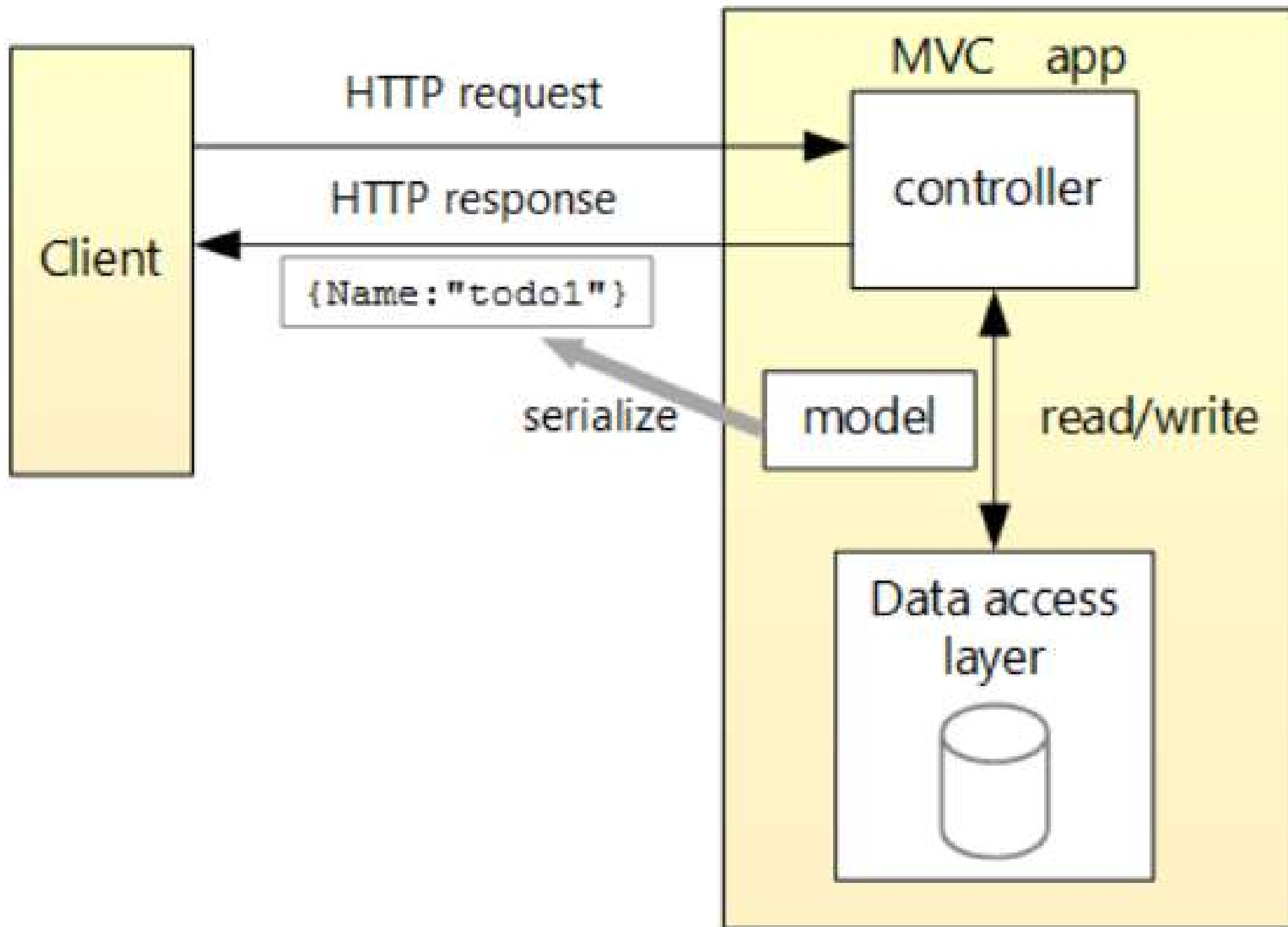


API	Description	Request body	Response body
GET /api/todoitems	Get all to-do items	None	Array of to-do items
GET /api/todoitems/{id}	Get an item by ID	None	To-do item
POST /api/todoitems	Add a new item	To-do item	To-do item
PUT /api/todoitems/{id}	Update an existing item	To-do item	None
DELETE /api/todoitems/{id}	Delete an item	None	None



# Visual Studio Installer

Installing — Visual Studio Community 2022

Workloads

Individual components

Language packs

Installation locations

 Need help choosing what to install? [More info](#)



## Web & Cloud (4)



### ASP.NET and web development

Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...



### Azure development

Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework....



### Python development

Editing, debugging, interactive development and source control for Python.



