# Data Configuration

Data/Table Configuration in Entity Framework Core (EF Core) refers to the way you define and configure the mapping between your .NET model classes and the database tables. This includes specifying how the columns in the tables correspond to the properties in your classes, setting up primary and foreign keys, configuring relationships, and applying constraints.

### Ways to Configure Entities in EF Core

1. **Data Annotations:** These are attributes that you can apply directly to your model classes and properties to configure the database schema.
2. **Fluent API:** This provides a more powerful and flexible way to configure your model. Fluent API configuration is done in the OnModelCreating method of your DbContext class.

# EF-Migration (Add, Revert):

* If we want to revert our Migration before Appling to database, we can easily revert it by using the command **Remove-Migration.**
* If the changes are applied to the database, then you want to revert the migration and the changes applied by this migration in database then you will follow the following steps:

1. First you apply the **Update-Database <Previous Migration on which you want to revert it again>.**
2. It will update the update the database according to previous migration.
3. Then you enter the command **Remove-Migration** it will remove your last migration.
4. Revert migration can revert all last migrations ono by one which has no effect on the database.

# DTOs (Data Transfer Object)

public static StudentDTO ToStudentDto( this Student StudentModel )

the this keyword in the parameter of the ToStudentDto method makes it an extension method, allowing it to be called as if it were an instance method on the Student class.

* By using dtos you can only return the properties that are necessary to display to the user.
* The unnecessary and sensitive data can not display to the user.

# Loggers

The default ASP.NET Core web app templates call [WebApplication.CreateBuilder](https://learn.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.builder.webapplication.createbuilder), which adds the following logging providers:

* [Console](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/logging/?view=aspnetcore-8.0#console)
* [Debug](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/logging/?view=aspnetcore-8.0#debug)
* [EventSource](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/logging/?view=aspnetcore-8.0#event-source)
* [EventLog](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/logging/?view=aspnetcore-8.0#welog): Windows only

