# BÀI 16: THỰC HÀNH 07 – XÂY DỰNG ỨNG DỤNG ASP.NET CORE WEB API

***Mục tiêu***

* Trình bày được một số khái niệm ASP.NET CORE WebAPI
* Phân biệt, so sánh ASP.NET WebAPI và ASP.NET CORE WebAPI
* Trình bày được các bước xây dựng ASP.NET CORE WebAPI
* Vận dụng và tổng hợp được để xây dựng được ứng dụng ASP.NET CORE WebAPI theo yêu cầu
* Nghiêm túc, chăm chỉ, nhiệt tình, tích cực trong học tập

Hướng dẫn thực hiện



**Các bước xây dựng ASP.NET WEB API Core**

Bước 1: Tạo cơ sở dữ liệu và tạo dự án ASP.NET Web API Core

Bước 2: Xây dựng các Entity

Bước 3: Viết Web API Controller

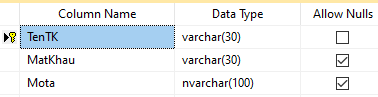
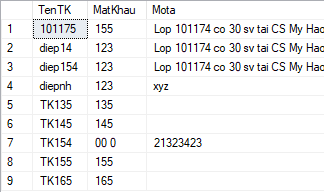
Bước 4: Thiết lập

Bước 5: Chạy và debug ứng dụng

#### Cài thư viện

Microsoft.AspNetCore.Identity.EntityFrameworkCore

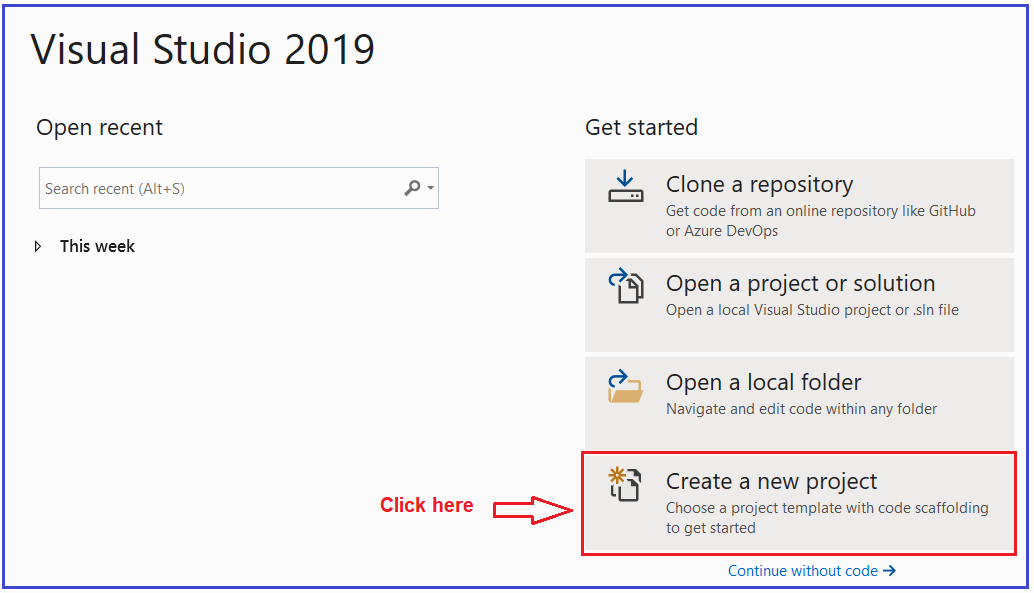
**Yêu cầu:** Xây dựng ASP.NET WEB API, trong đó cơ sở dữ liệu chứa bảng TaiKhoan gồm TenTK, MatKhau và Mota kiểu xâu ký tự và dữ liệu như sau:

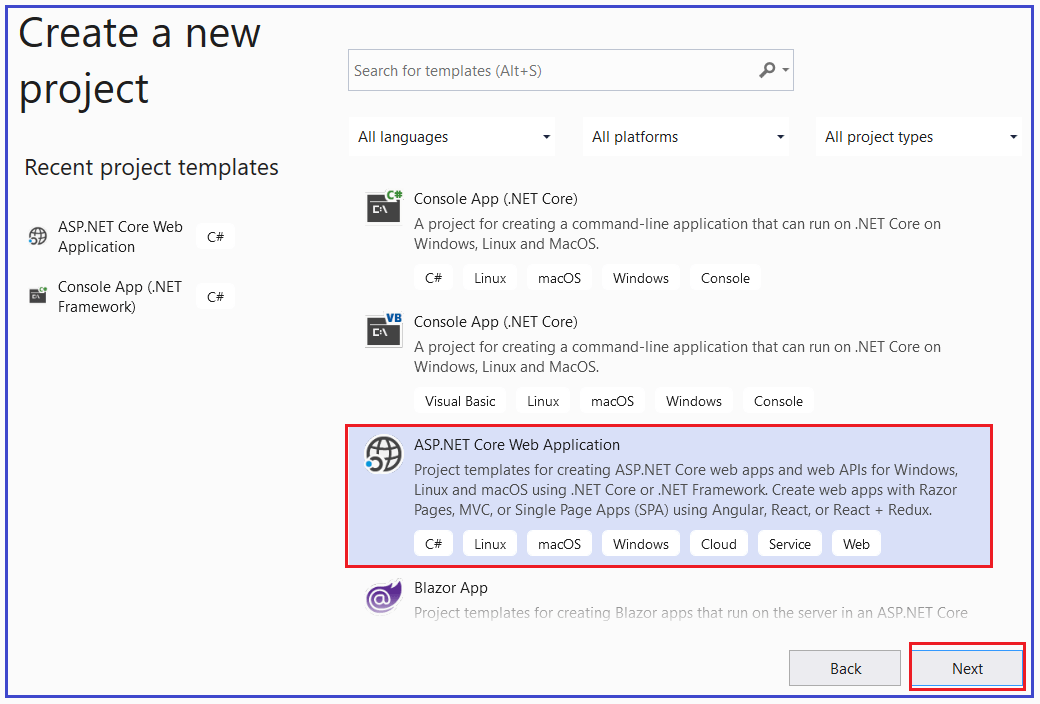
 

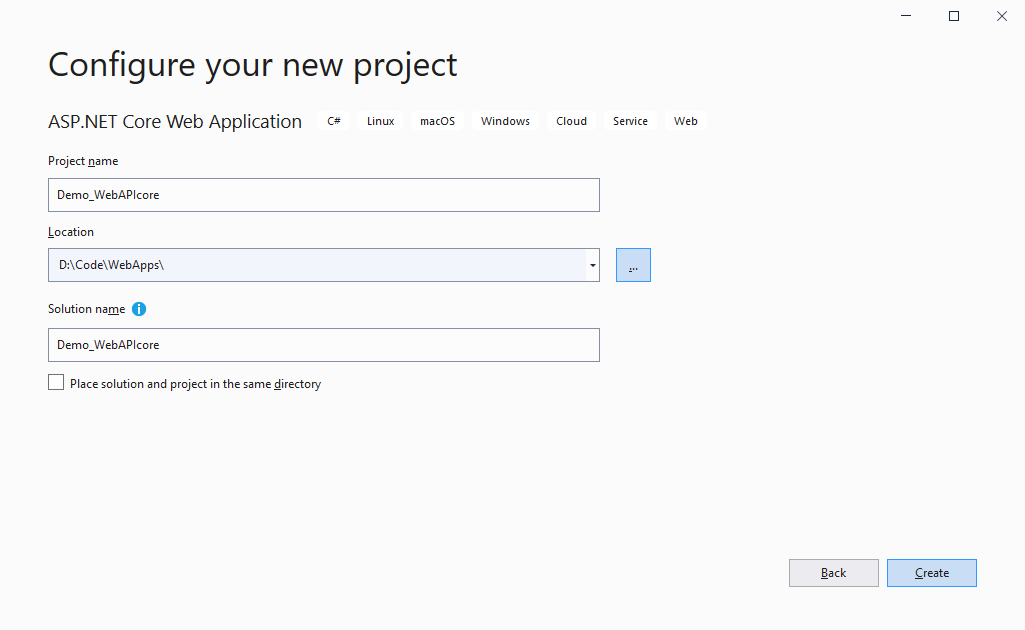
***Hướng dẫn***

#### Bước 1: Tạo CSDL và tạo dự án asp net core 3.0 web api using visual studio 2019

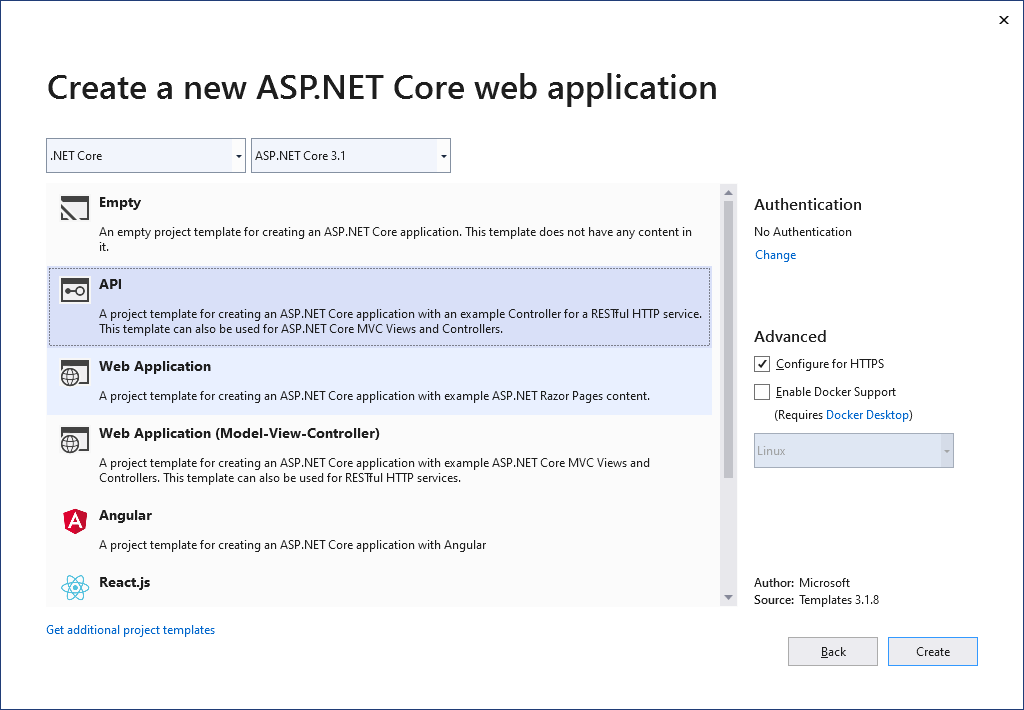
Khởi động Visual Studio => Click chọn **Create a new project** => chọn  **ASP.NET Core Web Application**để tạo một project ASP.NET Web Application => Click chọn **Next** button.