### Node.js development

Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.



## Desktop & Mobile (5)



### Mobile development with .NET

Build cross-platform applications for iOS, Android or Windows using Xamarin. This includes a preview of the ....



### .NET desktop development

Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...



# Visual Studio 2022

## Open recent

As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.

You can pin anything that you open frequently so that it's always at the top of the list.

## Get started



### Clone a repository

Get code from an online repository like GitHub or Azure DevOps



### Open a project or solution

Open a local Visual Studio project or .sln file



### Open a local folder

Navigate and edit code within any folder



### Create a new project

Choose a project template with code scaffolding to get started

[Continue without code →](#)

# Create a new project

## Recent project templates

A list of your recently accessed templates will be displayed here.

[Clear all](#)

All languages



All platforms



All project types



ASP.NET Core **Web API**

A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

C#

Linux

macOS

Windows

Cloud

Service

**Web**

**WebAPI**



ASP.NET Core **Web API**

A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

F#

Linux

macOS

Windows

Cloud

Service

**Web**

**WebAPI**



ASP.NET **Web** Application (.NET Framework)

Project templates for creating ASP.NET applications. You can create ASP.NET **Web** Forms, MVC, or **Web API** applications and add many other features in ASP.NET.

C#

Windows

Cloud

**Web**



ASP.NET **Web** Application (.NET Framework)

Project templates for creating ASP.NET applications. You can create ASP.NET **Web** Forms, MVC, or **Web API** applications and add many other features in ASP.NET.

Visual Basic

Windows

Cloud

**Web**



ASP.NET Core **Web** App

A project template for creating an ASP.NET Core application with example ASP.NET

Back

Next

# Configure your new project

ASP.NET Core Web API

C#

Linux

macOS

Windows

Cloud

Service

Web

WebAPI


Project name

ToDoApi

Location

C:\Users\CEDDIT\WSD\

...

Solution name 

ToDoApi

☐

Place solution and project in the same directory

# Additional information

## ASP.NET Core Web API

[C#](#)[Linux](#)[macOS](#)[Windows](#)[Cloud](#)[Service](#)[Web](#)[WebAPI](#)

Framework [i](#)

.NET 7.0 (Standard Term Support) ▼

Authentication type [i](#)

None ▼

☒ Configure for HTTPS [i](#)

☐ Enable Docker [i](#)

Docker QS [i](#)

Linux ▼

☒ Use controllers (uncheck to use minimal APIs) [i](#)

☒ Enable OpenAPI support [i](#)

☐ Do not use top-level statements [i](#)

[Back](#)[Create](#)

## Trust ASP.NET Core SSL Certificate



This project is configured to use SSL. To avoid SSL warnings in the browser you can choose to trust the self-signed certificate that ASP.NET Core has generated.

Would you like to trust the ASP.NET Core SSL certificate?

☐ Don't ask me again

Yes

No



## Security Warning



You are about to install a certificate from a certification authority (CA) claiming to represent:

localhost

Windows cannot validate that the certificate is actually from "localhost". You should confirm its origin by contacting "localhost". The following number will assist you in this process:

Thumbprint (sha1): 745D795B 51EB7E7E 98A11522 3B3ADAA4  
DEAE9D70

Warning:

If you install this root certificate, Windows will automatically trust any certificate issued by this CA. Installing a certificate with an unconfirmed thumbprint is a security risk. If you click "Yes" you acknowledge this risk.

Do you want to install this certificate?

Yes

No

```
var builder = WebApplication.CreateBuilder(args);
```

```
// Add services to the container.
```

```
builder.Services.AddControllers();
```

```
builder.Services.AddEndpointsApiExplorer();
```

```
builder.Services.AddSwaggerGen();
```

```
var app = builder.Build();
```

```
// Configure the HTTP request pipeline.
```

```
if (app.Environment.IsDevelopment())
```

```
{
```

```
    app.UseSwagger();
```

```
    app.UseSwaggerUI();
```

```
}
```

```
app.UseHttpsRedirection();
```

```
app.UseAuthorization();
```

```
app.MapControllers();
```

```
app.Run();
```

WeatherForecast.cs appsettings.json TodoItemsController.cs Program.cs TodoContext.cs TodoItem.cs

ToDoApi ToDoApi.WeatherForecast TemperatureF

namespace ToDoApi

{

2 references

public class WeatherForecast

{

1 reference

public DateOnly Date { get; set; }

2 references

public int TemperatureC { get; set; }

0 references

public int TemperatureF => 32 + (int)(TemperatureC / 0.5556);

