs SOAP API

- SOAP este prescurtarea de la **S**imple **O**bject **A**ccess **P**rotocol si REST este prescurtarea de la **R**epresentational **S**tate **T**ransfer.
- Deoarece SOAP este un protocol, respecta o serie de standarde care permit o comunicare mai buna intre client și server, în timp ce REST este un stil arhitectural care nu respecta standarde stricte, dar care respecta sase constrangeri care il definesc: **Uniform Interface, Client-Server, Stateless, Cacheable, Layered System, Code on Demand.**
- SOAP folosește uses doar limbajul XML pentru schimbul de informații, în schimb ce REST poate suporta si alte formate cum ar fi JSON sau Plain-text.
- REST poate folosi protocolul SOAP dar SOAP nu poate folosi REST.
- SOAP este dificil de implementat si are nevoie de mai multă lungime de banda pentru procesare, în schimb ce REST este mai ușor de implementat și are nevoie de mai putine resurse de procesare (ex: pentru smartphone-uri)
- SOAP oferă beneficiul de a suporta tranzacții care respecta acronimul ACID. Unele aplicații necesita capacitatea de tranzacționare pe care o oferă SOAP și pentru care REST nu oferă suport.
- Din punctul de vedere al securității de transmitere a datelor, SOAP foloseste SSL (**S**ecure **S**ocket **L**ayer) si WS-security, în schimb ce REST folosește SSL și HTTPS (Hypertext Transfer Protocol Secure). În cazul unui cont bancar cu parola, numar de card si alte date sensibile, protocolul SOAP este preferat protocolului REST datorită avantajului de securitate oferite de WS-security.

Rest API vs SOAP



Structura unui request si a unui raspuns intr-un REST API

{iT}Factory

Diferente intre REST API si SOAP API

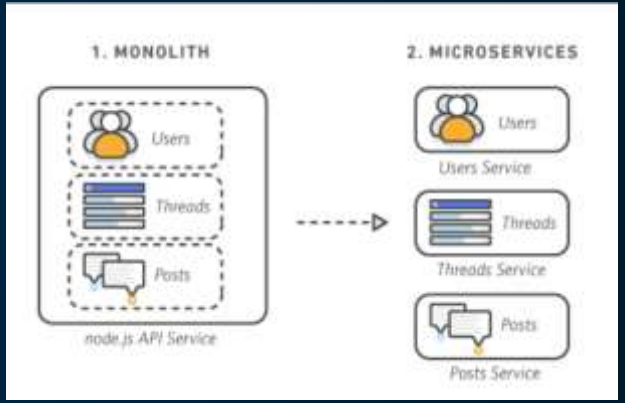
1	Implementare	Rest API nu respecta niciun standard, deoarece este un stil arhitectural	SOAP foloseste standarde oficiale, fiind un protocol de securitate
2	Comunicare Interna	REST API foloseste formate cum ar fi HTTP, JSON, URL, si XML pentru transfer de date	SOAP API foloseste doar HTTP si XML
3	Necesarul de resurse	Un REST API utilizează mai puține resurse în comparație cu SOAP API.	SOAP API necesita mai multe resurse de procesare deoarece necesita convertirea datelor în format XML care cresc nivelul de date care trebuie procesate
4	Descriere	REST API foloseste Web Application Description Language pentru a defini functionalitatile oferite de un Web Service	SOAP API foloseste tot Web Application Description Language pentru a defini functionalitatile oferite de un Web Service
5	Securitate	REST folosește SSL și HTTPS pentru securitate	SOAP foloseste SSL(Secure Socket Layer) si WS-security.
6	Abreviere	REST este prescurtarea de la Representational State Transfer.	SOAP este prescurtarea de la Simple Object Access Protocol
7	Interschimbabilitate	REST poate sa se folosească de SOAP SOAP ca si protocol pentru un web service.	OAP nu poate folosi REST, deoarece SOAP este un protocol si REST este un stil arhitectural.

{iT}Factory

Microservicii

Microserviciile reprezinta stiluri arhitecturale care structureaza o aplicatie într-o colecție de servicii mai ușor de gestionat.

Intr-un microsevice, aplicația software este formata din mai multe servicii independente care comunica între ele prin intermediul API-ului



{iT}Factory

XML vs JSON

JSON	XML
JSON are tipuri de date: string, number, array, Boolean	XML nu are tipuri de date (toate informațiile vor fi tratate ca și string)
Datele sunt accesibile ușor în obiectele JSON	Informațiile în format XML trebuie sa fie parsate.
JSON este suportat de majoritatea browserelor.	XML necesita parsare pentru compatibilitate intre browsere, lucru care uneori poate fi dificil
JSON nu oferă capacitate de afișare, motiv pentru care informațiile trebuie sa fie extrase prin intermediul Java Script	XML offers capacitate de afișare pentru ca este un markup language
Permite deserializare și serializare pe baza Java Script.	Dezvoltatorii trebuie sa scrie cod Java Script pentru serialize si deserialize din XML
Suporta doar UTF-8.	Suporta mai multe tipuri de criptari
Nu suporta comentarii	Suporta comentarii
Sunt ușor de citit	XML sunt mai dificil de citit si interpretat
Nu furnizează suport pentru namespaces	Furnizează suport pentru namespaces
Este mai putin sigur	Ofera mai multa securitate

- XML = E**x**tensible **M**arkup **L**anguage
- JSON = **J**ava**S**cript **O**bject **N**otation

{iT}Factory

Materiale Suplimentare

Tutoriale XML

https://www.w3schools.com/xml/
https://www.tutorialspoint.com/xml/index.htm
https://www.javatpoint.com/xml-tutorial
https://www.educba.com/software-development/software-development-tutorials/xml-tutorial/
https://beginnersbook.com/2018/10/xml-tutorial-learn-xml/
https://www.guru99.com/xml-tutorials.html

Tutoriale JSON

https://www.tutorialspoint.com/json/index.htm
https://www.javatpoint.com/json-tutorial
https://www.w3schools.in/json/
https://www.w3resource.com/JSON/introduction.php
https://www.guru99.com/json-tutorial-example.html
https://www.softwaretestinghelp.com/json-tutorial/

Tutorial Postman

Postman Tutorial