Đặt tên cho ứng dụng, chọn nơi lưu trữ và FrameWork => rồi chọn template API=> rồi Click chọn  **Create**button để tạo ứng dụng WebAPI NET core **Asp Net Core 3.0 web api**



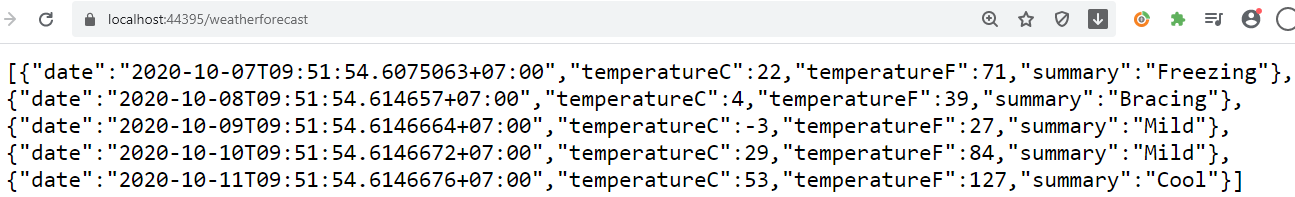
Cài package Microsoft.AspNetCore.Identity.EntityFrameworkCore

Sau khi ứng dụng được tạo ra, nó có sẵn controller template là WeatherForecastController

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Threading.Tasks;  using Microsoft.AspNetCore.Mvc;  using Microsoft.Extensions.Logging;  namespace Demo\_WebAPIcore.Controllers  {  [ApiController]  [Route("[controller]")]  public class WeatherForecastController : ControllerBase  {  private static readonly string[] Summaries = new[]  {  "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot", "Sweltering", "Scorching"  };  private readonly ILogger<WeatherForecastController> \_logger;  public WeatherForecastController(ILogger<WeatherForecastController> logger)  {  \_logger = logger;  }  [HttpGet]  public IEnumerable<WeatherForecast> Get()  {  var rng = new Random();  return Enumerable.Range(1, 5).Select(index => new WeatherForecast  {  Date = DateTime.Now.AddDays(index),  TemperatureC = rng.Next(-20, 55),  Summary = Summaries[rng.Next(Summaries.Length)]  })  .ToArray();  }  }  } |

#### Chạy ứng dụng NET Core template với url [https://localhost:<port>/WeatherForecast](https://localhost:%3cport%3e/WeatherForecast)

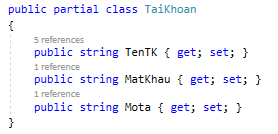
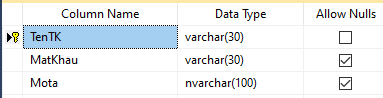
Kết quả như sau



#### Bước 2: Tạo models\Entity class

* Tạo thư mục Models.
* Add class trong Models folder

Chú ý entity class khớp với các trường trong bảng dữ liệu tương ứng



|  |
| --- |
| Models\TaiKhoan |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Threading.Tasks;  namespace Demo\_WebAPIcore.Models  {  public class TaiKhoan  {  public string TenTK { get; set; }  public string MatKhau { get; set; }  public string Mota { get; set; }  }  } |

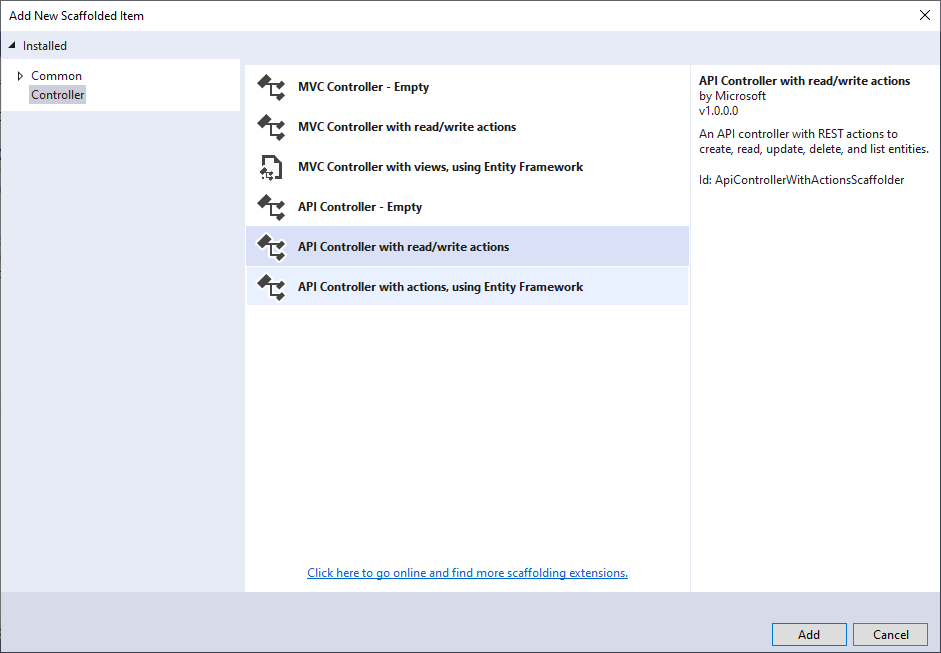
Có thể xây dựng lớp DataContext kế thừa từ lớp DbContext trong package Microsoft.EntityFrameworkCore;

