# Tutorial for Class number 4

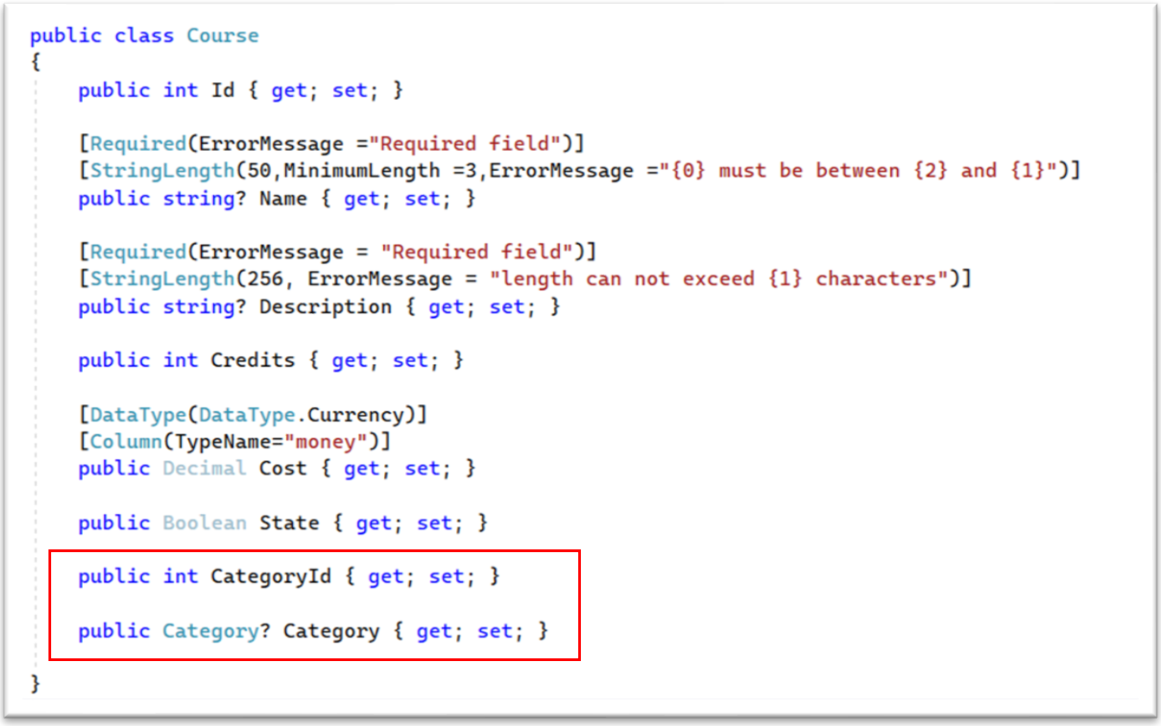
This exercise intends to complement de previous tutorial (Tutorial_Class4) adding a relational class model to the project. The application will continue to allow to perform CRUD operations without having to write SQL. The database system operations will be supported by **Entity Framework Core** and generated by the process **Code First**.

This tutorial also follows the “ASP.NET Core MVC with EF Core - tutorial series”

https://docs.microsoft.com/en-us/aspnet/core/data/ef-mvc/?view=aspnetcore-8.0

### First step

- Open de solution of previous class (Class03)
- Add a new class named **Course** to **Models** folder in project with the next content.



```
public class Course
{
    public int Id { get; set; }

    [Required(ErrorMessage ="Required field")]
    [StringLength(50,MinimumLength =3,ErrorMessage ="{0} must be between {2} and {1}")]
    public string? Name { get; set; }

    [Required(ErrorMessage = "Required field")]
    [StringLength(256, ErrorMessage = "length can not exceed {1} characters")]
    public string? Description { get; set; }

    public int Credits { get; set; }

    [DataType(DataType.Currency)]
    [Column(TypeName="money")]
    public Decimal Cost { get; set; }

    public Boolean State { get; set; }

    public int CategoryId { get; set; }

    public Category? Category { get; set; }
}
```

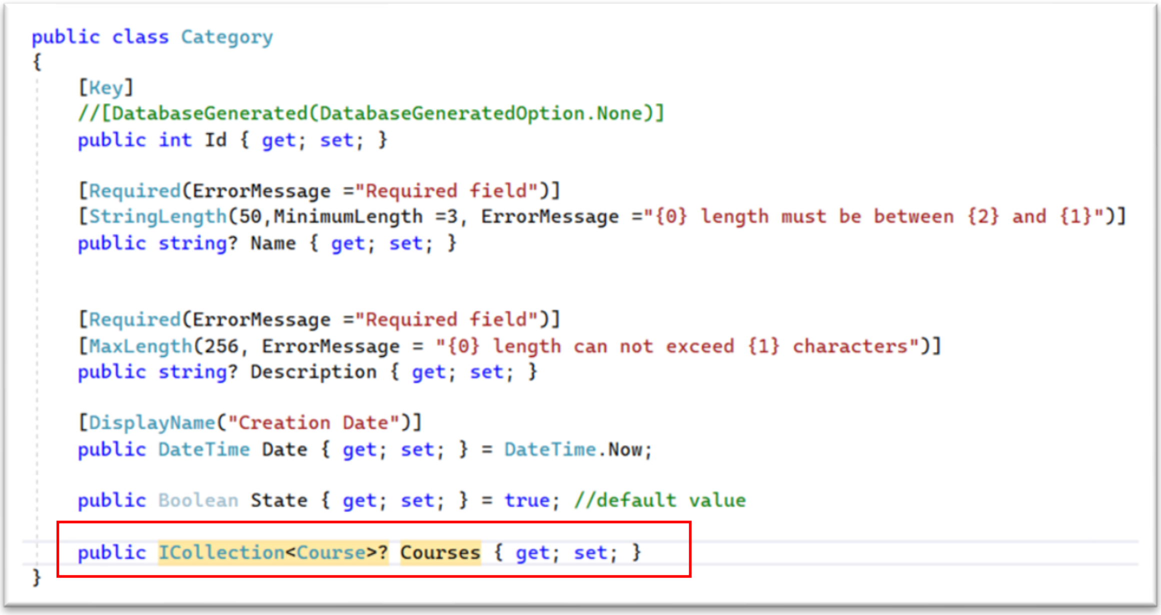
This class intends to represent a **Course** entity in a Database system.

