WebAPI

In this video we will discuss creating ASP.NET Web API service that returns data from a SQL Server database. We will be using this service as the basis for understanding many of the Web API concepts in our upcoming videos.

The Web API **EmployeeService** that we will be building will retrieve and return the data from the following SQL Server Employees table. We will be using Entity Framework to retrieve data from the database. You can use any technology of your choice to retrieve data from the database. For example you can even use raw ADO.NET.

  
  
Execute the following SQL Script using SQL Server Management studio. This script creates  
1. EmployeeDB database  
2. Creates the Employees table and populate it with sample data

Create Database EmployeeDB

Go

Use EmployeeDB

Go

Create table Employees

(

ID int primary key identity,

FirstName nvarchar(50),

LastName nvarchar(50),

Gender nvarchar(50),

Salary int

)

Go

Insert into Employees values ('Mark', 'Hastings', 'Male', 60000)

Insert into Employees values ('Steve', 'Pound', 'Male', 45000)

Insert into Employees values ('Ben', 'Hoskins', 'Male', 70000)

Insert into Employees values ('Philip', 'Hastings', 'Male', 45000)

Insert into Employees values ('Mary', 'Lambeth', 'Female', 30000)

Insert into Employees values ('Valarie', 'Vikings', 'Female', 35000)

Insert into Employees values ('John', 'Stanmore', 'Male', 80000)

Go  
**Creating a new ASP.NET Web API Project**  
1. Open Visual Studio and select File - New - Project  
  
2. In the "New Project" window  
Select "Visual C#" under "Installed - Templates"  
From the middle pane select, ASP.NET Web Application  
Name the project "EmployeeService" and click "OK"



3. On the next window, select "Web API" and click "OK"  
  
At this point you should have the Web API project created.  
  
**Adding ADO.NET Entity Data Model to retrieve data**  
1. We will have the Entity Model in a separate project.  
  
2. Right click on EmployeeService solution in the Solution Explorer and select Add - New Project  
  
3. In the Add New Project window  
Select Visual C# from the left pane  
Class Library Project from the Middle pane  
Name the project EmployeeDataAccess and click OK



4. Right click on EmployeeDataAccess project and select Add - New Item  
  
5. In the "Add New Item" window  
Select "Data" from the left pane  
Select ADO.NET Entity Data Model from the middle pane  
In the Name text box, type EmployeeDataModel and click Add



6. On the Entity Data Model Wizard, select "EF Designer from database" option and click next



7. On the next screen, click "New Connection" button   
  
8. On "Connection Properties" window, set  
Server Name = (local)  
Authentication = Windows Authentication  
Select or enter a database name = EmployeeDB  
Click OK and then click Next  
  
9. On the nex screen, select Entity Framework 6.x



10. On the nex screen, select "Employees" table and click Finish.



**Using the Entity Data Model in EmployeeService project**  
1. Right click on the references folder in the EmployeeService project and select "Add Reference"  
  
2. On the "Reference Manager" screen select "EmployeeDataAccess" project and click OK.



**Adding Web API Controller**  
1. Right click on the Controllers folder in EmployeeService project and select Add - Controller  
  
2. Select "Web API 2 Controller - Empty" and click "Add"  
  
3. On the next screen set the Controller Name = EmployeesController and click Add  
  
4. Copy and paste the following code in EmployeesController.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

using EmployeeDataAccess;

namespace EmployeeService.Controllers

{

public class EmployeesController : ApiController

{

public IEnumerable<Employee> Get()

{

using(EmployeeDBEntities entities = new EmployeeDBEntities())

{

return entities.Employees.ToList();

}

}

public Employee Get(int id)

{

using (EmployeeDBEntities entities = new EmployeeDBEntities())

{

return entities.Employees.FirstOrDefault(e => e.ID == id);

}

}

}

}  
5. At this point build the solution and navigate to /api/employees. You will get the following error.  
No connection string named 'EmployeeDBEntities' could be found in the application config file.  
  
6. This is because "Entity Framework" is looking for EmployeeDBEntities connection string in the web.config file of EmployeeService project. EmployeeDBEntities connection string is actually in App.config file of EmployeeDataAccess class library project. Include a copy of this connection string in web.config file.  
  
At this point when you navigate to /api/employees you should see all employees and when you navigate to /api/employees/1 you should see all the details of the employee whose Id=1  
  
In our next video we will discuss the concept of **Content Negotiation** in Web API