Viết phương thức khởi dụng và thuộc tính tương ứng lưu dữ liệu lấy từ CSDL về như sau

|  |
| --- |
| Models\TaiKhoanContext |
| using Microsoft.EntityFrameworkCore;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Threading.Tasks;  namespace Demo\_WebAPIcore.Models  {  public class TaiKhoanContext : DbContext  {  public TaiKhoanContext(DbContextOptions<TodoContext> options) : base(options)  {  }  public DbSet<TaiKhoan> taiKhoans { get; set; }  }  } |

#### Tạo Controller TaiKhoansController.cs

Tạo thư mục Controllers, rồi add Controller. Chọn API Controller – Empty hoặc API Controller with actions như sau:



Viết Controller

|  |
| --- |
| Controllers\TaiKhoansController.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Threading.Tasks;  using Microsoft.AspNetCore.Mvc;  using Microsoft.Extensions.Logging;  using System.Data;  using System.Data.SqlClient;  using Microsoft.Data.SqlClient;  using Demo\_WebAPIcore.Models;  namespace Demo\_WebAPIcore.Controllers  {  //Route("api/[controller]")]  [ApiController]  public class TaiKhoansController : ControllerBase  {  String strconnect = @"data source=DESKTOP-KL7N773\SQLEXPRESS2014; initial catalog=DEMOCRUD; integrated security=true;";  private ILogger<TaiKhoansController> \_logger;  public TaiKhoansController(ILogger<TaiKhoansController> logger)  {  \_logger = logger;  }  [HttpGet]  [Route("api/{Controller}")]  public List<Models.TaiKhoan> getall()  {  SqlConnection cn = new SqlConnection(strconnect);  String sqlString = @"select \* from taikhoan";  SqlDataAdapter da = new SqlDataAdapter(sqlString, cn);  DataTable dt = new DataTable();//dt chứa dữ liệu lấy  da.Fill(dt); //Đẩy dữ liệu da vào dt  List<Models.TaiKhoan> ds = new List<Models.TaiKhoan>();  foreach (DataRow x in dt.Rows)  {  //TaiKhoan db = new TaiKhoan();  //db.TenTK= x[0].ToString();  //db.MatKhau = x[1].ToString();  //db.Mota = x[2].ToString();  //ds.Add(db);  ds.Add(new TaiKhoan() { TenTK = x[0].ToString(), MatKhau = x[1].ToString(), Mota = x[2].ToString() });  }  return ds;  }  [HttpGet]  [Route("api/{Controller}/{TenTK}")]  public TaiKhoan Get(string TenTK)  {  // string strconnect = @"data source=DESKTOP-KL7N773\SQLEXPRESS2014; initial catalog=DEMOCRUD; integrated security=true;";  SqlConnection cn = new SqlConnection(strconnect);  // SqlDataAdapter da = new SqlDataAdapter(string.Format("select \* from taikhoan where TenTK='{0}'", TenTK), cn);  String sqlString = "select \* from taikhoan where TenTK=\'" + TenTK + "\'";  SqlDataAdapter da = new SqlDataAdapter(sqlString, cn);  DataTable dt = new DataTable();  da.Fill(dt);  TaiKhoan db = new TaiKhoan();  if (dt.Rows.Count > 0)  try  {  DataRow x = dt.Rows[0];  db.TenTK = x[0].ToString();  db.MatKhau = x[1].ToString();  db.Mota = x[2].ToString();  }  catch (Exception e) { }  return db;  }  [HttpPut]  [Route("api/{Controller}/{TenTK}")]  public TaiKhoan update(string TenTK, TaiKhoan abc)  {  SqlConnection cn = new SqlConnection(strconnect);  //String sqlString = @"select \* from taikhoan";  SqlDataAdapter da = new SqlDataAdapter(string.Format(@"select \* from taikhoan where TenTK='{0}'", TenTK), cn);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  cn.Open();  SqlCommand cmd = new SqlCommand(string.Format(@"update taikhoan set Matkhau='{0}',mota='{1}' where TenTK='{2}'", abc.MatKhau, abc.Mota, TenTK), cn);  cmd.ExecuteNonQuery();//cập nhật dữ liệu  cmd.Dispose();  cn.Close();  //abc.TenTK = TenTK;  return abc;  }  else  return null;  }  [HttpDelete]  [Route("api/{Controller}/{TenTK}")]  public Boolean Delete(string TenTK)//xl  {  SqlConnection cn = new SqlConnection(strconnect);  //String sqlString = @"select \* from taikhoan";  SqlDataAdapter da = new SqlDataAdapter(string.Format(@"select \* from taikhoan where TenTK='{0}'", TenTK), cn);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)//có  {  cn.Open();  //SqlCommand cmd = new SqlCommand(string.Format(@"delete from taikhoan where TenTK='{0}'", TenTK), cn);  String deleteString= "delete from taikhoan where TenTK = \'"+ TenTK+"\'";  //cmd.ExecuteNonQuery();//lỗi do lệnh ExecuteNonQuery để cập nhật dữ liệu  // 1. Instantiate a new command  SqlCommand cmd = new SqlCommand();  // 2. Set the CommandText property  cmd.CommandText = deleteString;  // 3. Set the Connection property  cmd.Connection = cn;  // 4. Call ExecuteNonQuery to send command  cmd.ExecuteNonQuery();    cmd.Dispose();  cn.Close();  return true;  }  else return false;  }  [HttpPost]  [Route("api/{Controller}")]  public TaiKhoan Post(TaiKhoan abc)  {  SqlConnection cn = new SqlConnection(strconnect);  //String sqlgetString = @"select \* from taikhoan";  //String postsqlString = "INSERT INTO taikhoan VALUES('{0}', '{1}', '{2})'", abc.TenTK, abc.MatKhau, abc.Mota);  SqlDataAdapter da = new SqlDataAdapter(string.Format("select \* from taikhoan where TenTK='{0}'", abc.TenTK), cn);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count == 0)//chưa có  {  cn.Open();  SqlCommand cmd = new SqlCommand(string.Format("INSERT INTO taikhoan VALUES ('{0}','{1}','{2})'", abc.TenTK, abc.MatKhau, abc.Mota), cn);  cmd.ExecuteNonQuery();  cmd.Dispose();  cn.Close();  return abc;  }  else return null;//đã có ko thêm  }  }  } |

