



**Subject:** WEB SERVICE DEVELOPMENT  
**Semester:** 6  
**ID:** 20CEUOS052  
**Roll No:** CE100  
**Name:** Patel Pratham A.  
**Lab:** 6

## 1) Controllers for ToDoApi :-

### ❖ ToDoItemsController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using ToDoApi.Models;

namespace ToDoApi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class ToDoItemsController : ControllerBase
    {
        private readonly ToDoContext _context;

        public ToDoItemsController(ToDoContext context)
        {
            _context = context;
        }
    }
}
```

```

    }

    // GET: api/ToDoItems
    [HttpGet]
    public async Task<ActionResult<IEnumerable<ToDoItem>>> GetToDoItems()
    {
        return await _context.ToDoItems.ToListAsync();
    }

    // GET: api/ToDoItems/5
    [HttpGet("{id}")]
    public async Task<ActionResult<ToDoItem>> GetToDoItem(long id)
    {
        var toDoItem = await _context.ToDoItems.FindAsync(id);

        if (toDoItem == null)
        {
            return NotFound();
        }

        return toDoItem;
    }

    // PUT: api/ToDoItems/5
    // To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?linkid=2123754
    [HttpPut("{id}")]
    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
// To protect from overposting attacks, see
https://go.microsoft.com/fwlink/?Linkid=2123754
[HttpPost]
public async Task<ActionResult<ToDoItem>> PostToDoItem(ToDoItem
ToDoItem)
{
    _context.ToDoItems.Add(toDoItem);
    await _context.SaveChangesAsync();

    return CreatedAtAction("GetToDoItem", new { id = toDoItem.Id },
ToDoItem);
}

// DELETE: api/ToDoItems/5
[HttpDelete("{id}")]
public async Task<IActionResult> DeleteToDoItem(long id)
{
    var toDoItem = await _context.ToDoItems.FindAsync(id);
    if (toDoItem == null)
    {
        return NotFound();
    }

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

    return NoContent();
}

private bool ToDoItemExists(long id)
{
    return _context.ToDoItems.Any(e => e.Id == id);
}
}
}

```

## ❖ WeatherForecastController.cs

```
using Microsoft.AspNetCore.Mvc;

namespace ToDoApi.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(Name = "GetWeatherForecast")]
        public IEnumerable<WeatherForecast> Get()
        {
            return Enumerable.Range(1, 5).Select(index => new WeatherForecast
            {
                Date = DateTime.Now.AddDays(index),
                TemperatureC = Random.Shared.Next(-20, 55),
                Summary = Summaries[Random.Shared.Next(Summaries.Length)]
            })
                .ToArray();
        }
    }
}
```

## 2) Models of ToDoApi :-

### ❖ ToDoContext.cs

```
using Microsoft.EntityFrameworkCore;
namespace ToDoApi.Models
{
    public class ToDoContext : DbContext
    {
        public ToDoContext(DbContextOptions<ToDoContext> options) :
base(options)
        {

        }

        public DbSet<ToDoItem> ToDoItems { get; set; } = null!;
    }
}
```

### ❖ ToDoItem.cs

```
namespace ToDoApi.Models
{
    public class ToDoItem
    {
        public long Id { get; set; }

        public string? Name { get; set; }

        public bool IsComplete { get; set; }
    }
}
```

### 3) Program.cs file:-

```
using Microsoft.EntityFrameworkCore;
using ToDoApi.Models;
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();
//builder.Services.AddDbContext<ToDoContext>(opt =>
opt.UseInMemoryDatabase("ToDoList"));

builder.Services.AddDbContext<ToDoContext>(opt => opt.UseSqlServer("Data
Source=(LocalDB)\\MSSQLLocalDB;Initial Catalog=ToDoApi;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"));

// 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();

app.MapControllers();

app.Run();
```

### 4) WeatherForecast.cs file:-

```
namespace ToDoApi
{
    public class WeatherForecast
    {
```

```

public DateTime Date { get; set; }

public int TemperatureC { get; set; }

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

public string? Summary { get; set; }
}

```

## Outputs:

POST /api/ToDoItems

Parameters

No parameters

Request body

application/json

```
{
  "id": 1,
  "name": "See Movie",
  "isComplete": false
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  https://localhost:7286/api/ToDoItems \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 1,
    "name": "See Movie",
    "isComplete": false
  }'
```

Request URL

https://localhost:7286/api/ToDoItems

Server response

Code

Details

201

Undocumented

Response body

