Configure Json File into StartUp Class

1.STEP:

Path location of json file: @"C:\FullPath\To\Your\Project\JSONFiles\"

FolderName:JSONFiles

Filename:Users.json

{

"UsersData": [

{

"FirstName": "John",

"LastName": "Doe",

"UserName": "john.doe",

"Email": "john.doe@example.com",

"Address": "123 Main St, Cityville",

"PhoneNo": "123-456-7890"

},

{

"FirstName": "Jane",

"LastName": "Smith",

"UserName": "jane.smith",

"Email": "jane.smith@example.com",

"Address": "456 Oak St, Townsville",

"PhoneNo": "987-654-3210"

},

{

"FirstName": "Jones",

"LastName": "dev",

"UserName": "jane.smith",

"Email": "jone@example.com",

"Address": "456 Oak St, Townsville",

"PhoneNo": "987-788-547"

},

{

"FirstName": "James",

"LastName": "Smith",

"UserName": "james.smith",

"Email": "james.smith@example.com",

"Address": "874 Oak St, Townsville",

"PhoneNo": "447-654-3210"

},

{

"FirstName": "Richardson",

"LastName": "jackson",

"UserName": "jane",

"Email": "jackson@example.com",

"Address": "541 Oak St, Townsville",

"PhoneNo": "874-654-3210"

},

{

"FirstName": "Peter",

"LastName": "son",

"UserName": "peterson",

"Email": "peter@example.com",

"Address": "741 Oak St, Townsville",

"PhoneNo": "874-654-3210"

}

]

}

2.Create Model Class

User.cs:

public class User

{

public string? FirstName { get; set; }

public string? LastName { get; set; }

public string? UserName { get; set; }

public string? Email { get; set; }

public string? Address { get; set; }

public string? PhoneNo { get; set; }

}

3.Implemented in StartUp Class

public class StartUp

{

private readonly IWebHostEnvironment \_env;

//private readonly IConfiguration \_configuration;

public IConfiguration configRoot { get; set; }

// constructor

public StartUp(IConfiguration configuration, IWebHostEnvironment env)

{

configRoot = configuration;

\_env = env;

}

public async void ConfigureServices(IServiceCollection services)

{

// ConfigJsonFiles

var path = Path.Combine(Environment.CurrentDirectory, @"JSONFiles\");

// for RSAClient.json

IConfiguration usersCongfig = new ConfigurationBuilder()

.SetBasePath(path)

.AddJsonFile("UsersData.json").Build();

// "UsersData" --- json file keyname

// User --- model class

var users = usersCongfig.GetSection("UsersData").Get<List<User>>();

services.AddSingleton(users);

}

public void Configure(WebApplication app,IWebHostEnvironment env)

{

app.UseSession();

app.UseRouting();

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI();

}

// use middleware for error handling

app.UseMiddleware<GlobalExceptionHandlingMiddleware>();

// add cors

app.UseCors(c=>c.AllowAnyHeader().AllowAnyOrigin().AllowAnyMethod());

app.UseHttpsRedirection();

// we have to must use for JWt authentication app.UseAuthentication();

app.UseAuthentication();

app.UseAuthorization();

app.MapControllers();

app.Run();

}

}

}

4.Usage In Contoller:

FileController.cs:

[Route("api/[controller]")]

[ApiController]

public class FilesController : ControllerBase

{

private readonly List<User> \_users;

public FilesController(

List<User> users

)

{

\_users = users; // load all jsonfile data here….

}

// config jsonfile from startup class

[HttpGet]

[Route("GetJsonFromConfigureService")]

public IActionResult GetJsonFromConfigureService()

{

try

{

var usersData = \_users;

if(usersData == null)

{

return NotFound("No json file data here...");

}

return Ok(new { data = usersData});

}

catch (Exception ex)

{

throw ex;

}

}

}