#### Bước 4: Thiết lập

#### setup Database and Implement Microsoft Identity in Asp Net Core 3.0 web API Project

**mở file appsettings.json** thêm connection string như sau:

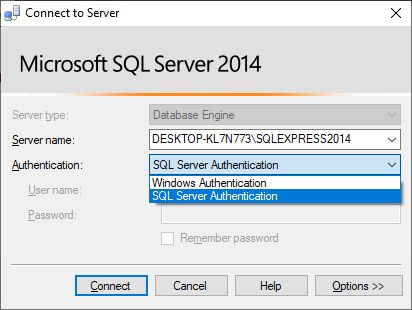
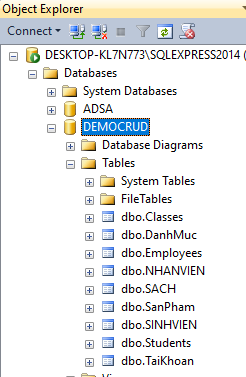
|  |
| --- |
| "ConnectionStrings": {  "DefaultConnection": "Data Source=DataSourceName; Initial Catalog=TokenBaseAuthDB;Integrated Security=True"  } |
| Khi kết nối CSDL copy lấy thông tin của connectionStrings của CSDL  data source=DESKTOP-KL7N773\SQLEXPRESS2014  initial catalog= DEMOCRUD  Integrated Security=True |
| {  "ConnectionStrings": {  "DefaultConnection": "Data Source=DESKTOP-KL7N773\\SQLEXPRESS2014; Initial Catalog=DEMOCRUD;Integrated Security=True"  },  "Logging": {  "LogLevel": {  "Default": "Information",  "Microsoft": "Warning",  "Microsoft.Hosting.Lifetime": "Information"  }  },  "AllowedHosts": "\*"  } |

Điểm quan trọng nhất mà bạn cần lưu ý là tệp launcherSettings.json này chỉ được sử dụng trong máy phát triển cục bộ

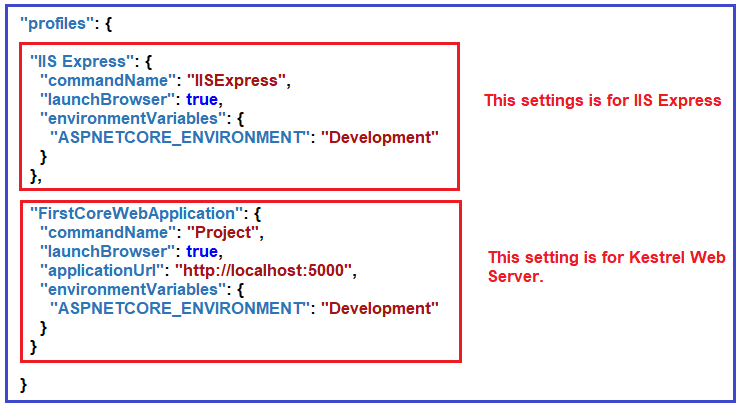
Chú ý! DataSourceName là tên server CSDL (mở hệ quản trị CSDL ra, connect để copy database name)