```
{
  "id": 1,
  "name": "See Movie",
  "isComplete": false
}
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 24 Jan 2023 08:45:54 GMT
location: https://localhost:7286/api/ToDoItems/1
server: Kestrel
```

GET

/api/ToDoItems/{id}

⌵

Parameters

Cancel

Name	Description
id * required integer(\$int64) (path)	<input type="text" value="1"/>

Execute

Clear

Responses

Curl

curl -X 'GET' \  
'https://localhost:7286/api/ToDoItems/1' \  
-H 'accept: text/plain'

Request URL

https://localhost:7286/api/ToDoItems/1

Server response

Code	Details
200	<div><div>Response body</div><div>{   "id": 1,   "name": "See Movie",   "isComplete": false }</div><div>Response headers</div><div>content-type: application/json; charset=utf-8 date: Tue, 24 Jan 2023 08:47:34 GMT server: Kestrel</div></div>

GET

/api/ToDoItems/{id}

⌵

Parameters

Cancel

Name	Description
id * required integer(\$int64) (path)	<input type="text" value="2"/>

Execute

Clear

Responses

Curl

curl -X 'GET' \  
'https://localhost:7286/api/ToDoItems/2' \  
-H 'accept: text/plain'

Request URL

https://localhost:7286/api/ToDoItems/2

Server response

Code	Details
404 <small>Undocumented</small>	<div><div>Error: response status is 404</div><div>Response body</div><div>{   "type": "https://tools.ietf.org/html/rfc7231section-6.5.4",   "title": "Not Found",   "status": 404,   "traceId": "00-407057eb09505a221be4592fc2d78aab-bf841e37e8681007-00" }</div><div>Response headers</div><div>content-type: application/problem+json; charset=utf-8 date: Tue, 24 Jan 2023 08:48:34 GMT server: Kestrel</div></div>



PUT

/api/ToDoItems/{id}

Parameters

Cancel

Reset

Name	Description
<b>id</b> <small>required</small> integer(\$int64) (path)	<input type="text" value="1"/>

Request body

application/json

```
{
  "id": 1,
  "name": "See Movie",
  "isComplete": true
}
```

Execute

Clear

Responses

Curl

```
curl -X 'PUT' \
  'https://localhost:7286/api/ToDoItems/1' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 1,
    "name": "See Movie",
    "isComplete": true
  }'
```

Request URL

https://localhost:7286/api/ToDoItems/1

Server response

Code	Details
204 <small>undocumented</small>	<div>Response headers<div>date: Tue, 24 Jan 2023 08:58:05 GMT server: Kestrel</div></div>

Responses

Code	Description	Links
200	Success	No links

PUT

/api/ToDoItems/{id}

Parameters

Cancel

Reset

Name	Description
<b>id</b> <small>required</small> integer(\$int64) (path)	<input type="text" value="2"/>

Request body

application/json

```
{
  "id": 1,
  "name": "See Movie",
  "isComplete": true
}
```

Execute

Clear

## Responses

### Curl

```
curl -X 'PUT' \
  'https://localhost:7286/api/ToDoItems/2' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 2,
    "name": "See Movie",
    "isComplete": true
  }'
```

### Request URL

https://localhost:7286/api/ToDoItems/2

### Server response

Code Details

400

Undocumented

Error: response status is 400

### Response body

```
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "Bad Request",
  "status": 400,
  "traceId": "00-d0614bd39059f6e2ede73e159d6ce34f-0bc4356dc86145d3-00"
}
```

### Response headers

```
content-type: application/problem+json; charset=utf-8
date: Tue, 24 Jan 2023 08:51:18 GMT
server: Kestrel
```

### Responses

Code Description

Links

200

Success

No links

POST /api/ToDoItems

### Parameters

Cancel

Reset

No parameters

### Request body

application/json

```
{
  "id": 2,
  "name": "Buy Popcorns",
  "isComplete": true
}
```

Execute

Clear

## Responses

### Responses

### Curl

```
curl -X 'POST' \
  'https://localhost:7286/api/ToDoItems' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 2,
    "name": "Buy Popcorns",
    "isComplete": true
  }'
```

### Request URL

https://localhost:7286/api/ToDoItems

### Server response

Code Details

201

Undocumented

### Response body

```
{
  "id": 2,
  "name": "Buy Popcorns",
  "isComplete": true
}
```

### Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 24 Jan 2023 08:52:08 GMT
location: https://localhost:7286/api/ToDoItems/2
server: Kestrel
```

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  "https://localhost:7286/api/ToDoItems" \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7286/api/ToDoItems

Server response

Code

Details

200

Response body

```
{
  "id": 2,
  "name": "Buy Popcorns",
  "isComplete": true
},
{
  "id": 1,
  "name": "See Movie",
  "isComplete": true
}
]
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 24 Jan 2023 08:52:36 GMT
server: Kestrel
```

DELETE /api/ToDoItems/{id}

^

Parameters

Cancel

Name

Description

id \* required

integer(\$int64)

(path)

1

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
  "https://localhost:7286/api/ToDoItems/1" \
  -H 'accept: */*'
```

Request URL

https://localhost:7286/api/ToDoItems/1

Server response

Code

Details

204

Response headers

```
date: Tue, 24 Jan 2023 08:53:01 GMT
server: Kestrel
```

GET

/api/ToDoItems

^

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7286/api/ToDoItems' \
  -H 'accept: text/plain'
```

Request URL

```
https://localhost:7286/api/ToDoItems
```

Server response

CodeDetails

200

Response body

```
{
  "Id": 2,
  "Name": "Buy Popcorns",
  "IsComplete": true
}
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 24 Jan 2023 08:53:25 GMT
server: Kestrel
```