1 reference

public string? Summary { get; set; }

}

}

WeatherForecastController.cs x WeatherForecast.cs appsettings.json TodoItemsController.cs Program.cs TodoContext.cs TodoItem.cs

ToDoApi

ToDoApi.Controllers.WeatherForecastController Get()

```
using Microsoft.AspNetCore.Mvc;

namespace ToDoApi.Controllers
{
    [ApiController]
    [Route("[controller]")]
    3 references
    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;

        0 references
        public WeatherForecastController(ILogger<WeatherForecastController> logger)
        {
            _logger = logger;
        }

        [HttpGet(Name = "GetWeatherForecast")]
        0 references
        public IEnumerable<WeatherForecast> Get()
        {
            return Enumerable.Range(1, 5).Select(index => new WeatherForecast
```



```
[HttpGet(Name = "GetWeatherForecast")]
```

0 references

```
public IEnumerable<WeatherForecast> Get()
```

```
{
```

```
    return Enumerable.Range(1, 5).Select(index => new WeatherForecast
```

```
    {
```

```
        Date = DateOnly.FromDateTime(DateTime.Now.AddDays(index)),
```

```
        TemperatureC = Random.Shared.Next(-20, 55),
```

```
        Summary = Summaries[Random.Shared.Next(Summaries.Length)]
```


```
    })
```

```
    .ToArray();
```

```
}
```

Swagger UI

https://localhost:7051/swagger/index.html

 **Swagger**  
Supported by SMARTBEAR

Select a definition

ToDoApi v1

# ToDoApi

1.0 OAS3

<https://localhost:7051/swagger/v1/swagger.json>

## WeatherForecast

GET

/WeatherForecast

Parameters

No parameters

Responses

Code	Description	Links
200	Success	No links

Media type

text/plain

Controls Accept header.

Try it out

Swagger UI

https://localhost:7051/swagger/index.html

Swagger

Select a definition

ToDoApi v1

Swagger UI

https://localhost:7051/swagge...

Swagger

Select a definition

ToDoApi v1

ToDoApi

1.0

OAS3

https://localhost:7051/swagger/v1/swagger.json

WeatherForecast

^

GET

/WeatherForecast

^

Parameters

Cancel

No parameters

Execute

Responses

Code	Description	Links
200	Success	No links

Swagger UI

https://localhost:7051/swagger... A ☆

```
'https://localhost:7051/WeatherForecast' \
-H 'accept: text/plain'
```

Request URL

https://localhost:7051/WeatherForecast

Server response

Code	Details
200	<p>Response body</p> <pre>[   {     "date": "2023-01-07",     "temperatureC": -15,     "temperatureF": 6,     "summary": "Warm"   },   {     "date": "2023-01-08",     "temperatureC": 23,     "temperatureF": 73,     "summary": "Chilly"   },   {     "date": "2023-01-09",     "temperatureC": 54,     "temperatureF": 129,     "summary": "Scorching"   },   {     "date": "2023-01-10",     "temperatureC": -9,     "temperatureF": 16,     "summary": "Cool"   },   {     "date": "2023-01-11",     "summary": "Cool"   } ]</pre> <p>Download</p>

Response headers

```
content-type: application/json; charset=utf-8
date: Fri, 06 Jan 2023 09:54:59 GMT
server: Kestrel
```





```
[{"date": "2023-01-07", "temperatureC": 26, "temperatureF": 78, "summary": "Cool"},  
{"date": "2023-01-08", "temperatureC": 31, "temperatureF": 87, "summary": "Freezing"},  
{"date": "2023-01-09", "temperatureC": 34, "temperatureF": 93, "summary": "Scorching"},  
{"date": "2023-01-10", "temperatureC": 46, "temperatureF": 114, "summary": "Mild"},  
{"date": "2023-01-11", "temperatureC": 21, "temperatureF": 69, "summary": "Freezing"}]
```

## Solution Explorer



Search Solution Explorer (Ctrl+;)



Solution 'ToDoApi' (1 of 1 project)



**ToDoApi**



Connected Services



Dependencies



Properties



Controllers



Models



TodoItem.cs



appsettings.json



Program.cs



WeatherForecast.cs

 namespace ToDoApi.Models

{

0 references

 public class TodoItem

{

0 references

public long Id { get; set; }

0 references

public string? Name { get; set; }

0 references

public bool IsComplete { get; set; }

}

}

Tools

Extensions

Window

Help

Search (Ctrl+Q)



ToDoApi

Get Tools and Features...

Manage Preview Features



Connect to Database...



Connect to Server...

SQL Server



Data Lake



Code Snippets Manager...

Ctrl+K, Ctrl+B

Choose Toolbox Items...