Initial Catalog là tên của CSDL

Nếu CSDL có sd tài khoản mật khẩu thì thêm trường user = True



#### launchSettings.json file

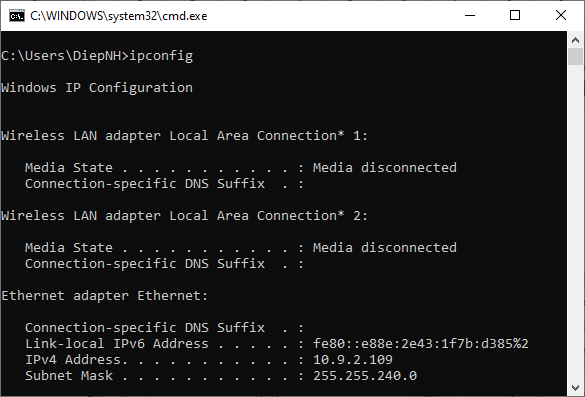


|  |
| --- |
| ASP.NET Core launchSettings.json file |
| {  "$schema": "http://json.schemastore.org/launchsettings.json",  "iisSettings": {  "windowsAuthentication": false,  "anonymousAuthentication": true,  "iisExpress": {  "applicationUrl": "http://localhost:49480",  "sslPort": 44395  }  },  "profiles": {  "IIS Express": {  "commandName": "IISExpress",  "launchBrowser": true,  "launchUrl": "weatherforecast",  "environmentVariables": {  "ASPNETCORE\_ENVIRONMENT": "Development"  }  },  "Demo\_WebAPIcore": {  "commandName": "Project",  "launchBrowser": true,  "launchUrl": "weatherforecast",  "applicationUrl": "https://localhost:5001;http://localhost:5000",  "environmentVariables": {  "ASPNETCORE\_ENVIRONMENT": "Development"  }  }  }  } |

Như cấu hình ở trên nên chạy url mặc định "launchUrl": "weatherforecast

Có thể cấu hình địa chỉ IP của máy thay cho localhost

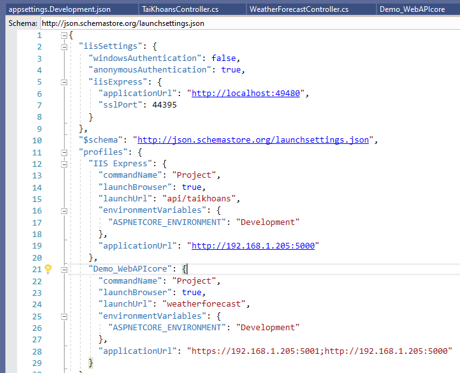
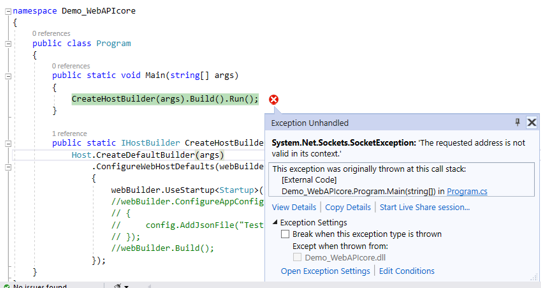
Giả sử thông tin dịa chỉ ip của máy tính là 10.0.2.109 như sau



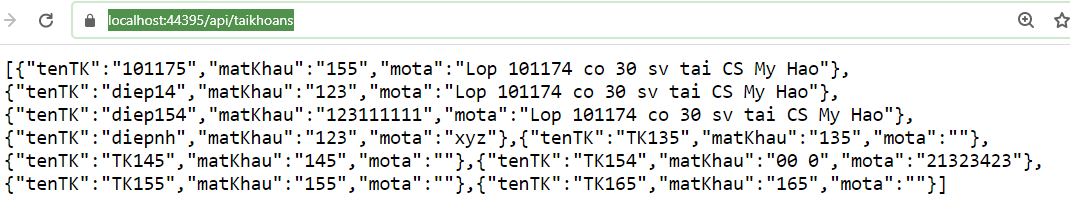
Thì thiết lập url như sau:

|  |
| --- |
|  |
| {  "iisSettings": {  "windowsAuthentication": false,  "anonymousAuthentication": true,  "iisExpress": {  "applicationUrl": "http://10.9.2.109:5000",//"http://localhost:49480",  "sslPort": 5000  }  },  "$schema": "http://json.schemastore.org/launchsettings.json",  "profiles": {  "IIS Express": {  "commandName": "Project",  "launchBrowser": true,  "launchUrl": "api/taikhoans",  "environmentVariables": {  "ASPNETCORE\_ENVIRONMENT": "Development"  },  "applicationUrl": "http://10.9.2.109:5000"  },  "Demo\_WebAPIcore": {  "commandName": "Project",  "launchBrowser": true,  "launchUrl": "weatherforecast",  "environmentVariables": {  "ASPNETCORE\_ENVIRONMENT": "Development"  },  "applicationUrl": "https://10.9.2.109:5001;http://10.9.2.109:5000"  //"applicationUrl": "https://192.168.1.205:5001;http://192.168.1.205:5000"  }  }  } |