## Log Levels in Web API:

1. Trace.
2. Debug.
3. Information.
4. Warning.
5. Error.
6. Critical.
7. None.

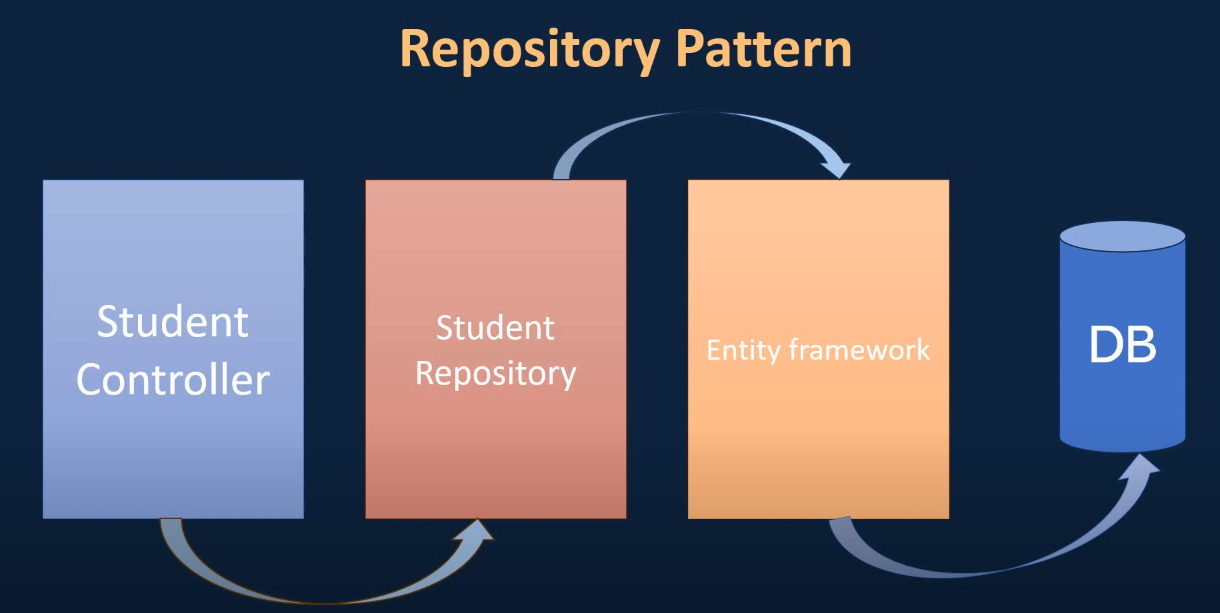
# Validation

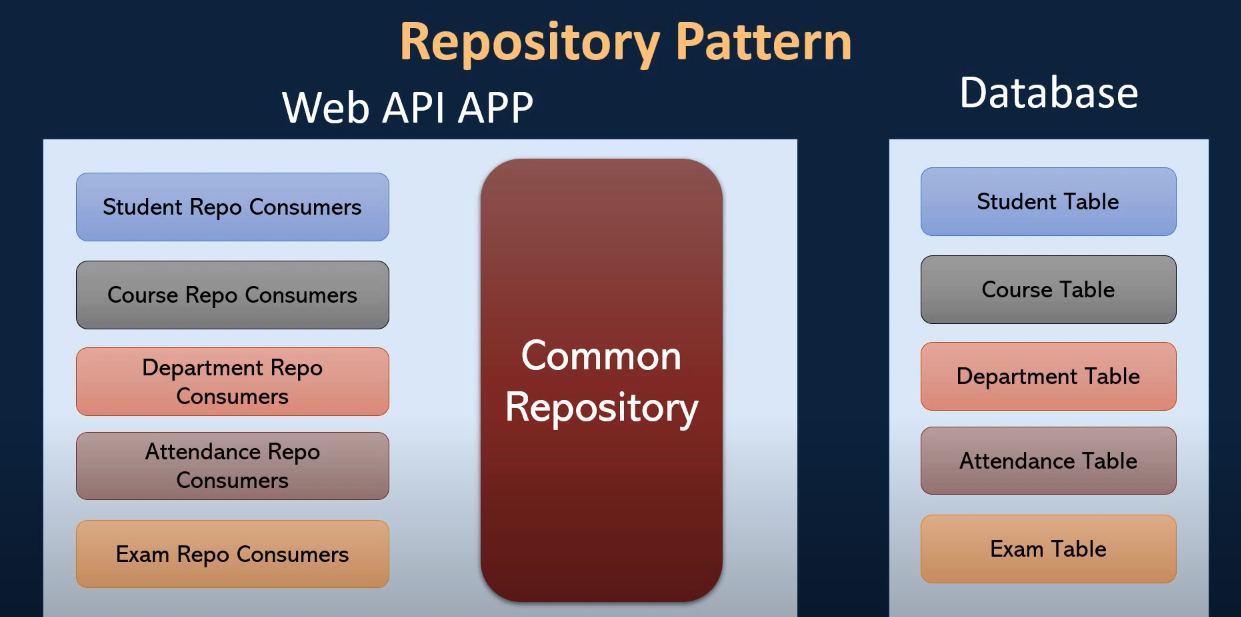
## Here are some of the built-in validation attributes:

* [[ValidateNever]](https://learn.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.mvc.modelbinding.validation.validateneverattribute): Indicates that a property or parameter should be excluded from validation.
* [[CreditCard]](https://learn.microsoft.com/en-us/dotnet/api/system.componentmodel.dataannotations.creditcardattribute): Validates that the property has a credit card format. Requires [jQuery Validation Additional Methods](https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.19.1/additional-methods.js).
* [[Compare]](https://learn.microsoft.com/en-us/dotnet/api/system.componentmodel.dataannotations.compareattribute): Validates that two properties in a model match.
* [[EmailAddress]](https://learn.microsoft.com/en-us/dotnet/api/system.componentmodel.dataannotations.emailaddressattribute): Validates that the property has an email format.
* [[Phone]](https://learn.microsoft.com/en-us/dotnet/api/system.componentmodel.dataannotations.phoneattribute): Validates that the property has a telephone number format.
* [[Range]](https://learn.microsoft.com/en-us/dotnet/api/system.componentmodel.dataannotations.rangeattribute): Validates that the property value falls within a specified range.
* [[RegularExpression]](https://learn.microsoft.com/en-us/dotnet/api/system.componentmodel.dataannotations.regularexpressionattribute): Validates that the property value matches a specified regular expression.
* [[Required]](https://learn.microsoft.com/en-us/dotnet/api/system.componentmodel.dataannotations.requiredattribute): Validates that the field isn't null. See [[Required] attribute](https://learn.microsoft.com/en-us/aspnet/core/mvc/models/validation?view=aspnetcore-8.0#non-nullable-reference-types-and-required-attribute) for details about this attribute's behavior.
* [[StringLength]](https://learn.microsoft.com/en-us/dotnet/api/system.componentmodel.dataannotations.stringlengthattribute): Validates that a string property value doesn't exceed a specified length limit.
* [[Url]](https://learn.microsoft.com/en-us/dotnet/api/system.componentmodel.dataannotations.urlattribute): Validates that the property has a URL format.
* [[Remote]](https://learn.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.mvc.remoteattribute): Validates input on the client by calling an action method on the server. See [[Remote] attribute](https://learn.microsoft.com/en-us/aspnet/core/mvc/models/validation?view=aspnetcore-8.0#remote-attribute) for details about this attribute's behavior.

