Cerinte NodeJS

A. Crearea unui endpoint care asculta pe port-ul 9999. Endpoint-ul va primi mesaje JSON de forma:

{

"resourceType" : "Practitioner",

"id": "1",

"name": [{"family":"TestFamily","given":["TestGiven"],"text":"TestFamily TestGiven"}],

"facility": [

{

"value": "1",

"system": "http://us.gov/NPI",

"name": "Facility Name"

},

{

"value": "2",

"system": "http://us.gov/NPI",

"name": "Other Facility Name"

}

],

"active": true

}

JSON-ul de mai sus reprezinta un medic care lucreaza la mai multe clinici.

Cerinte:

1. La primirea unui JSON sa se afiseze in consola informatiile "name" si "facility" DOAR daca statusul este "active".

2. Daca proprietatea "id" nu este prezenta sau daca tipul resursei nu este "Practitioner" sa se dea mesaj de eroare.

3. Daca valoarea proprietatii "id" s-a mai introdus in cadrul aceleiasi rulari sa se dea un mesaj de eroare.

--------------------------------------------------------------------------------

B. Endpointul creat anterior trebuie sa accepte fisiere csv semnalate de header-ul HTTP "Content-type": "text/csv".

Aceste fisiere vor avea forma:

Header row: (va fi prezent si in fisierul csv, exact in aceasta forma)

ID, FamilyName, GivenName, FacilityId, SystemId, NameId, Active

Randuri exemplu din fisierul CSV:

1, Popescu, George, 12, http://ro.gov/NPI, Spital Tulcea, true

1, Popescu, George, 13, http://ro.gov/NPI, Spital Sfantu Gheorghe, true

1, Popescu, George, 13, http://ro.gov/NPI, Spital Constanta, false

2, Ionescu, Catalin, 12, http://ro.gov/NPI, Spital Tulcea, true

Dupa cum se poate observa daca un medic lucreaza la mai multe spitale el va fi multiplicat in randuri diferite pentru fiecare spital in parte.

Cerinte:

1. Dupa primirea csv-ului se va alcatui o lista cu toate spitalele la care lucreaza un anumit medic si va fi afisata in consola. Se vor afisa DOAR spitalele la care medicul are statusul "active": true (ultima prorietate din csv).

Exemplu:

Popescu George: Spital Tulcea, Spital Sfantu Gheorghe

Ionescu Catalin: Spital Tulcea

2. Daca csv-ul contine nume diferite pentru acelasi ID se va afisa o eroare.

--------------------------------------------------------------------------------

C. Se va securiza endpoint-ul facut anterior prin adaugarea unui header de autentificare numit x-vamf-jwt.

Acesta va avea valori Base64 encoded.

Exemplu:

ewogICAgImF1dGhlbnRpY2F0ZWQiOiB0cnVlLAogICAgImlzcyI6ICJKV1QgQnVpbGRlciIsCiAgICAiZmFjaWxpdHkiOiBbIjEyIiwgIjEzIl0KICAgICJyb2xlcyI6IFsKICAgIAkiQWRtaW4iCiAgCV0KfQ==

reprezinta

{

"authenticated": true,

"iss": "JWT Builder",

"facility": ["12", "13"]

"roles": [

"Admin"

]

}

Cerinte:

1. Se va permite accesul doar rolurilor "Admin" sau "Practitioner". In cazul altui rol se va intoarce o eroare.

2. Daca in body (varianta JSON de la pct 1) sau fisier CSV (varianta fisier de la pct 2) exista medici care figureaza la facility-uri care nu sunt in lista de "facility" din header-ul de autentificare aceste intrari vor fi ignorate.

Informatii ajutatoare:

Pentru testare se va folosi programul "Postman" (sau similar).