NuGet Package Manager



Package Manager Console



Manage NuGet Packages for Solution...



Package Manager Settings

Create GUID

External Command 2



WCF Service Configuration Editor

External Tools...

Theme



Command Line



Import and Export Settings...

Customize...



Options...

Browse

Installed

Updates

Consolidate

inmemory



Include prerelease

## Manage Packages for Solution

Package source: nuget.org



**Microsoft.EntityFrameworkCore.InMemory** by Microsoft, **140M** downloads  
In-memory database provider for Entity Framework Core (to be used for testing purposes).

7.0.1



**AspNetCore.HealthChecks.UI.InMemory.Storage** by Xabaril Contributors, **5.08M** downloads  
HealthChecks.UI.InMemory.Storage package contains the required classes to use InMemory provider in the UI

6.0.5



**HotChocolate.Subscriptions.InMemory** by ChilliCream authors and contributors, **6.37M** downloads  
Contains an in-memory implementation for a Hot Chocolate GraphQL subscription provider.

12.16.0



**InMemory** by Behzad Khosravifar, **36.7K** downloads  
Cache long time by auto refresh expire time and simple invocation rate checker

1.0.7

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.



Do not show this again



Microsoft.EntityFrameworkCore nuget.org

Versions - 0

<input checked="" type="checkbox"/>	Project	Version	Installed
<input checked="" type="checkbox"/>	ToDoApi		

Installed: not installed

Uninstall

Version: Latest stable 7.0.1

Install

Options

Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.

[Copy](#)

## ToDoApi

### Installing:

- Microsoft.EntityFrameworkCore.7.0.1
- Microsoft.EntityFrameworkCore.Abstractions.7.0.1
- Microsoft.EntityFrameworkCore.Analyzers.7.0.1
- Microsoft.EntityFrameworkCore.InMemory.7.0.1
- Microsoft.Extensions.Caching.Abstractions.7.0.0
- Microsoft.Extensions.Caching.Memory.7.0.0
- Microsoft.Extensions.DependencyInjection.7.0.0
- Microsoft.Extensions.DependencyInjection.Abstractions.7.0.0
- Microsoft.Extensions.Logging.7.0.0
- Microsoft.Extensions.Logging.Abstractions.7.0.0
- Microsoft.Extensions.Options.7.0.0
- Microsoft.Extensions.Primitives.7.0.0

☐ Do not show this again[OK](#)[Cancel](#)

The following package(s) require that you accept their license terms before installing.

**Microsoft.EntityFrameworkCore.Abstractions** Author(s): Microsoft  
MIT

**Microsoft.EntityFrameworkCore** Author(s): Microsoft  
MIT

**Microsoft.EntityFrameworkCore.Analyzers** Author(s): Microsoft  
MIT

**Microsoft.EntityFrameworkCore.InMemory** Author(s): Microsoft  
MIT

By clicking "I Accept," you agree to the license terms for the package(s) listed above. If you do not agree to the license terms, click "I Decline."

I Accept

I Decline

# Solution Explorer



Search Solution Explorer (Ctrl+;)



Solution 'ToDoApi' (1 of 1 project)

ToDoApi

Connected Services

Dependencies

Properties

Controllers

Models

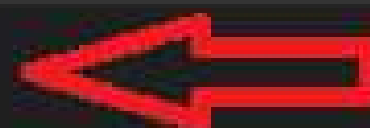
C# TodoContext.cs

C# TodoItem.cs

appsettings.json

C# Program.cs

C# WeatherForecast.cs





ToDoContext.cs\* X TodoItem.cs

ToDoApi

ToDoApi.Models.ToDoContext

ToDoItems

```
{  
    using Microsoft.EntityFrameworkCore;  
  
    namespace ToDoApi.Models  
    {  
        2 references  
        public class ToDoContext : DbContext  
        {  
            0 references  
            public ToDoContext(DbContextOptions<ToDoContext> options)  
                : base(options)  
            {  
            }  
            0 references  
            public DbSet<ToDoItem> ToDoItems { get; set; } = null!;  
        }  
    }  
}
```

```
using Microsoft.EntityFrameworkCore;  
using ToDoApi.Models;
```

```
var builder = WebApplication.CreateBuilder(args);
```

```
// Add services to the container.
```

```
builder.Services.AddControllers();
```

Dependency Injection

```
builder.Services.AddDbContext<TodoContext>(opt =>  
    opt.UseInMemoryDatabase("ToDoList"));
```

```
// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
```

```
builder.Services.AddEndpointsApiExplorer();
```

```
builder.Services.AddSwaggerGen();
```

```
var app = builder.Build();
```

```
// Configure the HTTP request pipeline.
```

```
if (app.Environment.IsDevelopment())
```

```
{
```

```
    app.UseSwagger();
```

```
    app.UseSwaggerUI();
```

```
}
```

```
app.UseHttpsRedirection();
```

```
app.UseAuthorization();
```

