



# TEHNOLOGII WEB AVANSATE

Studiu de caz  
Angular14 și  
.NET Framework

Claudiu DOMUTA  
Dan GOTA  
Alexandra FANCA  
Adela POP  
Honoriu VALEAN

Claudiu Domuța • Dan Goța • Alexandra Fanca  
Adela Pop • Honoriu Vălean

# Tehnologii Web Avansate -Studiu de caz Angular14 și .NET Framework

Editura RISOPRINT

Cluj-Napoca 2023

## Contents

Introducere / 3

Ce este REST? / 3

Ce se va realiza? / 3

Unelte necesare / 3

    Instalarea Uneltelor necesare pentru implementarea Back-end / 4

    Instalarea Uneltelor necesare pentru implementarea Front-end / 4

    Instalarea Bazei de Date / 4

Implementare Server / 4

    .Net6 Web Api Framework / 4

    Entity Framework Core / 4

    Crearea Proiectului in Visual Studio / 5

    Implementarea Codului / 5

        Crearea de modele (entitati) / 5

        Conexiunea la Baza de Date / 5

        Realizarea Clasei Controller / 5

        Testare Backend / 5

Implementare Front-End / 6

    Angular / 6

        Introducere in Angular / 6

        Creare Proiect / 6

        Rularea Aplicatiei / 6

        Implementare / 6

Bibliografie / 8

## Introducere

Lucrarea de față prezintă modul cum se poate implementa o aplicație web full stack, folosind tehnologii avansate de programare.

Așadar, în prima parte a lucrării propuse, se va prezenta modul cum se poate expune un serviciu/server, respectând principiul **RESTful Web API**, implementat în **ASP.NET Core 6.0** și **Entity Framework Core**.

În partea a doua, se va arăta cum serviciul API creat poate fi consumat de către un client frontend (website) implementat cu ajutorul framework-ului **Angular**.

### Ce este REST?

Serviciile **REST** (Representational State Transfer), cunoscute și sub denumirea de servicii sau API-uri (Application Programming Interfaces) RESTful, sunt un tip de arhitectură pentru serviciile web. Acestea urmează un set de principii și directive pentru construirea de API-uri scalabile și interoperabile. REST este frecvent utilizat pentru comunicarea între o aplicație client, precum o aplicație mobilă, și un server.

### Ce se va realiza?

Aplicația va consta într-un sistem de management al angajaților unei companii. Utilizatorul va putea vizualiza, adăuga, modifica și șterge angajații prezenți în sistem folosind o aplicație dinamică web. Astfel, aplicația prezintă și implementează cele 4 acțiuni fundamentale ce pot fi aplicate asupra oricărei entități software:

- Creare;
- Citire;
- Editare;
- Ștergere.

Operațiile mai sus enumerate asupra angajaților se va face folosind **ORM-ul Entity Framework Core**.

## Unelte necesare

Înainte de implementare sunt necesare instalarea unor tool-uri (unelte) și pachete software care vor ajuta la scrierea codului și testarea funcționalităților expuse:

- Visual Studio
- Microsoft .Net 6 SDK
- NuGet package manager
- Node.js
- Angular
- VueJs

## Instalarea Uneltelor necesare pentru implementarea Back-end

În această secțiune se prezintă pașii de instalare a tool-urilor necesare pentru a implementa codul de server.

## Instalarea Uneltelor necesare pentru implementarea Front-end

În această secțiune se prezintă pașii de instalare a tool-urilor necesare pentru a implementa codul de front-end.

## Instalarea Bazei de Date

Bazele de date relaționale, utilizate în cadrul acestei lucrări, vor fi create cu ajutorul sistemului de gestiune a bazelor de date oferit la Microsoft. Pentru ca crearea bazelor de date relaționale gestionate de **SQL Server Express** să se facă cu ușurință, Microsoft pune la dispoziția programatorilor una gratuită **Microsoft SQL Server Management Studio Express (SSMSE)** care oferă o interfață grafică prietenoasă, intuitivă pentru gestiunea bazelor de date relaționale.