This class has a field that represents the **Category** of the course, which is one of existing items in the Categories table. This is the FOREIGN KEY concept of the relational databases. The foreign key and navigation properties in the Course entity reflects the following relationships:

- A **Category** is assigned to one **Course**, so there's a **CategoryID** foreign key and a **Category** navigation property.

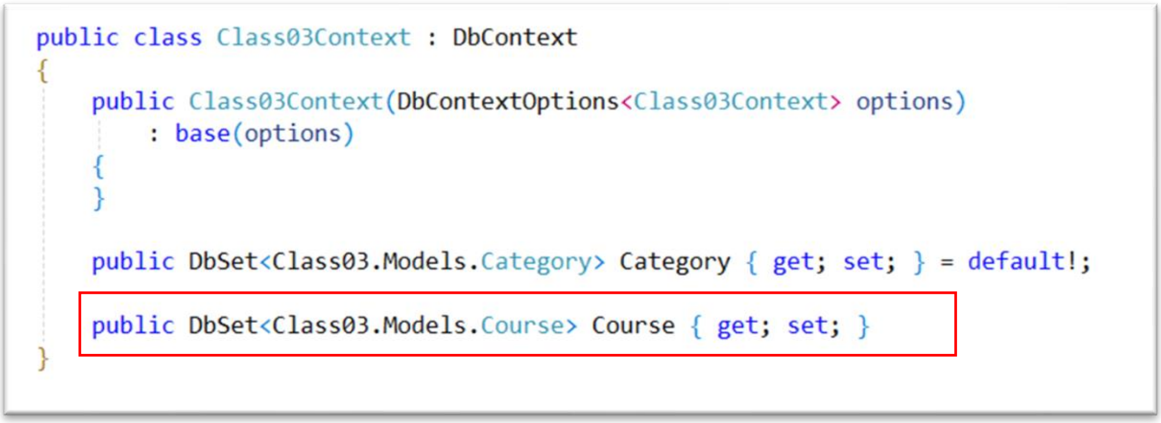
The **Column** attribute is used to change SQL data type mapping so that the column will be defined using the SQL Server money type in the database.

- Now we need to change the Category class to represent that one Category must have multiple Courses. A **Category** may have **many** courses, so there's a **Courses navigation property**



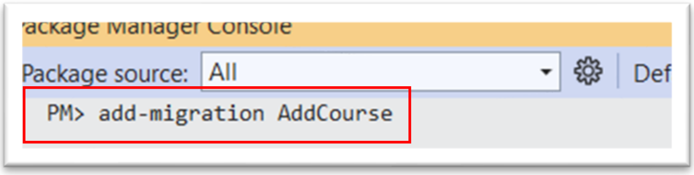
```
public class Category
{
    [Key]
    //[DatabaseGenerated(DatabaseGeneratedOption.None)]
    public int Id { get; set; }

    [Required(ErrorMessage ="Required field")]
    [StringLength(50,MinimumLength =3, ErrorMessage ="{0} length must be between {2} and {1}")]
    public string? Name { get; set; }


    [Required(ErrorMessage ="Required field")]
    [MaxLength(256, ErrorMessage = "{0} length can not exceed {1} characters")]
    public string? Description { get; set; }

    [DisplayName("Creation Date")]
    public DateTime Date { get; set; } = DateTime.Now;

    public Boolean State { get; set; } = true; //default value

    public ICollection<Course>? Courses { get; set; }
}
```

- Now we need to add the class Course to the Data Context:



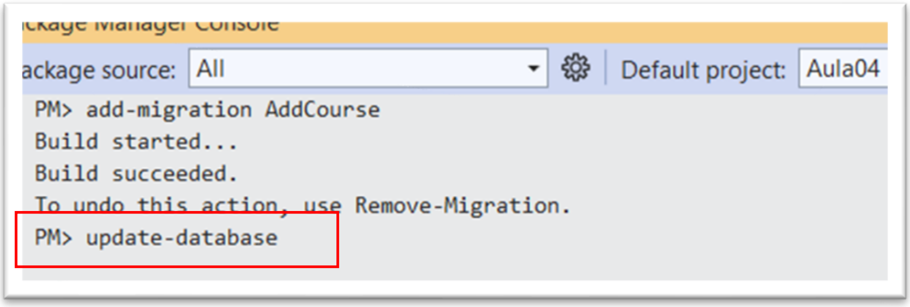
```
public class Class03Context : DbContext
{
    public Class03Context(DbContextOptions<Class03Context> options)
        : base(options)
    {
    }

    public DbSet<Class03.Models.Category> Category { get; set; } = default!;

    public DbSet<Class03.Models.Course> Course { get; set; }
}
```

- To reflect these changes in the Model, we need to create a new Migration:



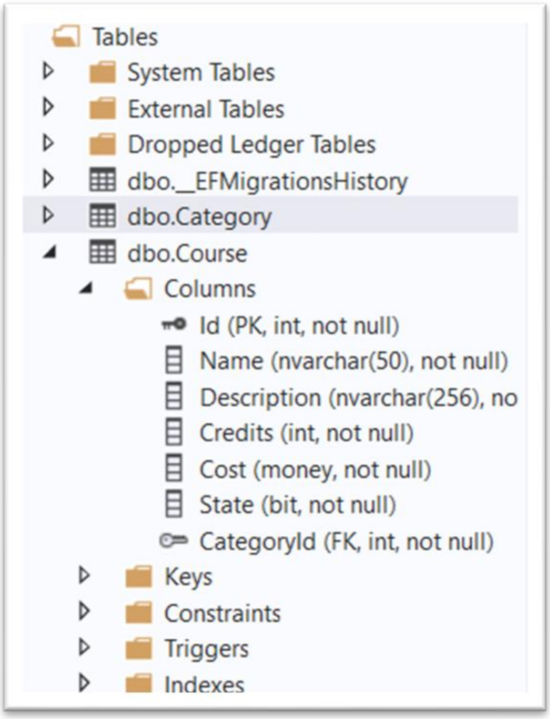
```
PM> add-migration AddCourse
```

- Next we will alter the database by executing the command update-database.



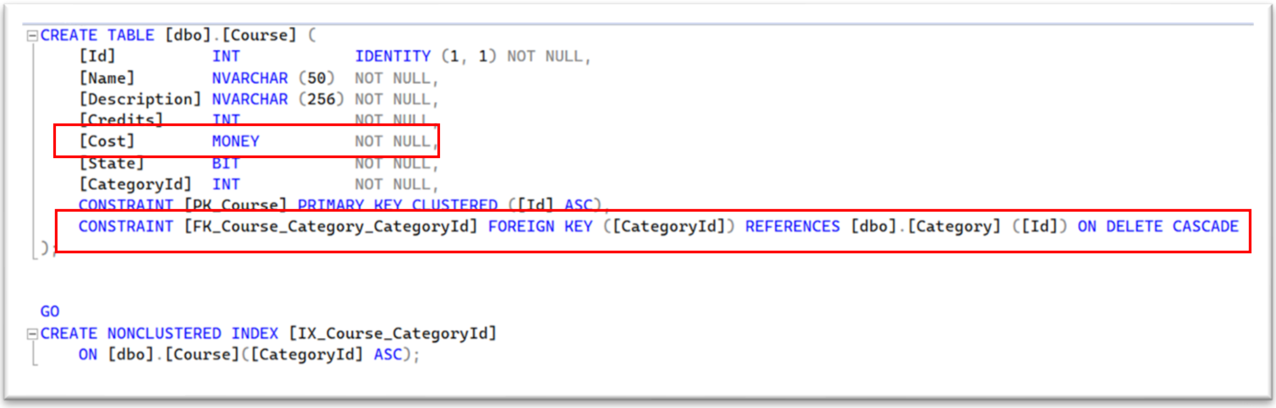
```
PM> add-migration AddCourse
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
PM> update-database
```

Confirm in **SQL Server Object Explorer** the new table created in the database.



```
Tables
  System Tables
  External Tables
  Dropped Ledger Tables
  dbo.__EFMigrationsHistory
  dbo.Category
  dbo.Course
    Columns
      Id (PK, int, not null)
      Name (nvarchar(50), not null)
      Description (nvarchar(256), no
      Credits (int, not null)
      Cost (money, not null)
      State (bit, not null)
      CategoryId (FK, int, not null)
    Keys
    Constraints
    Triggers
    Indexes
```

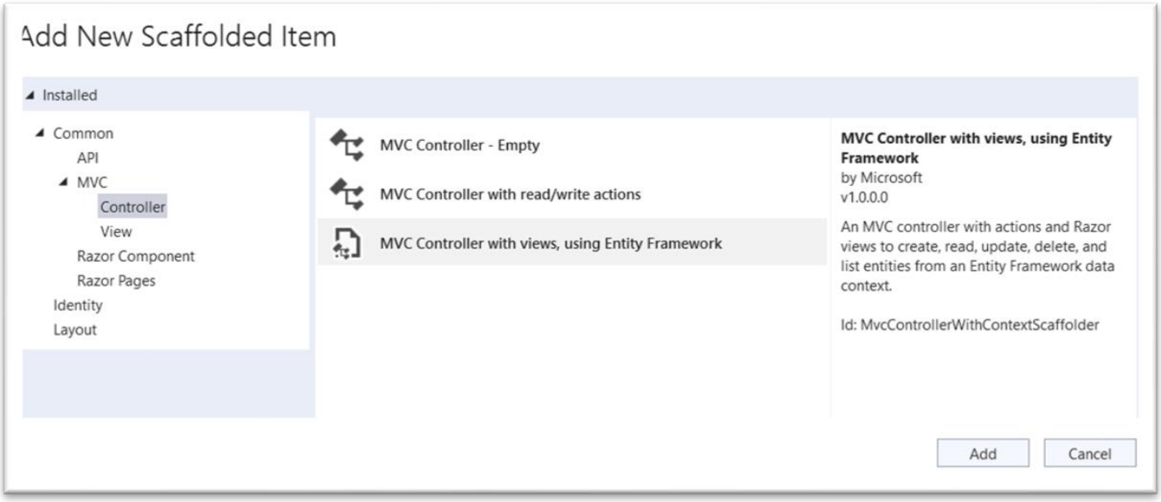
Using the <u>mouse right button</u> over the **dbo.Course** we ca use the <u>View Code</u> option to see the SQL code that generated the table. Here we can see the creation of the **foreign key** in the Category table, and the SQL data type **Money**, reflecting the data annotation **[Column(TypeName = "money")]**



```
CREATE TABLE [dbo].[Course] (
    [Id]          INT            IDENTITY (1, 1) NOT NULL,
    [Name]        NVARCHAR (50)  NOT NULL,
    [Description] NVARCHAR (256) NOT NULL,
    [Credits]     INT            NOT NULL,
    [Cost]        MONEY          NOT NULL,
    [State]       BIT            NOT NULL,
    [CategoryId]  INT            NOT NULL,
    CONSTRAINT [PK_Course] PRIMARY KEY CLUSTERED ([Id] ASC),
    CONSTRAINT [FK_Course_Category_CategoryId] FOREIGN KEY ([CategoryId]) REFERENCES [dbo].[Category] ([Id]) ON DELETE CASCADE
);


GO
CREATE NONCLUSTERED INDEX [IX_Course_CategoryId]
    ON [dbo].[Course]([CategoryId] ASC);
```

