[CSW Documentation](https://www.udemy.com/course/the-complete-guide-to-angular-2/)

Contents

[1 Architecture 3](#_Toc170310025)

[1.1 Clean Architecture 3](#_Toc170310026)

[1.1.1 Domain 3](#_Toc170310027)

[1.1.2 Application 3](#_Toc170310028)

[1.1.3 Presentation 3](#_Toc170310029)

[1.1.4 Infrastructure 3](#_Toc170310030)

[2 Data 4](#_Toc170310031)

[2.1 Scaffolding 4](#_Toc170310032)

[2.2 Migrations 4](#_Toc170310033)

[3 CQRS + MediaR 5](#_Toc170310034)

[3.1 CQRS Intro 5](#_Toc170310035)

[3.2 CQRS Implementation 6](#_Toc170310036)

[4 Retrieving & Creating Entities 7](#_Toc170310037)

[5 Swagger 9](#_Toc170310038)

[6 ASP.NET Core Identity 10](#_Toc170310039)

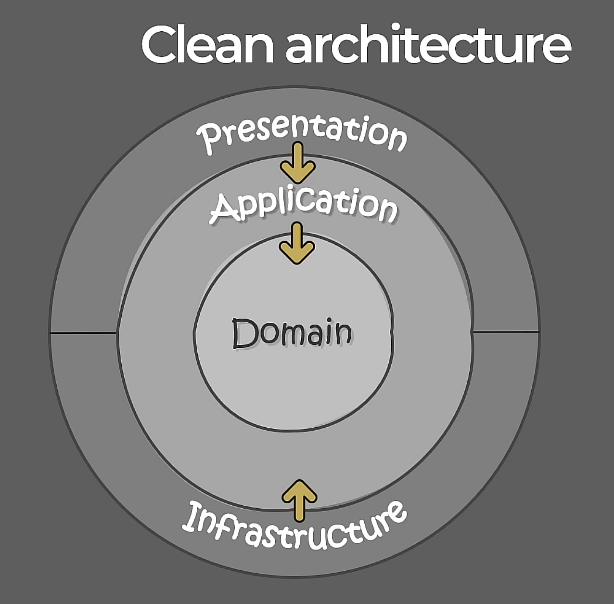
[7 Exception Handling 12](#_Toc170310040)

[8 To Do 13](#_Toc170310041)

# Architecture

## Clean Architecture

Created by Uncle Bob (Robert Martin.

Most common architecture in .NET projects

### Domain

Entity or core module that encapsulates the business logic.

### Application

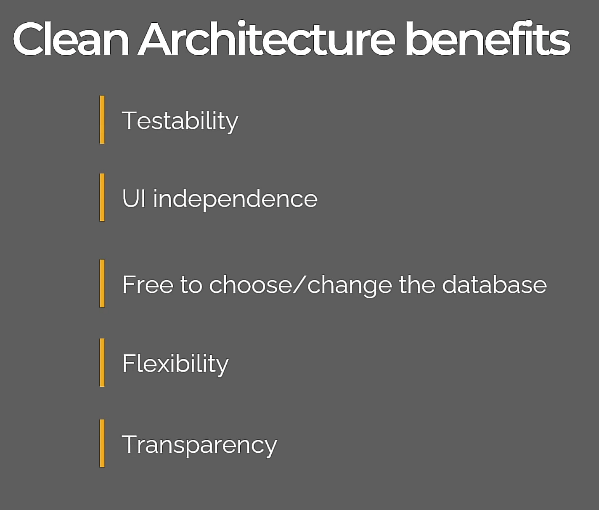
Use cases modules.

Application logic, e.g. reacting to a request received by our system from the user.

### Presentation

e.g. a website, a web API interface, …

### Infrastructure

Contains e.g. code for connection to DB, uploading files, …

# Data

## Scaffolding

Run command in Csw.Infrastructure project:

dotnet ef dbcontext scaffold "Server=tcp:fm6oz44eml.database.windows.net,1433;Initial Catalog=centraldb;Persist Security Info=False;User ID=dba;Password=Cswcswcsw1;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;" Microsoft.EntityFrameworkCore.SqlServer --table [dbo].[animal] --table [dbo].[animal\_action] --table [dbo].[animal\_action\_type] --table [dbo].[animal\_breed] --table [dbo].[animal\_coat] --table [dbo].[animal\_color] --table [dbo].[animal\_final\_certificate\_print\_log] --table [dbo].[animal\_organisation\_association] --table [dbo].[animal\_owner\_log] --table [dbo].[country] --table [dbo].[email\_campaign] --table [dbo].[email\_campaign\_unique\_id] --table [dbo].[identification] --table [dbo].[identification\_localisation] --table [dbo].[identification\_type] --table [dbo].[identifier] --table [dbo].[mailjet\_status] --table [dbo].[mailjet\_temp] --table [dbo].[organisation] --table [dbo].[owner] --table [dbo].[owner\_address\_log] --table [dbo].[passport] --table [dbo].[postal\_address] --table [dbo].[postal\_code] --table [dbo].[ref\_animal\_color] --table [dbo].[ref\_breed\_name] --table [dbo].[ref\_coat\_name] --table [dbo].[ref\_species] --table [dbo].[ref\_species\_name] --table [dbo].[title] --table [dbo].[user] --table [dbo].[language] --table [dbo].[association] --context-dir Persistence --output-dir ../Csw.Domain/Entities --force

## Migrations

Set startup project to Idc.API

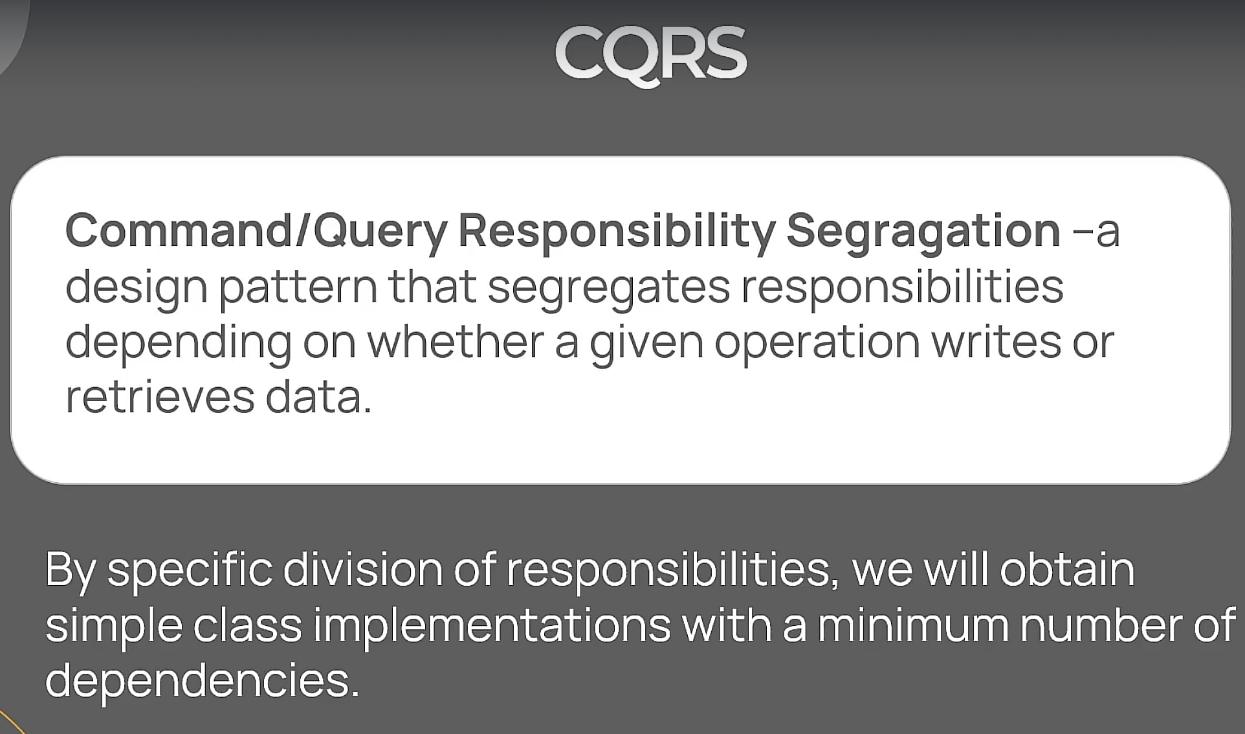
Type command in tools/nuget package manager console with default project = csw.infrastructure