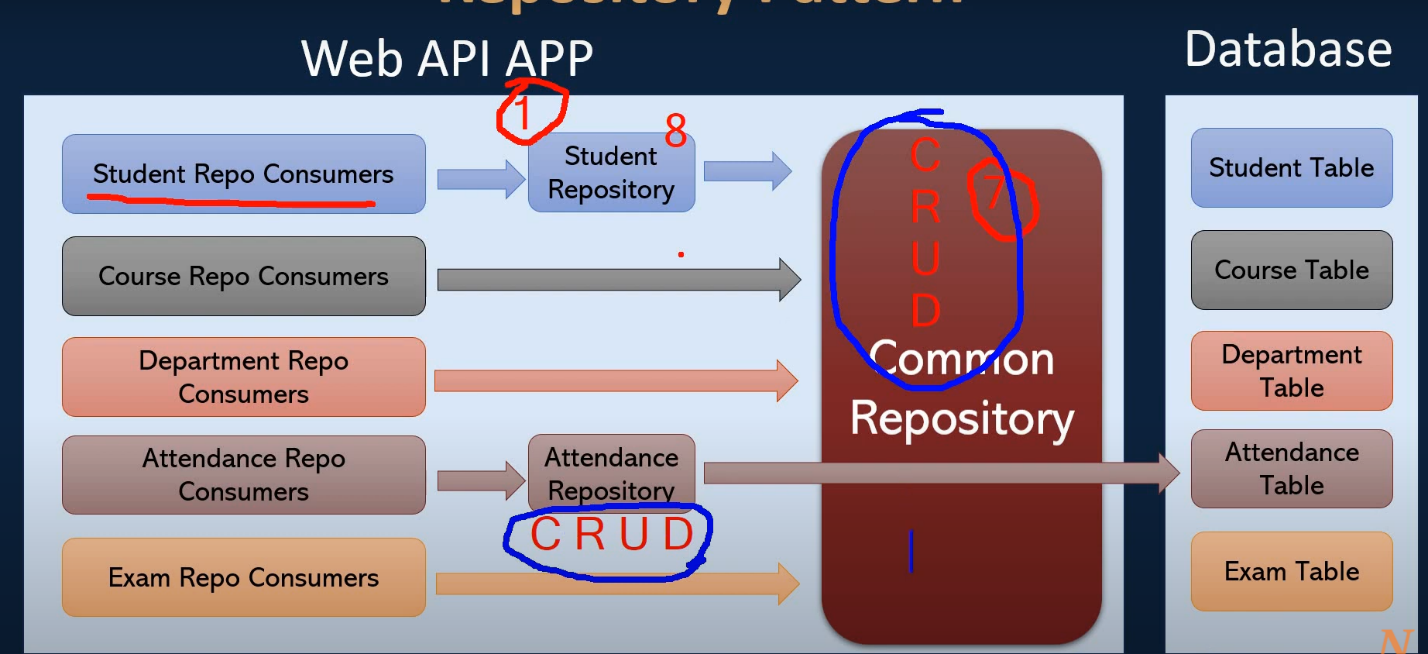
# Repository Pattern

* Repository Pattern is an abstraction of the data Access Layer.
* It hides the details of how exactly the data is saved or retrieving from the underlying table.





Inheriting Common Repository



# EF-Relations

## One-To-Many Relation:

* An important point to note here is that [C# nullable reference types](https://learn.microsoft.com/en-us/dotnet/csharp/tutorials/nullable-reference-types) are being used, so the nullability of the reference navigation is used to determine whether or not the foreign key property is nullable, and therefore whether the relationship is optional or required. If nullable reference types are not being used, then the shadow foreign key property will be nullable by default, making the relationship optional by default. In this case, use IsRequired to force the shadow foreign key property to be non-nullable and make the relationship required.
* As before, this relationship is [discovered by convention](https://learn.microsoft.com/en-us/ef/core/modeling/relationships/conventions). For cases where the navigations, foreign key, or required/optional nature of the relationship are not discovered by convention, these things can be configured explicitly.