

ASP .Net Core MVC

Série N°2 (suite) : opérations CRUD

I. Ajouter un modele Produit

1. Ajouter un modèle Produit avec les propriétés suivantes :

classe	propriété	accessibilité	type	Annotation
Produit	Id	public	int	clé primaire
	Nom	public	string	Obligatoire Taille max 50 caractères
	Code	public	int	obligatoire
	PrixProduit	public	int	Obligatoire Alias : Prix du produit Valeur entre 1 et 100000
	Description	public	string	-

2. Initialiser la table produit avec quelques données dans la classe ApplicationDbContext

```
public DbSet<Produit> Produits { get; set; }

modelBuilder.Entity<Produit>().HasData(

    new Produit { Id = 1, Nom = "Iphone 14", Description = "iPhone 14 Pro. Avec un
appareil photo principal de 48 MP pour capturer des détails époustouflants. Dynamic
Island et l'écran toujours activé, qui offrent une toute nouvelle expérience sur
iPhone", PrixProduit=10000, Code=123456789 },

    new Produit { Id = 2, Nom = "Imprimante hp deskjet 2710", Description = "Toutes
les fonctions de base, désormais faciles à utiliser. Imprimez, numérisez et copiez
les documents quotidiens, et profitez d'une connexion simple et sans fil",
PrixProduit = 5000, Code = 546456789 }

);
```

3. Dans la console de gestionnaire des package, exécuter la migration et la mise à jour de la base de données
4. Créer un contrôleur Produit avec les opérations CRUD (suivre les tapes du TP2)

II. Ajouter une clé étrangère

5. Ajouter au modèle Produit une clé étrangère vers le modèle Catégorie

```
public int categorieId { get; set; }  
[ForeignKey("categorieId")]  
  
public Category categorie { get; set; }
```

6. Modifier l'initialisation de la table produit dans ApplicationDbContext en renseignant la propriété categorieId avec des valeur de ID de Catégorie

```
modelBuilder.Entity<Produit>().HasData(  
    new Produit { Id = 1, Nom = "Iphone 14", Description = "iPhone 14 Pro. Avec un  
appareil photo principal de 48 MP pour capturer des détails époustouflants. Dynamic  
Island et l'écran toujours activé, qui offrent une toute nouvelle expérience sur  
iPhone", PrixProduit=10000, Code=123456789, categorieId = 1},  
    new Produit { Id = 2, Nom = "Imprimante hp deskjet 2710", Description = "Toutes  
les fonctions de base, désormais faciles à utiliser. Imprimez, numérisez et copiez  
les documents quotidiens, et profitez d'une connexion simple et sans fil",  
PrixProduit = 5000, Code = 546456789 , categorieId = 3 }  
);
```

7. Dans la console de gestionnaire des package, exécuter la migration et la mise à jour de la base de données

III. Création d'un ViewModel

8. Dans le dossier Models, ajouter un sous dossier ViewModels
9. Ajouter dans ce dossier une classe ProduitVM avec le code suivant

```
public class ProduitVM  
{  
    public Produit produit { get; set; }  
    public IEnumerable<SelectListItem> categorieListe { get; set; }  
}
```

10. Référencer le package using Microsoft.AspNetCore.Mvc.Rendering;

11. Dans le fichier _ViewImports.cshtml, ajouter la déclaration

```
@using EcomApp.Models.ViewModels
```

12. Dans la vue Create de Produit :

- Coupler la vue avec le model @model ProduitVM
- Ajouter une balise select pour afficher les categories

```
<div class="mb-3 row">
  <label asp-for="produit.categorieId"></label>
  <select asp-for="produit.categorieId" asp-items="@Model.categorieListe"
class="form-select">
    <option disabled selected>--choisir une catégorie--</option>
  </select>
</div>
```

13. Modifier l'action Create du controleur Produit

```
public IActionResult Create()
{
    ProduitVM produitVM = new()
    {
        categorieListe = _db.Categories.ToList().Select(u => new SelectListItem
        { Text = u.Nom,
          Value = u.Id.ToString()
        }),
        produit = new Produit()
    };
    return View(produitVM);
}
```

```
[HttpPost]
public IActionResult Create(ProduitVM obj)
{
    if(ModelState.IsValid)
    {
        _db.Produits.Add(obj.produit);
        _db.SaveChanges();
        return RedirectToAction("Index");
    }
    else return View();
}
```

14. Ajouter l'annotation [ValidateNever] sur les propriété categorieListe de ProduitVM et categorie de Produit

IV. Ajouter une propriété image

15. Ajouter une propriété au modèle Produit pour stocker l'url d'image

```
[ValidateNever]
```

```
public string imageURL { get; set; }
```

16. Modifier l'initialisation de la table produit dans ApplicationDbContext en renseignant la propriété imageURL à une chaîne vide

17. Dans la vue Create, ajouter au formulaire 'file upload input' pour stocker la propriété imageURL

```
<div class="mb-3 row">
  <label asp-for="produit.imageURL"></label>
  <input type="file" name="file" class="form-control" />
</div>
```

Ajouter dans la balise création du formulaire

```
<form method="post" enctype="multipart/form-data" >
```

18. Dans le dossier 'wwwroot', ajouter un sous dossier 'Images'

19. Dans le contrôleur Produit, ajouter une propriété IWebHostEnvironment

```
public class ProduitController : Controller
{
    private readonly ApplicationDbContext _db;
    private readonly IWebHostEnvironment _webHostEnvironment;

    public ProduitController(ApplicationDbContext db, IWebHostEnvironment webHost)
    {
        _db = db;
        _webHostEnvironment = webHost;
    }
}
```

20. Modifier l'action Create pour enregistrer l'image du produit

```
[HttpPost]
public IActionResult Create(ProduitVM obj, IFormFile file)
{
    string wwwRootPath = _webHostEnvironment.WebRootPath;
    if (file != null)
    {
        string
        fileName = Guid.NewGuid().ToString() + Path.GetExtension(file.FileName); // donner un nom
        aléatoire
        string filePath = Path.Combine(wwwRootPath, @"Images");
        using (var fileStream = new FileStream(Path.Combine(filePath, fileName),
        FileMode.Create))
        {
            file.CopyTo(fileStream);
        }
    }
}
```

```
    }
    obj.produit.imageURL = @"Images\" + fileName;
}

if(ModelState.IsValid)
{
    _db.Produits.Add(obj.produit);
    _db.SaveChanges();
    return RedirectToAction("Index");
}
else return View();
}
```

21. Dans la vue Index, ajouter une balise pour afficher l'image

```
<td>
    
</td>
```