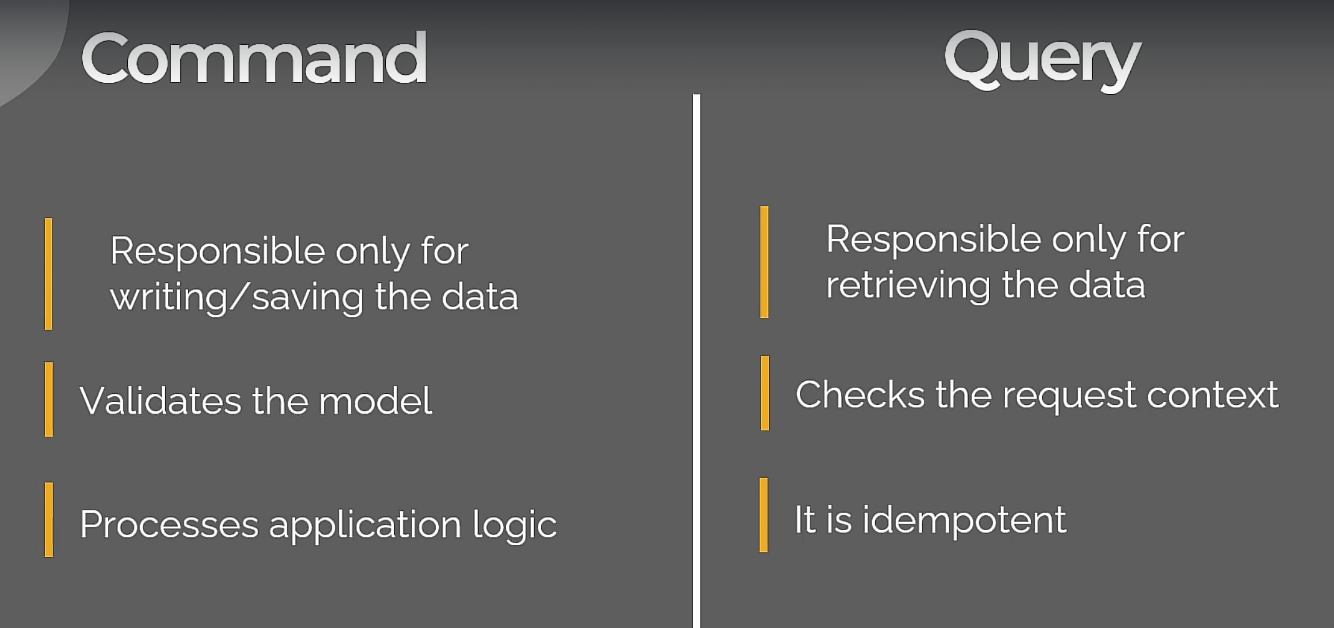
Add-Migration InitialCreate

update-database

# CQRS + MediaR

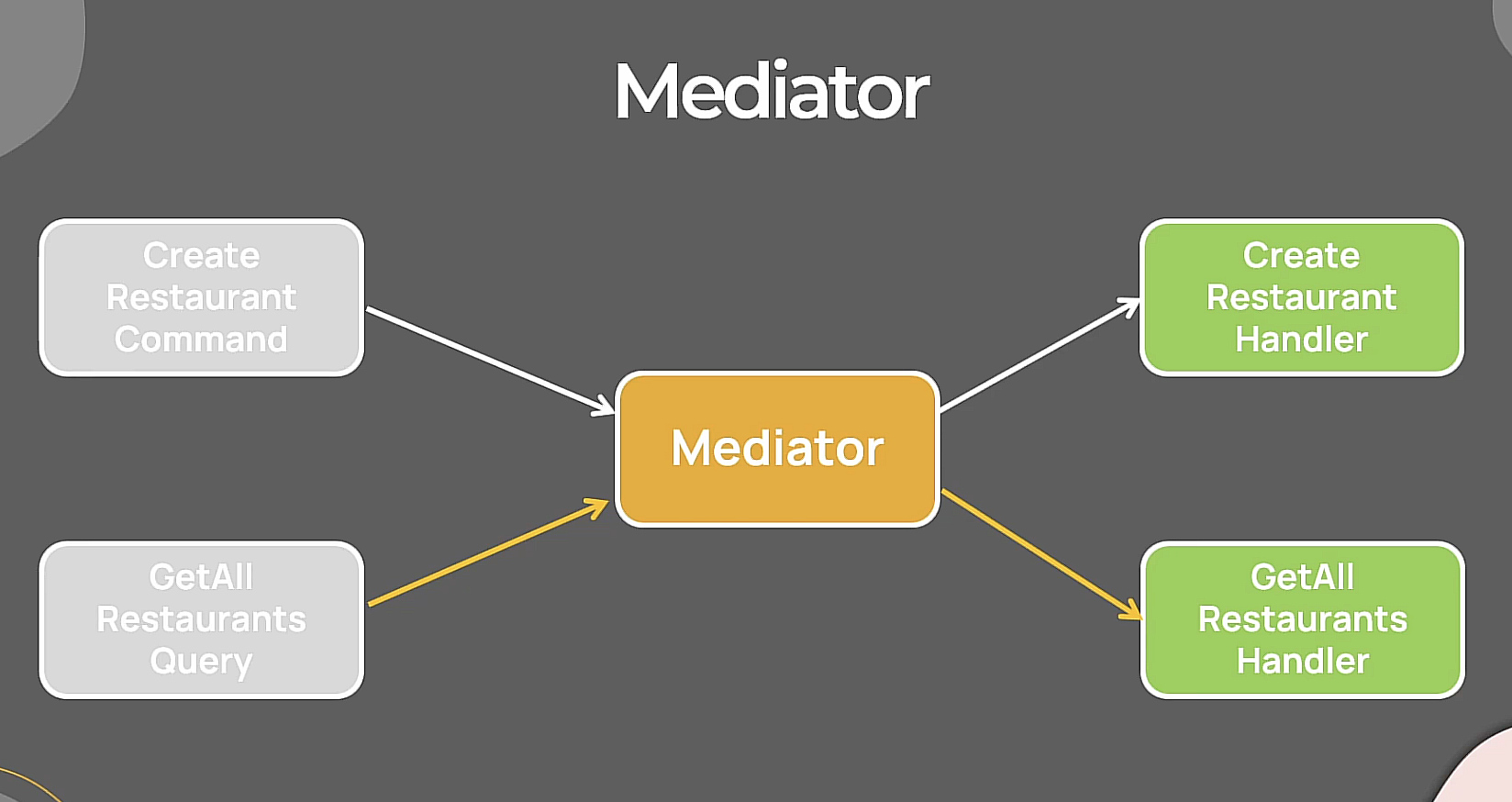
## CQRS Intro





An idempotent operation is one where a request can be retransmitted or retried with no additional side effects, a property that is very beneficial in distributed systems.