## Implementare Server

În acest capitol se urmărește crearea unui proiect de **API** folosind **.Net6 Web Api Framework**. Acest proiect se va conecta cu baza de date și va permite 4 operații importante asupra unei liste de angajați:

- Adaugare angajat în baza de date
- Editarea unui angajat
- Găsirea listei de angajați din baza de date
- Ștergerea unui angajat

## .Net6 Web Api Framework

**ASP.NET Web API** este un framework folosit pentru crearea de servicii HTTP care pot fi accesate de către orice client extern cum ar fi browsere (Internet Explorer, Chrome, etc), aplicații mobile (Android, IOS, etc). Aceasta reprezintă o platformă ideală pentru implementarea aplicațiilor RESTful în .NET Framework.

## Entity Framework Core

Entity Framework (Core) este un ORM (Object Relational Mapper) ce realizează maparea dintre modelele realizate în C# și tabele din baza de date. Un model nu este nimic altceva decât o clasă ce conține drept atribute una sau mai multe proprietăți. De asemenea, există două modalități prin care Entity Framework operează:

- **Code First:** Tabele din baza de date vor fi generate pe baza modelelor din C# printr-un procedeu numit **migrare (migration)**
- **Database First:** Modele din C# sunt generate pe baza unor scripturi SQL

În general, se preferă opțiunea Code First deoarece este mai ușor de implementat și de înțeles fiind mai des utilizată în practică. Acestea sunt motivele pentru care și în această lucrare se va folosi această opțiune.

## Crearea Proiectului in Visual Studio

În această secțiune se prezintă pașii pentru crearea proiectului de API folosind Visual Studio.

## Implementarea Codului

În acest subcapitol se va implementa codul necesar în realizarea aplicației API.

### Crearea de modele (entități)

Modelele/Entitățile reprezintă clase folosite pentru a stoca datele în baza de date, în interiorul căruia se efectuează operațiile de CRUD, dar de asemenea reprezintă informația care va fi trimisă între partea de frontend și backend.

### Conexiunea la Baza de Date

Pentru a crea stringul de conexiune către baza de date se va deschide Microsoft SQL Server Management Studio instalat anterior și se va obține numele serverului.

### Realizarea Clasei Controller

În această parte a lucrării se prezintă realizarea unei clase de tip controller. Aceste tipuri de clase au rolul de a răspunde operațiilor HTTP ( GET, POST, etc.), dar de asemenea vor avea rolul de a realiza operațiile CRUD către baza de date fie în mod direct fie prin intermediul altor servicii adiționale.

În general toate clasele care vor avea rolul de controller vor fi create într-un folder **Controllers**. Prin urmare înainte de crearea claselor de tip controller, se va crea un folder cu numele **Controllers** în interiorul proiectului deja creat în Visual Studio.

#### *Obține angajații din DB*

#### *Adaugarea de angajat nou*

#### *Obține angajat dupa id*

#### *Update Angajat*

#### *Stergere Angajat*

#### Testare Backend

Pentru testarea implementării părții backend se poate folosi unul din următoarele unelte: Postman sau Swagger.

În lucrarea de față se va folosi Swagger deoarece este foarte bine integrat cu aplicațiile .NET. Swagger este o unealtă care ajută la dezvoltarea, descrierea și testarea aplicațiilor RESTful APIs. Prin folosirea Swagger pentru testare se pot observa următoarele specificații ale aplicației:

- Care sunt operațiile suportate de către API-ul expus;
- Care sunt parametrii API-ului și ce returnează;
- Dacă pentru a realiza o anumită operație este nevoie de autorizare sau nu;
- Etc.