- Controller...
- Razor Component...
- New Item... Ctrl+Shift+A
- Existing Item... Shift+Alt+A
- New Scaffolded Item...
- New Folder
- Container Orchestrator Support...
- Docker Support...
- Application Insights Telemetry...
- Client-Side Library...
- Machine Learning Model...
- New Azure WebJob Project
- Existing Project as Azure WebJob
- Class...
- New EditorConfig

Search Solution Explorer (Ctrl+;)

Solution 'ToDoApi' (1 of 1 project)

- ToDoApi
  - Connected Services
  - Dependencies
  - Properties
  - Controllers

View in Browser (Firefox) Ctrl+Shift+W  
Browse With...

Add

Scope to This

New Solution Explorer View

Exclude From Project

- Cut Ctrl+X
- Copy Ctrl+C
- Delete Del
- Rename F2

- Copy Full Path
- Open Folder in File Explorer
- Open in Terminal

Properties Alt+Enter

# Add New Scaffolded Item

## Installed

### Common

API

#### MVC

Razor Component

Razor Pages

Identity

Layout



MVC Area



MVC Controller - Empty



MVC Controller with read/write actions



MVC Controller with views, using Entity Framework



API Controller - Empty



API Controller with read/write actions



API Controller with actions, using Entity Framework



API with read/write endpoints



API with read/write endpoints, using Entity Framework



Razor View - Empty



Razor View



Razor Component

### API Controller with actions, using Entity Framework

by Microsoft  
v1.0.0.0

An API controller with REST actions to create, read, update, delete, and list entities from an Entity Framework data context.

Id: ApiControllerWithContextScaffolder

Add

Cancel



## Add API Controller with actions, using Entity Framework

Model class

TodoItem (ToDoApi.Models)

Data context class

TodoContext (ToDoApi.Models)



Controller name

TodoItemsController

Add

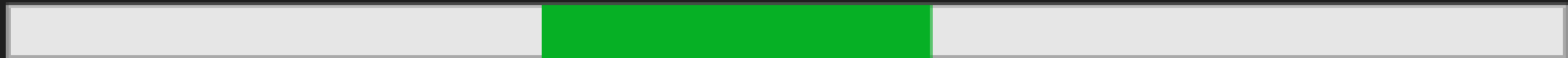
Cancel

Microsoft Visual Studio

Scaffolding...

Installing NuGet Packages. This may take a few moments.