## Second Step

- Add a controller with the template "MVC Controller with views, using Entity Framework". This will add to the project one controller class and all views code for CRUD operations over the model data.

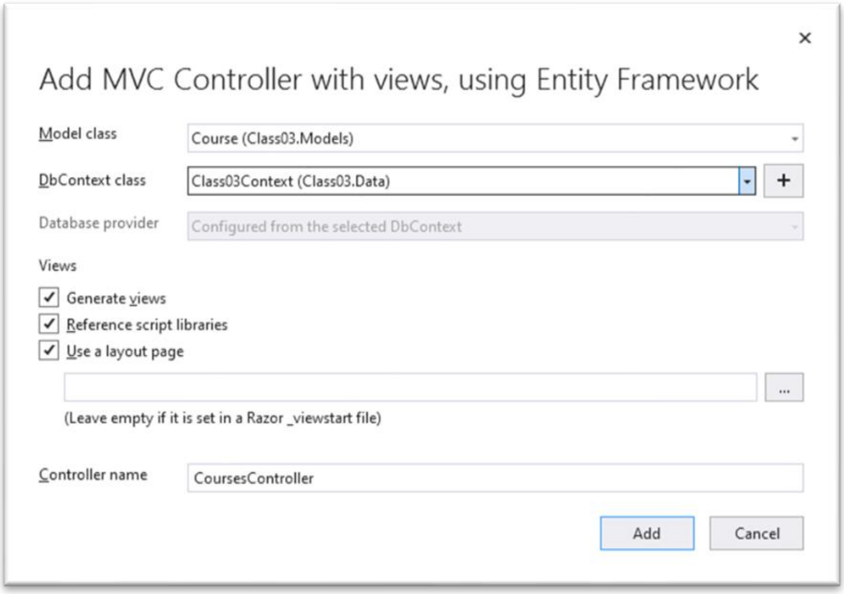


**MVC Controller with views, using Entity Framework**
by Microsoft
v1.0.0.0

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

Id: MvcControllerWithContextScaffolder

- Choose the **Course** class as model class and the existing Data Context class **Aula03Context** (or what else name you gave to it) - in this case you should not create a new one.



Add MVC Controller with views, using Entity Framework

Model class: Course (Class03.Models)

DbContext class: Class03Context (Class03.Data)

Database provider: Configured from the selected DbContext

Views

☑ Generate views

☑ Reference script libraries

☑ Use a layout page

(Leave empty if it is set in a Razor _viewstart file)