## CQRS Implementation



# Retrieving & Creating Entities

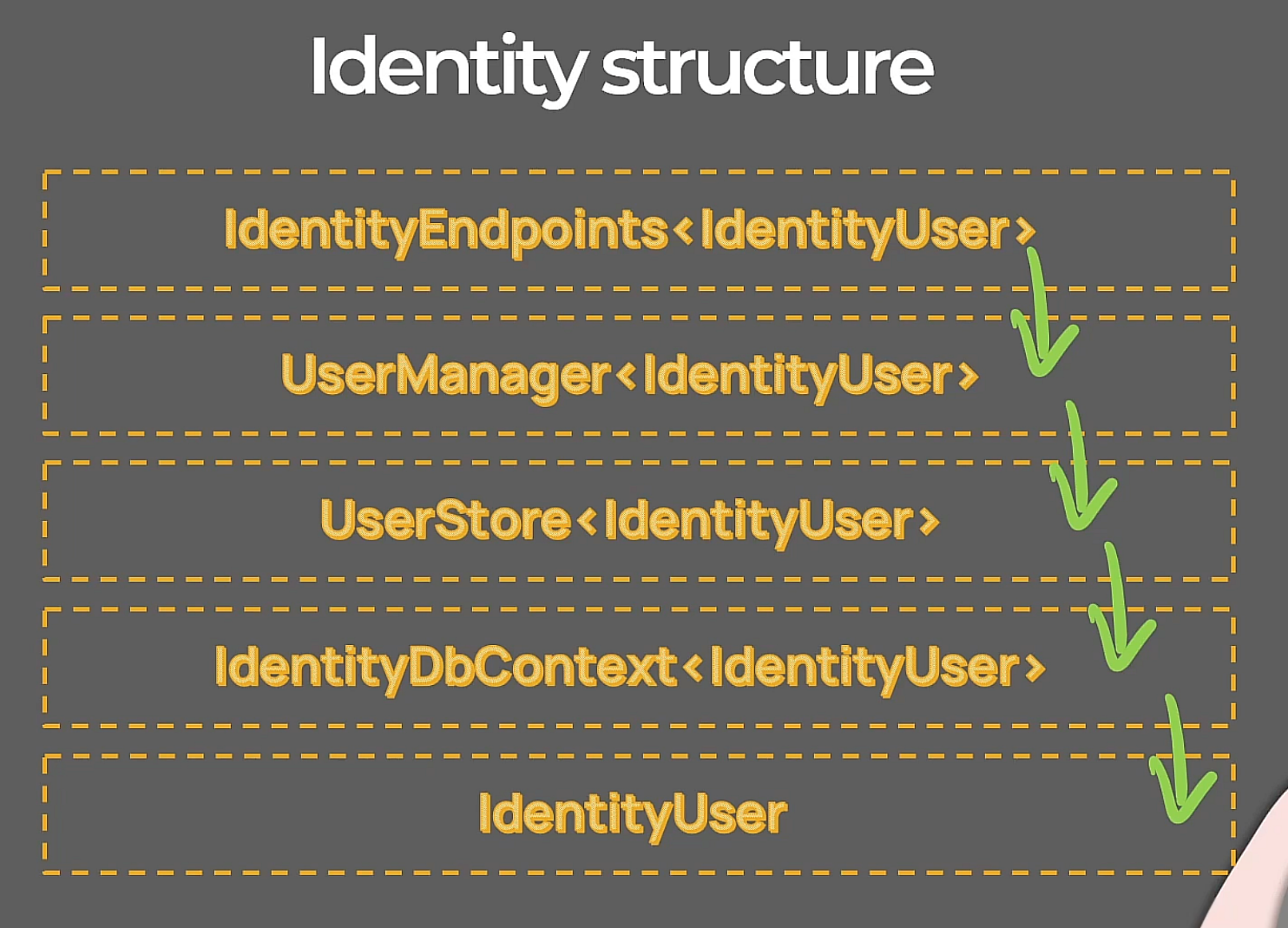
Example with getting Identifiers

1. One Time:
   1. Install AutoMapper package
   2. Inject AutoMapper in CSW.Application\Extensions\ServiceCollectionExtensions.cs
   3. Install FluentValidation.AspNetCore in Application
   4. Inject AddValidatorsFromAssembly in CSW.Application\Extensions\ServiceCollectionExtensions.cs
   5. Install MediatR package in Application
2. Create Dto in CSW.Application\Identifiers\Dtos\
3. Create Csw.Domain\Repositories\IIdentifiersRepository.cs
4. Create Csw.Infrastructure\Repositories\IdentifiersRepository.cs
5. Inject services.AddScoped<IIdentifiersRepository, IdentifierRepository>(); in   
   Csw.Infrastructure\Extensions\ServiceCollectionExtensions.cs
6. Create CSW.Application\Identifiers\Commands\CreateIdentifier\CreateIdentifierCommand.cs
7. Create CSW.Application\Identifiers\Commands\CreateIdentifier\CreateIdentifierCommandHandler.cs
8. Create CSW.Application\Identifiers\Commands\CreateIdentifier\CreateIdentifierCommandValidator.cs
9. Create CSW.Application\Identifiers\Commands\UpdateIdentifier\UpdateIdentifierCommand.cs
10. Create CSW.Application\Identifiers\Commands\UpdateIdentifier\UpdateIdentifierCommandHandler.cs