Microsoft.EntityFrameworkCore.SqlServer



Microsoft Visual Studio



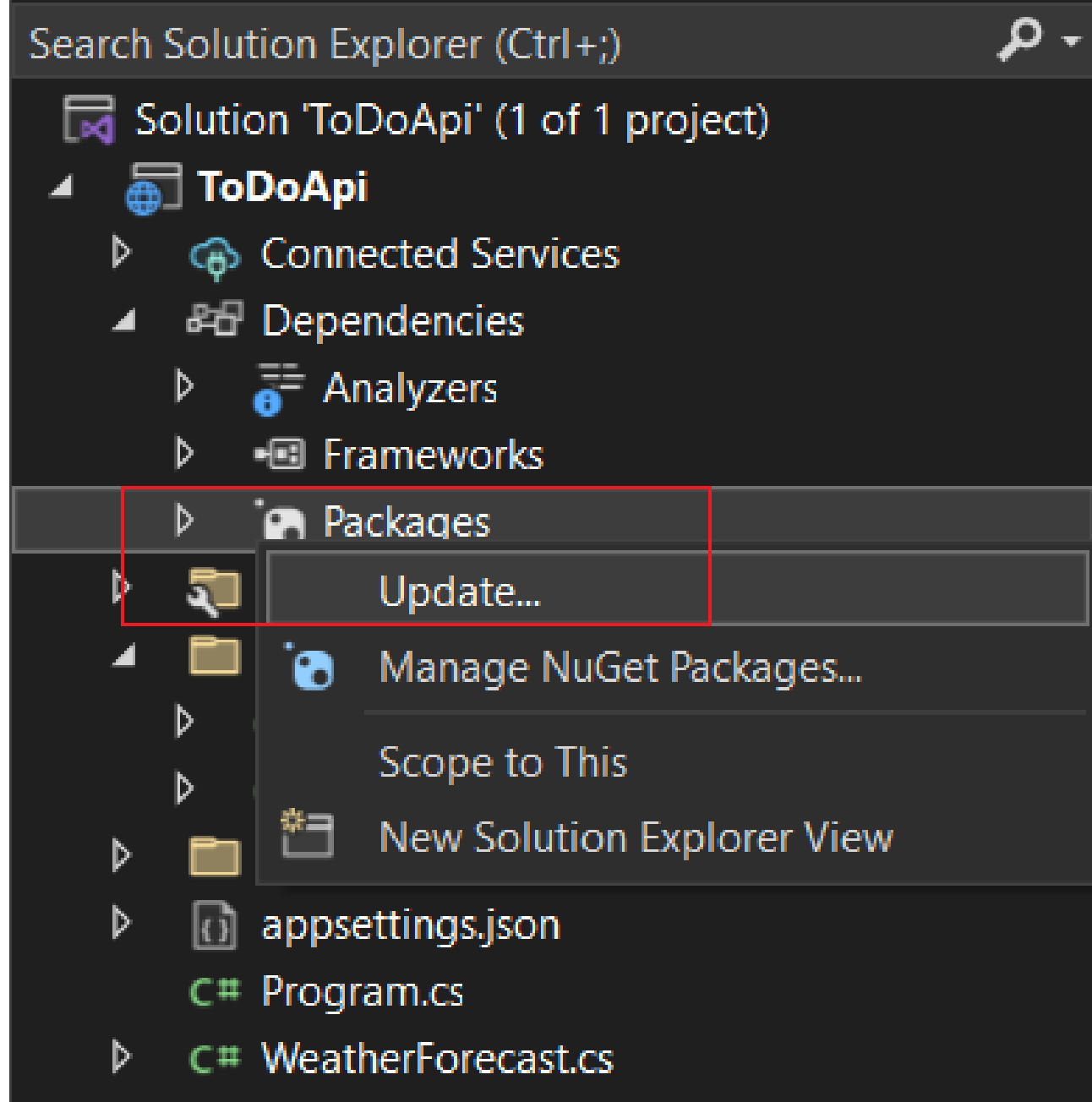
**select Add to try scaffolding a second time.**



Error

There was an error running the selected code generator:  
'Install the package  
Microsoft.VisualStudio.Web.CodeGeneration.Design and try again.'

OK



If previous solution doesn't work, try to update the packages.



```
namespace ToDoApi.Controllers
```

```
{
```

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

```
    [ApiController]
```

1 reference

```
    public class TodoItemsController : ControllerBase
```

```
    {
```

```
        private readonly TodoContext _context;
```

0 references

```
        public TodoItemsController(TodoContext context)
```

```
        {
```

```
            _context = context;
```

```
        }
```

```
        // GET: api/TodoItems
```

```
        [HttpGet]
```

0 references

```
        public async Task<ActionResult<IEnumerable<TodoItem>>> GetTodoItems()
```

```
        {
```

```
            if (_context.TodoItems == null)
```

```
            {
```

```
                return NotFound();
```

```
            }
```

```
            return await _context.TodoItems.ToListAsync();
```

```
        }
```

```
// GET: api/ToDoItems/5
```

```
[HttpGet("{id}")]
```

0 references

```
public async Task<ActionResult<ToDoItem>> GetToDoItem(long id)
```

```
{
```

```
    if (_context.ToDoItems == null)
```

```
    {
```

```
        return NotFound();
```

```
    }
```

```
    var todoItem = await _context.ToDoItems.FindAsync(id);
```

```
    if (todoItem == null)
```

```
    {
```

```
        return NotFound();
```

```
    }
```

```
    return todoItem;
```

```
}
```

```
// PUT: api/ToDoItems/5
```

```
[HttpPut("{id}")]
```

0 references

```
public async Task<IActionResult> PutToDoItem(long id, ToDoItem todoItem)
{
    if (id != todoItem.Id) {
        return BadRequest();
    }

    _context.Entry(todoItem).State = EntityState.Modified;

    try {
        await _context.SaveChangesAsync();
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!ToDoItemExists(id)) {
            return NotFound();
        }
        else {
            throw;
        }
    }
    return NoContent();
}
```

```
// PUT: api/ToDoItems/5
```

```
[HttpPut("{id}")]
```

0 references

```
public async Task<IActionResult> PutToDoItem(long id, ToDoItem todoItem)
{
    if (id != todoItem.Id) {
        return BadRequest();
    }

    _context.Entry(todoItem).State = EntityState.Modified;

    try {
        await _context.SaveChangesAsync();
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!ToDoItemExists(id)) {
            return NotFound();
        }
        else {
            throw;
        }
    }
    return NoContent();
}
```



```
// POST: api/ToDoItems
```

```
[HttpPost]
```