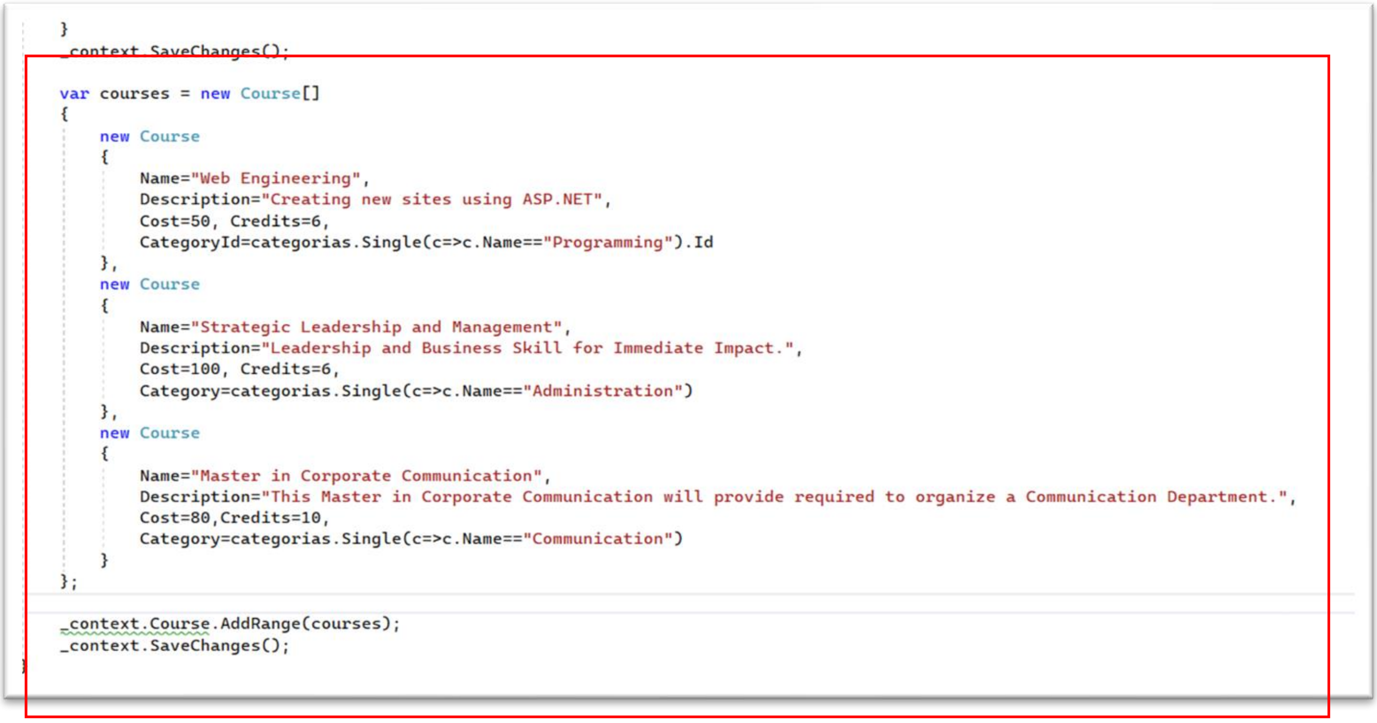
Controller name: CoursesController

- Accept or modify the suggested name for the controller (**CoursesController**).

### Third step

For running the application for the first time it is important to have some data in the database.

- Open the **DbInitializer.cs** file in the **Data** folder, and append the following code, which create new courses in the database.

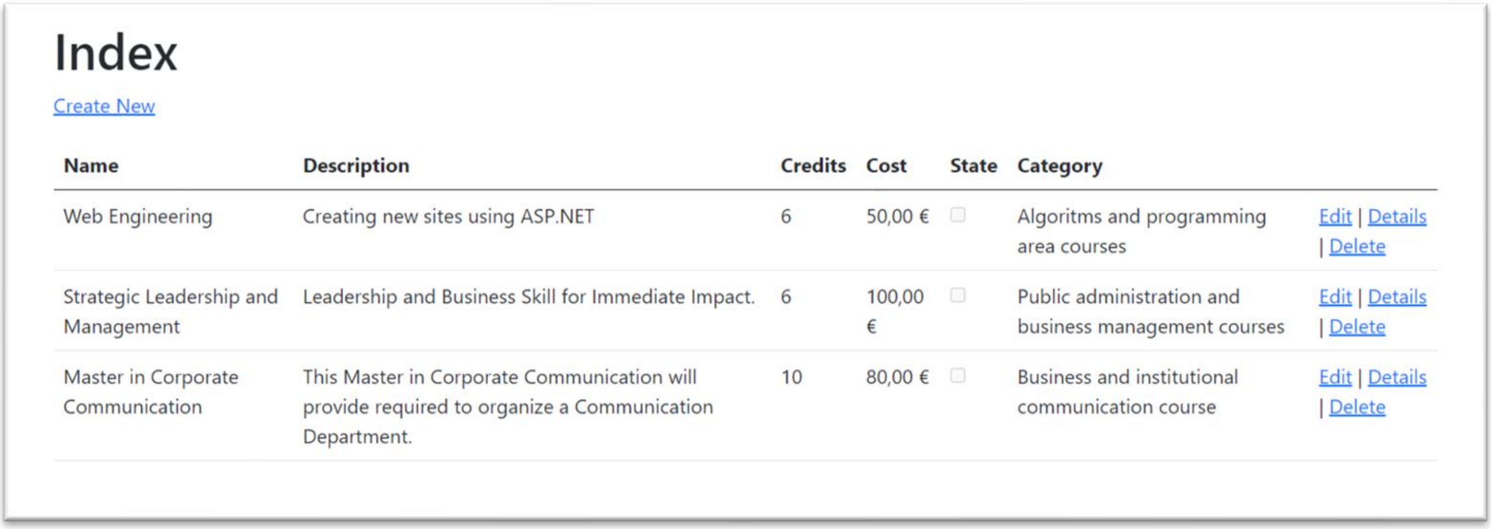


```
    }
    _context.SaveChanges();

    var courses = new Course[]
    {
        new Course
        {
            Name="Web Engineering",
            Description="Creating new sites using ASP.NET",
            Cost=50, Credits=6,
            CategoryId=categorias.Single(c=>c.Name=="Programming").Id
        },
        new Course
        {
            Name="Strategic Leadership and Management",
            Description="Leadership and Business Skill for Immediate Impact.",
            Cost=100, Credits=6,
            Category=categorias.Single(c=>c.Name=="Administration")
        },
        new Course
        {
            Name="Master in Corporate Communication",
            Description="This Master in Corporate Communication will provide required to organize a Communication Department.",
            Cost=80,Credits=10,
            Category=categorias.Single(c=>c.Name=="Communication")
        }
    };

    _context.Course.AddRange(courses);
    _context.SaveChanges();
}
```

To create these courses in the start-up of the application, we need to delete all existing data in the database.

