**1. Project Structure**

dotnetapp/

│

├── Controllers/

│ └── MobilePhoneController.cs

│

├── Models/

│ └── MobilePhone.cs

│

├── Repository/

│ └── MobilePhoneRepository.cs

│

├── Services/

│ ├── IMobilePhoneService.cs

│ └── MobilePhoneService.cs

│

├── Program.cs

└── dotnetapp.csproj

**2. Models**

**Models/MobilePhone.cs**

namespace dotnetapp.Models

{

public class MobilePhone

{

public int MobilePhoneId { get; set; }

public string Brand { get; set; }

public string Model { get; set; }

public decimal Price { get; set; }

public int StockQuantity { get; set; }

}

}

**3. Repository**

**Repository/MobilePhoneRepository.cs**

using dotnetapp.Models;

namespace dotnetapp.Repository

{

public class MobilePhoneRepository

{

private readonly List<MobilePhone> \_mobilePhones = new();

private int \_idCounter = 1;

public List<MobilePhone> GetAll()

{

return \_mobilePhones;

}

public MobilePhone GetById(int id)

{

return \_mobilePhones.FirstOrDefault(m => m.MobilePhoneId == id);

}

public MobilePhone Add(MobilePhone mobilePhone)

{

mobilePhone.MobilePhoneId = \_idCounter++;

\_mobilePhones.Add(mobilePhone);

return mobilePhone;

}

public bool Update(int id, MobilePhone mobilePhone)

{

var existing = \_mobilePhones.FirstOrDefault(m => m.MobilePhoneId == id);

if (existing == null) return false;

existing.Brand = mobilePhone.Brand;

existing.Model = mobilePhone.Model;

existing.Price = mobilePhone.Price;

existing.StockQuantity = mobilePhone.StockQuantity;

return true;

}

public bool Delete(int id)

{

var existing = \_mobilePhones.FirstOrDefault(m => m.MobilePhoneId == id);

if (existing == null) return false;

\_mobilePhones.Remove(existing);

return true;

}

}

}

**4. Service Interface**

**Services/IMobilePhoneService.cs**

using dotnetapp.Models;

namespace dotnetapp.Services

{

public interface IMobilePhoneService

{

List<MobilePhone> GetMobilePhones();

MobilePhone GetMobilePhone(int id);

MobilePhone SaveMobilePhone(MobilePhone mobilePhone);

MobilePhone UpdateMobilePhone(int id, MobilePhone mobilePhone);

bool DeleteMobilePhone(int id);

}

}

**5. Service Implementation**

**Services/MobilePhoneService.cs**

using dotnetapp.Models;

using dotnetapp.Repository;

namespace dotnetapp.Services

{

public class MobilePhoneService : IMobilePhoneService

{

private readonly MobilePhoneRepository \_repository;

public MobilePhoneService(MobilePhoneRepository repository)

{

\_repository = repository;

}

public List<MobilePhone> GetMobilePhones()

{

return \_repository.GetAll();

}

public MobilePhone GetMobilePhone(int id)

{

return \_repository.GetById(id);

}

public MobilePhone SaveMobilePhone(MobilePhone mobilePhone)

{

return \_repository.Add(mobilePhone);

}

public MobilePhone UpdateMobilePhone(int id, MobilePhone mobilePhone)

{

var updated = \_repository.Update(id, mobilePhone);

return updated ? mobilePhone : null;

}

public bool DeleteMobilePhone(int id)

{

return \_repository.Delete(id);

}

}

}

**6. Controller**

**Controllers/MobilePhoneController.cs**

using Microsoft.AspNetCore.Mvc;

using dotnetapp.Models;

using dotnetapp.Services;

namespace dotnetapp.Controllers

{

[ApiController]

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

public class MobilePhoneController : ControllerBase

{

private readonly IMobilePhoneService \_service;

public MobilePhoneController(IMobilePhoneService service)

{

\_service = service;

}

[HttpGet]

public IActionResult GetAllMobilePhones()

{

var phones = \_service.GetMobilePhones();

return Ok(phones); // 200 OK

}

[HttpGet("{id}")]

public IActionResult GetMobilePhoneById(int id)

{

var phone = \_service.GetMobilePhone(id);

if (phone == null) return NotFound(); // 404

return Ok(phone); // 200 OK

}

[HttpPost]

public IActionResult AddMobilePhone([FromBody] MobilePhone mobilePhone)

{

if (mobilePhone == null) return BadRequest(); // 400

var createdPhone = \_service.SaveMobilePhone(mobilePhone);

return CreatedAtAction(nameof(GetMobilePhoneById), new { id = createdPhone.MobilePhoneId }, createdPhone); // 201

}

[HttpPut("{id}")]

public IActionResult UpdateMobilePhone(int id, [FromBody] MobilePhone mobilePhone)

{

if (mobilePhone == null) return BadRequest(); // 400

var updated = \_service.UpdateMobilePhone(id, mobilePhone);

if (updated == null) return NotFound(); // 404

return NoContent(); // 204

}

[HttpDelete("{id}")]

public IActionResult DeleteMobilePhone(int id)

{

var deleted = \_service.DeleteMobilePhone(id);

if (!deleted) return NotFound(); // 404

return NoContent(); // 204

}

}

}

**7. Program.cs (Dependency Injection & Swagger)**

**Program.cs**

using dotnetapp.Repository;

using dotnetapp.Services;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container

builder.Services.AddSingleton<MobilePhoneRepository>();

builder.Services.AddScoped<IMobilePhoneService, MobilePhoneService>();

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

var app = builder.Build();

app.UseSwagger();

app.UseSwaggerUI(c =>

{

c.SwaggerEndpoint("/swagger/v1/swagger.json", "MobilePhone API V1");

});

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run("http://localhost:8080");

**8. Status Codes Summary**

| **Operation** | **Status Code** |
| --- | --- |
| Successful retrieval (GET) | 200 OK |
| Successful creation (POST) | 201 Created |
| Successful update (PUT) | 204 No Content |
| Successful deletion (DELETE) | 204 No Content |
| Invalid request (missing fields) | 400 Bad Request |
| Item not found (GET/PUT/DELETE) | 404 Not Found |
| Unsupported HTTP method | 405 Method Not Allowed |