**Testare metodei obținere listei de angajați**

**Testarea metodei de adaugare**

**Testarea metodelor de obținere angajat după un id, editare și ștergere angajat**

## Implementare Front-End

În această parte a lucrării propuse se prezintă codul necesar în vederea creării aplicației UI folosind tehnologia Angular.

### Angular

Introducere în Angular

Creare Proiect

În vederea creării unui proiect de tip front-end folosind tehnologia Angular se vor urma pașii descriși în această secțiune.

Rularea Aplicației

## Fixarea Problemei legat de ExecutionPolicy

### Implementare

În acest subcapitol se prezintă pașii de implementare a codului aplicației UI. Pentru început se va curăța codul generat implicit deoarece acesta conține doar cod html static. Pentru a realiza acest lucru se identifică fișierul `app.component.html` din proiect și se șterge conținutul fișierului mai puțin tag-ul **router-outlet**, care este necesar în realizarea rutării acțiunilor.

## Implementare Bara de Navigare

### Implementare afișare angajați

În continuare se va adăuga un tabel html care va afișa detalii despre angajații în pagina html Angajati tocmai creată.

### Implementare adăugare angajat

În această secțiune se prezintă codul necesar pentru adăugarea unui angajat. Pentru aceasta se efectuează pașii descriși în această secțiune.

### Vizualizare, Editare și ștergere date angajat

În această parte a lucrării se va implementa butoane de vizualizare, modificare și ștergere a unei înregistrări.

#### Vizualizare și editare angajat

Pentru funcționalitatea de editare se va crea o nouă componentă în fișierul de **components** al aplicației.

#### Stergere Angajat

În această secțiune se prezintă codul UI necesar operației de ștergere a unui angajat.

### Angajati

| Id                                   | Name     | Email        | Phone     | Salary | Department    |                          |
|--------------------------------------|----------|--------------|-----------|--------|---------------|--------------------------|
| fe69e2b4-580a-4f82-8a74-823c6e92a3ce | dadada   | s@a.ro       | 365       | 6354   | IT            | <a href="#">Editeaza</a> |
| d16d9422-d653-4641-88cb-8e212a48b911 | John Doe | JD@yahoo.com | 999888777 | 60000  | Resurse Umane | <a href="#">Editeaza</a> |

## Bibliografie

- [1] <https://www.contrive.mobi/aviorapi/HTTPMETHODS.html#:~:text=The%20primary%20or%20most%2Dcommonly,but%20are%20utilized%20less%20frequently.>
- [2] <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API#:~:text=A%20RESTful%20API%20is%20an,deleting%20of%20operations%20concerning%20resources.>
- [3] <https://www.linkedin.com/pulse/understanding-rest-architecture-gabriel-gitonga>
- [4] ", „What Are RESTful Web Services?“, " 2013.
- [ 5] ", „What is a REST API?“, [Online]. Available: <https://www.ibm.com/topics/restapis>.
- [ 6] A. Ehsan, M. A. M. E. Abuhaliqa, C. Catal și D. Mishra, „RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions,“ pp. 1-4, 2022.
- [7] <https://visualstudio.microsoft.com/downloads/>
- [8] <https://dotnet.microsoft.com/en-us/download>
- [9] <https://code.visualstudio.com/download>
- [10] <https://nodejs.org/en/download/current>
- [11] <https://www.tutorialsteacher.com/webapi#:~:text=ASP.NET%20Web%20API%20is,applications%20on%20the%20.NET%20Framework.>
- [12] <https://dotnet.microsoft.com/en-us/apps/aspnet/apis>
- [13] <https://www.postman.com/>
- [14] <https://swagger.io/docs/specification/2-0/what-is-swagger/>
- [15] <https://getbootstrap.com/>
- [16] <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>
- [17] <https://www.microsoft.com/en-NZ/download/details.aspx?id=8961>
- [18] <https://learn.microsoft.com/en-us/ef/core/>
- [19] Entity Framework Core - The Complete Guide (.NET Core 7), Bhrugen Patel