And we are ready to test the application 🧨 🎆

- run on **https://localhost:????/courses/** to see all courses listed…

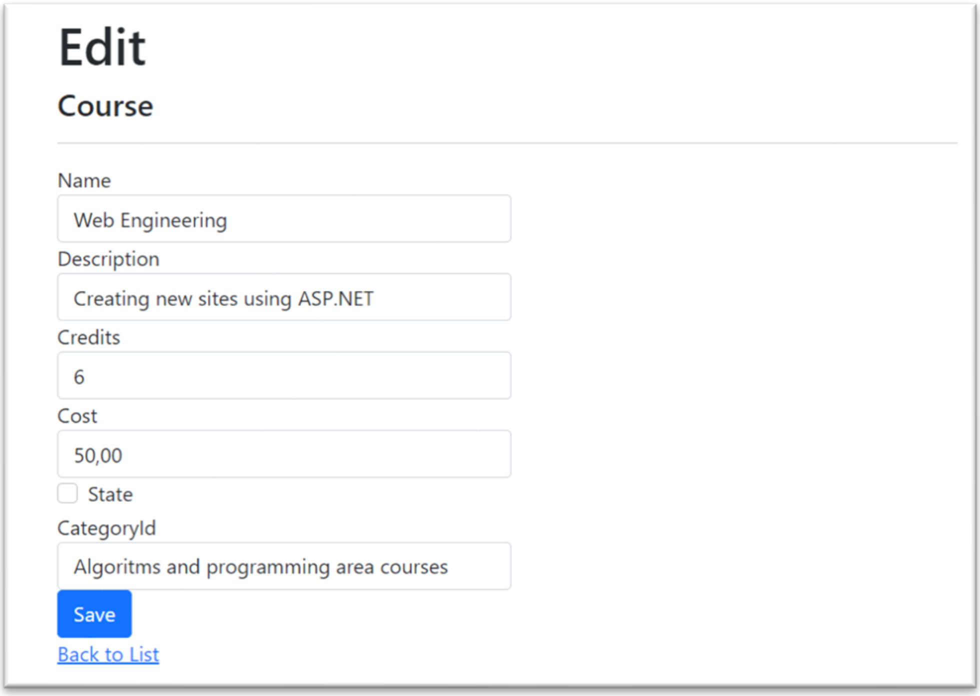


# Index

Create New

| Name | Description | Credits | Cost | State | Category | |
|---|---|---|---|---|---|---|
| Web Engineering | Creating new sites using ASP.NET | 6 | 50,00 € | ☐ | Algoritms and programming area courses | Edit \| Details \| Delete |
| Strategic Leadership and Management | Leadership and Business Skill for Immediate Impact. | 6 | 100,00 € | ☐ | Public administration and business management courses | Edit \| Details \| Delete |
| Master in Corporate Communication | This Master in Corporate Communication will provide required to organize a Communication Department. | 10 | 80,00 € | ☐ | Business and institutional communication course | Edit \| Details \| Delete |

- … or editing one of the elements listed in **https://localhost:????/courses/Edit/1** (by clicking Edit link)



# Edit

## Course

Name: Web Engineering

Description: Creating new sites using ASP.NET

Credits: 6

Cost: 50,00

☐ State

CategoryId: Algoritms and programming area courses

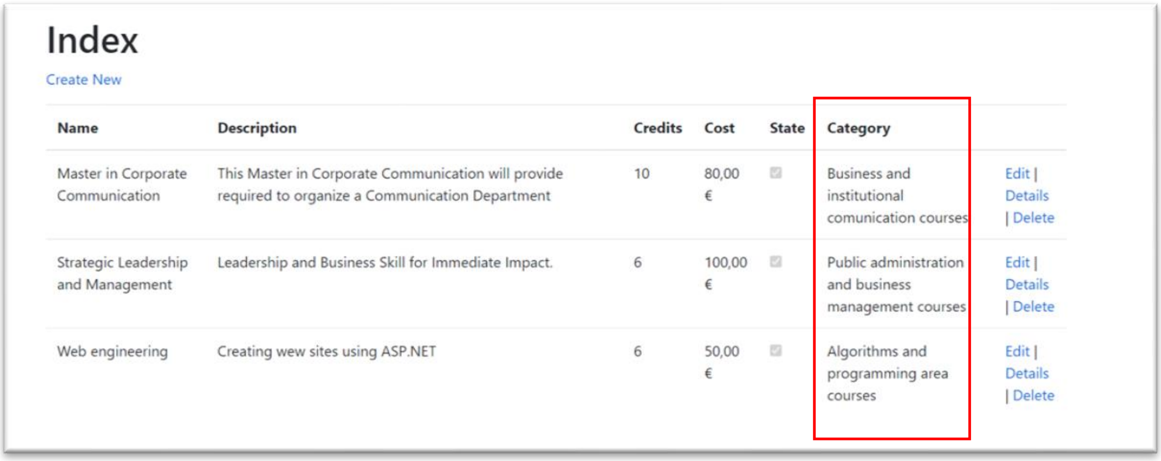
Save

Back to List

You should test all CRUD operations over courses elements. 😆 🙄

**Homework:** 😈

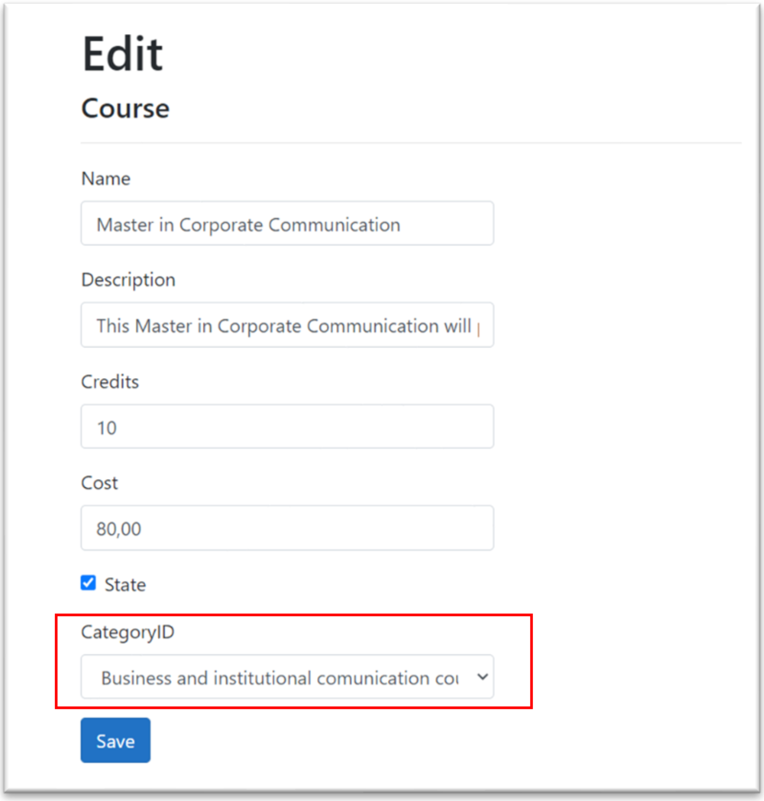
- Change the code where needed to show the category **Name** instead the **Description** in all the views (list, details, edit, create and delete).