Còn lại ko đúng sẽ phát sinh lỗi



#### Run and Test

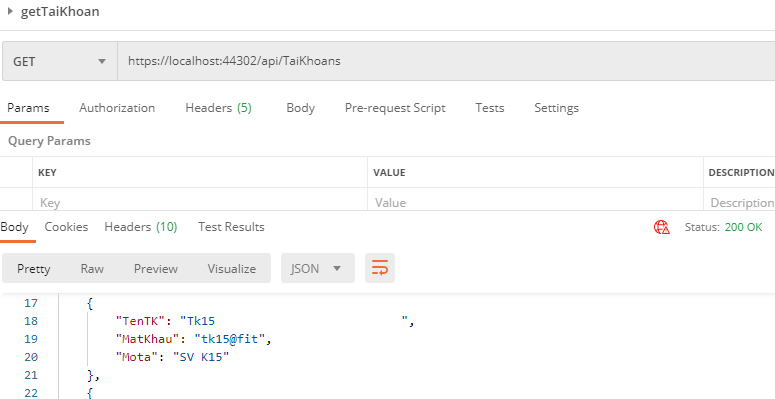


Chọn body và dữ liệu dạng raw, dữ liệu dạng json

|  |
| --- |
| {  "TenTK": "Tk15",  "MatKhau": "123",  "Mota": "Toàn bộ SV khóa K15"  } |

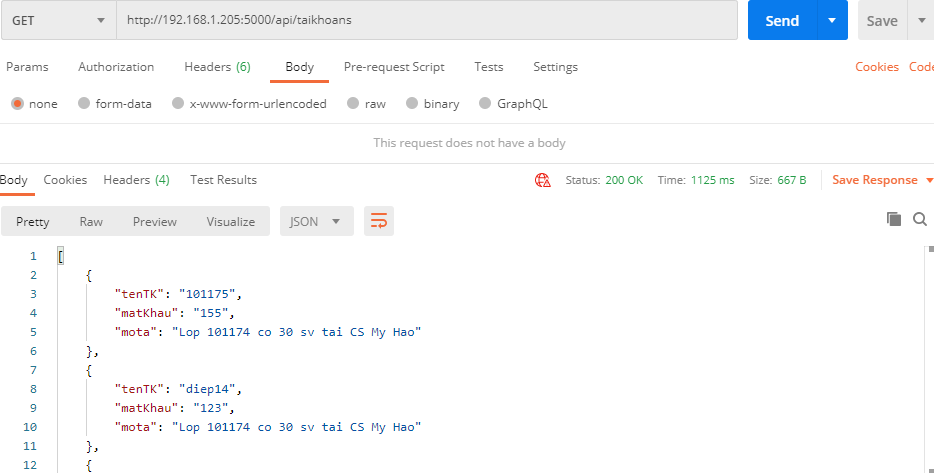
|  |  |
| --- | --- |
| {  "TenTK": "Tk15",  "MatKhau": "tk15@fit",  "Mota": "SV K15"  } |  |