0 references

```
public async Task<ActionResult<ToDoItem>> PostToDoItem(ToDoItem todoItem)
```

```
{
```

```
    if (_context.ToDoItems == null)
```

```
    {
```

```
        return Problem("Entity set 'DbContext.ToDoItems' is null.");
```

```
    }
```

```
    _context.ToDoItems.Add(todoItem);
```

```
    await _context.SaveChangesAsync();
```

```
    return CreatedAtAction(nameof(GetToDoItem), new { id = todoItem.Id }, todoItem);
```

```
}
```

```
// DELETE: api/ToDoItems/5
```

```
[HttpDelete("{id}")]
```

0 references

```
public async Task<IActionResult> DeleteToDoItem(long id)
{
    if (_context.ToDoItems == null)
    {
        return NotFound();
    }

    var todoItem = await _context.ToDoItems.FindAsync(id);
    if (todoItem == null)
    {
        return NotFound();
    }

    _context.ToDoItems.Remove(todoItem);
    await _context.SaveChangesAsync();

    return NoContent();
}
```

1 reference

```
private bool TodoItemExists(long id)
{
    return (_context.TodoItems?.Any(e => e.Id == id)).GetValueOrDefault();
}
```



Select a definition

ToDoApi v1

# ToDoApi 1.0 OAS3

<https://localhost:7051/swagger/v1/swagger.json>

## Todoltems ^

GET

/api/TodoItems



POST

/api/TodoItems



GET

/api/TodoItems/{id}



PUT

/api/TodoItems/{id}



DELETE

/api/TodoItems/{id}



## WeatherForecast ^

GET

/WeatherForecast





POST

/api/ToDoItems



Parameters

Cancel

Reset

No parameters

Request body

application/json



```
{  
  "id": 1,  
  "name": "Buy Milk",  
  "isComplete": false  
}
```

Execute

## Responses

### Curl

```
curl -X 'POST' \
  'https://localhost:7051/api/ToDoItems' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 1,
    "name": "Buy Milk",
    "isComplete": false
  }'
```



### Request URL

<https://localhost:7051/api/ToDoItems>

### Server response

#### Code

#### Details

201

Undocumented

#### Response body

```
{
  "id": 1,
  "name": "Buy Milk",
  "isComplete": false
}
```



Download

#### Response headers

```
content-type: application/json; charset=utf-8
date: Mon, 16 Jan 2023 06:46:39 GMT
location: https://localhost:7051/api/ToDoItems/1
server: Kestrel
x-firefox-spdy: h2
```

GET

/api/ToDoItems



Parameters

Cancel

No parameters

Execute

GET

/api/ToDoItems/{id}



## Parameters

Cancel

Name

Description

**id** \* required

integer(\$int64)

(path)

1

Execute

# Responses

## Curl

```
curl -X 'GET' \  
  'https://localhost:7051/api/ToDoItems/1' \  
  -H 'accept: text/plain'
```



## Request URL

```
https://localhost:7051/api/ToDoItems/1
```

## Server response

### Code

### Details

200

### Response body

```
{  
  "id": 1,  
  "name": "Buy Milk",  
  "isComplete": false  
}
```



Download

### Response headers

```
content-type: application/json; charset=utf-8  
date: Mon, 16 Jan 2023 07:11:39 GMT  
server: Kestrel  
x-firefox-spdy: h2
```

GET

/api/ToDoItems/{id}



## Parameters

Cancel

Name

Description

id \* required

integer(\$int64)

(path)

2

Execute

Clear

# Responses

## Curl

```
curl -X 'GET' \  
  'https://localhost:7051/api/ToDoItems/2' \  
  -H 'accept: text/plain'
```



## Request URL

```
https://localhost:7051/api/ToDoItems/2
```

## Server response

### Code

### Details

404

*Undocumented* Error: Not Found

### Response body

```
{  
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.4",  
  "title": "Not Found",  
  "status": 404,  
  "traceId": "00-a27e542e63322bb674f100396d27431f-7e3b912d11e75c17-00"  
}
```



Download

### Response headers

```
content-type: application/problem+json; charset=utf-8  
date: Mon,16 Jan 2023 07:13:02 GMT  
server: Kestrel  
x-ff-spdy: h2
```

**PUT****/api/ToDoItems/{id}****Parameters****Cancel****Reset****Name****Description****id** \* required**integer(\$int64)**  
(path)

1

**Request body**

application/json



```
{  
  "id": 1,  
  "name": "Buy Bread",  
  "isComplete": true  
}
```