# Index

Create New

| Name | Description | Credits | Cost | State | Category | |
|---|---|---|---|---|---|---|
| Master in Corporate Communication | This Master in Corporate Communication will provide required to organize a Communication Department | 10 | 80,00 € | ☑ | Business and institutional comunication courses | Edit \| Details \| Delete |
| Strategic Leadership and Management | Leadership and Business Skill for Immediate Impact. | 6 | 100,00 € | ☑ | Public administration and business management courses | Edit \| Details \| Delete |
| Web engineering | Creating wew sites using ASP.NET | 6 | 50,00 € | ☑ | Algorithms and programming area courses | Edit \| Details \| Delete |



# Edit

## Course

Name

Master in Corporate Communication

Description

This Master in Corporate Communication will

Credits

10

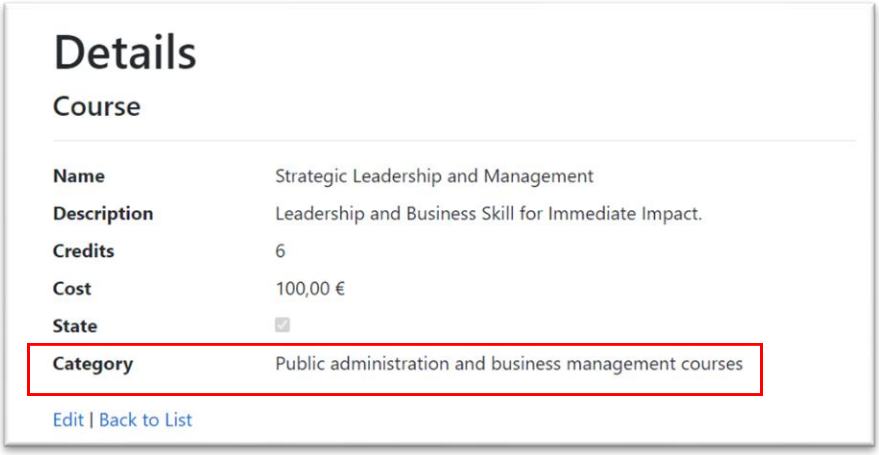
Cost

80,00

☑ State

CategoryID

Business and institutional comunication co

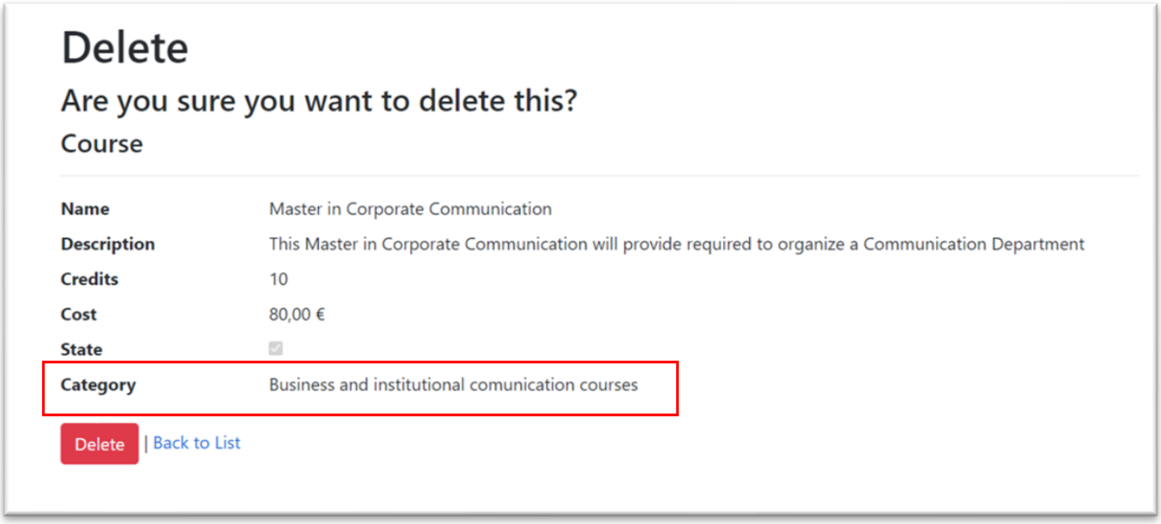


# Details

## Course

| | |
|---|---|
| **Name** | Strategic Leadership and Management |
| **Description** | Leadership and Business Skill for Immediate Impact. |
| **Credits** | 6 |
| **Cost** | 100,00 € |
| **State** | ☑ |
| **Category** | Public administration and business management courses |