11. Create CSW.Application\Identifiers\Commands\UpdateIdentifier\UpdateIdentifierCommandValidator.cs
12. Create CSW.Application\Countries\Commands\DeleteIdentifier\DeleteIdentifierHandlerCommand.cs
13. Create CSW.Application\Identifiers\Commands\DeleteIdentifier\DeleteIdentifierCommand.cs
14. Create CSW.Application\Identifiers\Queries\GetAllIdentifiers\GetAllIdentifiersQuery.cs
15. Create CSW.Application\Identifiers\Queries\GetAllIdentifiers\GetAllIdentifiersQueryHandler.cs
16. Create CSW.Application\Identifiers\Queries\GetIdentifierById\GetIdentifierByIdQuery.cs
17. Create CSW.Application\Identifiers\Queries\GetIdentifierById\GetIdentifierByIdQueryHandler.cs
18. Create Profile in CSW.Application\Identifiers\Dtos\
19. Create/Update Idc.API\Controllers\IdentifiersController.cs

# Swagger

<https://localhost:7125/swagger>   
to play with API

# A screenshot of a computer Description automatically generatedASP.NET Core Identity



# Exception Handling

Create generic try catch

* Create Idc.API\MiddleWares\ErrorHandlingMiddle.cs
* Add builder.Services.AddScoped<ErrorHandlingMiddleWare>(); and  
  app.UseMiddleware<ErrorHandlingMiddleWare>();   
  in  
  Idc.API\Program.cs (One time)
* Create exceptions in Csw.Domain\Exceptions\

# To Do

* Validation in API