## Responses

### Curl

```
curl -X 'PUT' \
  'https://localhost:7051/api/ToDoItems/1' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 1,
    "name": "Buy Bread",
    "isComplete": true
  }
'
```



### Request URL

`https://localhost:7051/api/ToDoItems/1`

### Server response

#### Code

#### Details

204

*Undocumented*

#### Response headers

**No Content**

```
date: Mon, 16 Jan 2023 07:16:48 GMT
server: Kestrel
x-firefox-spdy: h2
```

PUT

/api/ToDoItems/{id}



## Parameters

Cancel

Reset

Name

Description

**id** \* required

integer(\$int64)

(path)

2

Request body

application/json



```
{
  "id": 1,
  "name": "Buy Bread",
  "isComplete": true
}
```

## Responses

### Curl

```
curl -X 'PUT' \  
  'https://localhost:7051/api/ToDoItems/2' \  
  -H 'accept: */*' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "id": 1,  
    "name": "Buy Bread",  
    "isComplete": true  
  }'  
,
```



### Request URL

https://localhost:7051/api/ToDoItems/2

### Server response

#### Code

#### Details

400

*Undocumented* Error: Bad Request

### Response body

```
{  
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",  
  "title": "Bad Request",  
  "status": 400,  
  "traceId": "00-33c680f3e592005a7737431d207c5bf0-524a3b0d17af8939-00"  
}
```



Download

### Response headers

```
content-type: application/problem+json; charset=utf-8  
date: Mon, 16 Jan 2023 07:20:06 GMT  
server: Kestrel  
x-firefox-spdy: h2
```

POST

/api/ToDoItems



Parameters

Cancel

Reset

No parameters

Request body

application/json



```
{  
  "id": 2,  
  "name": "HR Meeting",  
  "isComplete": true  
}
```

Execute

Clear

## Responses

### Curl

```
curl -X 'POST' \
  'https://localhost:7051/api/ToDoItems' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 2,
    "name": "HR Meeting",
    "isComplete": true
  }'
```



### Request URL

`https://localhost:7051/api/ToDoItems`

### Server response

#### Code

#### Details

201

*Undocumented*

#### Response body

```
{
  "id": 2,
  "name": "HR Meeting",
  "isComplete": true
}
```



Download

#### Response headers

```
content-type: application/json; charset=utf-8
date: Mon, 16 Jan 2023 07:22:50 GMT
location: https://localhost:7051/api/ToDoItems/2
server: Kestrel
x-firefox-spdy: h2
```

**GET**

**/api/ToDoItems**



**Parameters**

**Cancel**

No parameters

**Execute**

**Clear**

## Responses

### Curl

```
curl -X 'GET' \  
  'https://localhost:7051/api/ToDoItems' \  
  -H 'accept: text/plain'
```



### Request URL

```
https://localhost:7051/api/ToDoItems
```

### Server response

#### Code

#### Details

200

#### Response body

```
[  
  {  
    "id": 1,  
    "name": "Buy Bread",  
    "isComplete": true  
  },  
  {  
    "id": 2,  
    "name": "HR Meeting",  
    "isComplete": true  
  }  
]
```



Download

#### Response headers

```
content-type: application/json; charset=utf-8  
date: Mon, 16 Jan 2023 07:24:15 GMT  
server: Kestrel  
x-firefox-spdy: h2
```

**DELETE**`/api/ToDoItems/{id}`

## Parameters

[Cancel](#)


Name	Description
<b>id</b> * required	
integer(\$int64)	
(path)	

[Execute](#)



## Responses

### Curl

```
curl -X 'DELETE' \  
  'https://localhost:7051/api/ToDoItems/1' \  
  -H 'accept: */*' 
```

### Request URL

```
https://localhost:7051/api/ToDoItems/1
```

### Server response

Code	Details
------	---------

204	<b>No Content</b>
-----	-------------------

*Undocumented*

#### Response headers

```
date: Mon, 16 Jan 2023 07:25:44 GMT  
server: Kestrel  
x-firefox-spdy: h2
```

GET

/api/ToDoItems



Parameters

Cancel

No parameters

Execute

## Responses

### Curl

```
curl -X 'GET' \  
  'https://localhost:7051/api/ToDoItems' \  
  -H 'accept: text/plain'
```



### Request URL

```
https://localhost:7051/api/ToDoItems
```

### Server response

#### Code

#### Details

200

#### Response body

```
[  
  {  
    "id": 2,  
    "name": "HR Meeting",  
    "isComplete": true  
  }  
]
```



Download

#### Response headers

```
content-type: application/json; charset=utf-8  
date: Mon, 16 Jan 2023 07:27:06 GMT  
server: Kestrel  
x-ff-spdy: h2
```