Edit | Back to List



# Delete

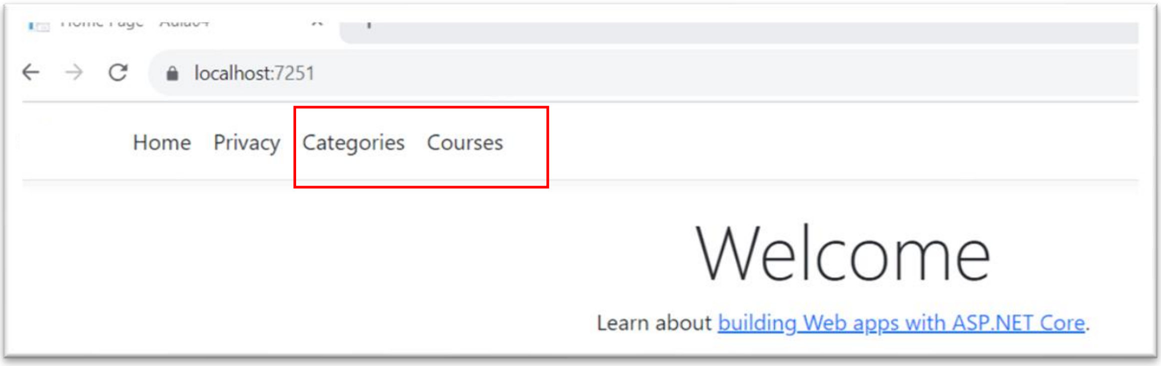
## Are you sure you want to delete this?

### Course

| | |
|---|---|
| **Name** | Master in Corporate Communication |
| **Description** | This Master in Corporate Communication will provide required to organize a Communication Department |
| **Credits** | 10 |
| **Cost** | 80,00 € |
| **State** | ☑ |
| **Category** | Business and institutional comunication courses |

Delete | Back to List

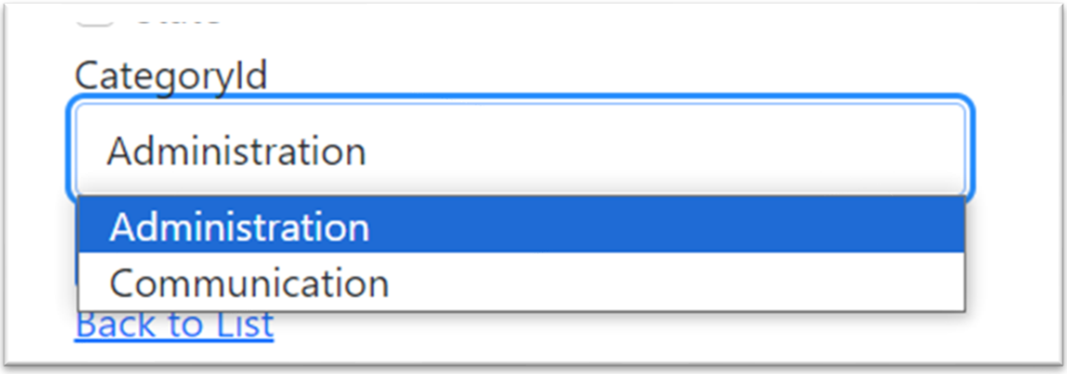
- Create menu items in layout page to manage the categories and courses.



Welcome

Learn about building Web apps with ASP.NET Core.

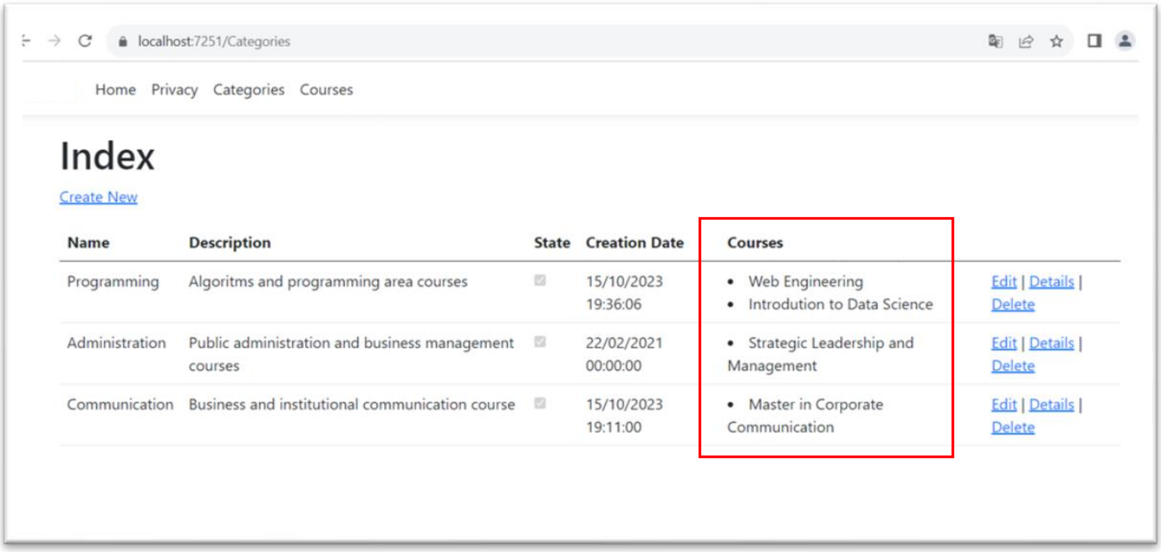
- When creating or editing a new course, only appears to select the categories which State is true.



CategoryId

Administration

- . When list the categories, show all courses with that category



Index

Create New

| Name | Description | State | Creation Date | Courses | |
|---|---|---|---|---|---|
| Programming | Algoritms and programming area courses | ☑ | 15/10/2023 19:36:06 | • Web Engineering<br>• Introdution to Data Science | Edit \| Details \| Delete |
| Administration | Public administration and business management courses | ☑ | 22/02/2021 00:00:00 | • Strategic Leadership and Management | Edit \| Details \| Delete |
| Communication | Business and institutional communication course | ☑ | 15/10/2023 19:11:00 | • Master in Corporate Communication | Edit \| Details \| Delete |