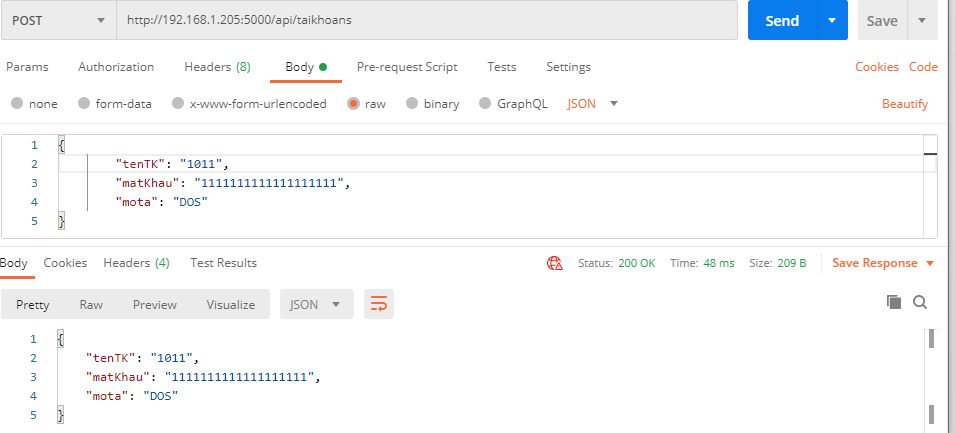
Kết quả sau khi update

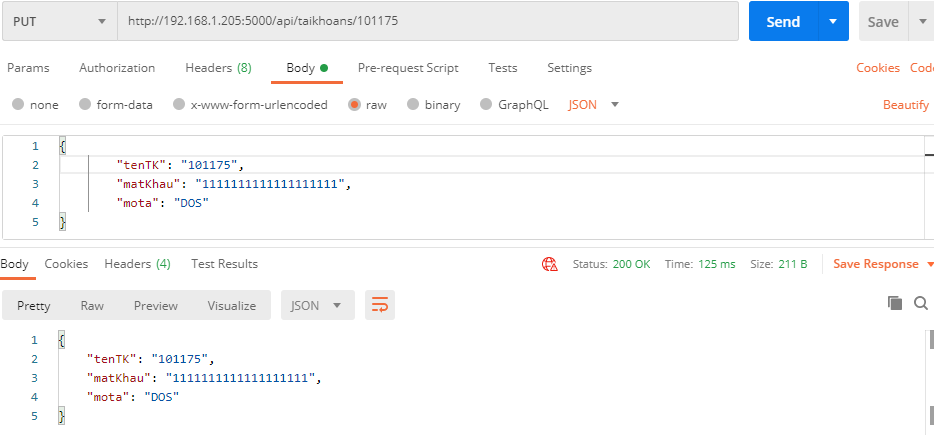


Khi thay đổi địa chỉ IP và cổng lắng nghe

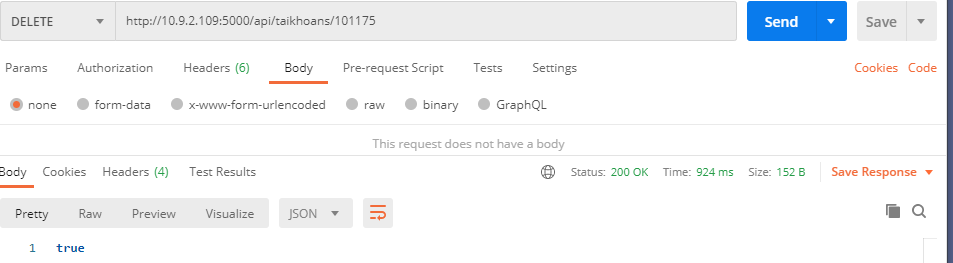
Trường hợp IP address của máy tính là 192.168.1.205 với http port 5000, https ports 5001







Trường hợp IP address của máy tính là 10.9.2.109 với http port 5000, https ports 5001



Bài tập tự làm

**Bài tập 1** Xây dựng ASP.NET WEB API Quản lý nhân viên (Thêm/sửa/xóa/tìm kiếm/thống kê thông tin tài khoản, nhân viên)

Gợi ý CSDL như sau:

|  |
| --- |
| CREATE TABLE ThongTinTaiKhoan(  TenTK char(30) PRIMARY KEY FOREIGN KEY REFERENCES TaiKhoan(TenTK),  HoTen nvarchar(30),  GioiTinh nchar(4),  NgaySinh date,  DiaChi nvarchar(30),  Email char(30),  SDT char(12)  )  INSERT INTO ThongTinTaiKhoan  VALUES ('diepnh',N'Nguyễn Hoàng Ðiệp', N'Nữ', '08/26/1984', N'CNPM/CNTT/ÐHSPKTHýngYên', 'Diep82003@gmail.com', '0964360105')  INSERT INTO ThongTinTaiKhoan  VALUES ('TK125','tk125','No','',N'CNPM/CNTT/ÐHSPKTHY','TK125@gmail.com','') |
| CREATE TABLE NHANVIEN(  maNV varchar(30) PRIMARY KEY,  TenNV nvarchar(50) NOT NULL,  GioiTinh nvarchar(30) default N'Nam',  NgaySinh date,  email varchar(30),  SDT char(30),  --ChiTiet nvarchar(100)  )  INSERT INTO NHANVIEN  VALUES ('diepnh',N'Nguyễn Hoàng Ðiệp', N'Nữ', '08/26/1984', 'Diep82003@gmail.com', '0964360105')  INSERT INTO NHANVIEN  VALUES ('nangnth',N'Nguyễn Thi Hải Năng', N'Nữ', '09/19/1984', 'hainangtk1@gmail.com', '0964360105')  INSERT INTO NHANVIEN  VALUES ('haunv',N'Nguyễn Văn Hậu', N'Nam', '06/06/1980', 'Diep82003@gmail.com', '')  INSERT INTO NHANVIEN  VALUES ('quyetnv',N'Nguyễn Văn Quyết', N'Nam', '01/6/1987', 'QuyetICT@gmail.com', '') |
| CREATE PROC ThemNHANVIEN (  @maNV varchar(30),  @TenNV nvarchar(50),  @GioiTinh nvarchar(30),  @NgaySinh date,  @email varchar(30),  @SDT char(30))  As  BEGIN  INSERT INTO NHANVIEN  VALUES (@maNV, @TenNV, @GioiTinh, @NgaySinh,@email, @SDT)  END;  EXEC ThemNHANVIEN 'tiennm',N'Nguyễn Minh Tiến', N'Nam', '07/22/1986', 'minhtien@gmail.com', '' |
| CREATE PROC SuaNHANVIEN (  @maNV varchar(30),  @TenNV nvarchar(50),  @GioiTinh nvarchar(30),  @NgaySinh date,  @email varchar(30),  @SDT char(30))  As  BEGIN  UPDATE NHANVIEN  SET TenNV =@TenNV ,  GioiTinh =@GioiTinh ,  NgaySinh =@NgaySinh ,  email =@email ,  SDT=@SDT  WHERE maNV=@maNV  END;  EXEC SuaNHANVIEN 'tiennm',N'Nguyễn Minh Tiến', N'Nam', '07/07/1986', 'minhtien@gmail.com', '0999999999' |
| CREATE PROC XoaNHANVIEN ( @maNV varchar(30))  As  BEGIN  DELETE FROM NHANVIEN  WHERE maNV=@maNV  END;  EXEC XoaNHANVIEN'diepnh' |

**Bài tập 2** Xây dựng ASP.NET WEB API, trong đó cơ sở dữ liệu QLSanPham gồm bảng DanhMuc và Bảng SanPham có mối quan hệ như sau:

