

## TP1 : Initiation à ASP.NET Core MVC

On vous demande de créer une application web ASP.NET Core MVC afin de gérer les opérations de base sur les employés d'une entreprise.

On vous donne le code de la classe Employee :

```
public class Employee
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Departement { get; set; }
    public int Salary { get; set; }
}
```

- Ajouter la classe Employee dans le dossier Model.
- Sous le dossier Model créer un nouveau dossier Repositories.
- Dans ce nouveau dossier créer l'interface suivante :

```
public interface IRepository<T>
{
    IList<T> GetAll();
    T FindByID(int id);
    void Add(T entity);
    void Update(int id, T entity);
    void Delete(int id);
    List<T> Search(string term);
}
```

- Créer aussi dans le dossier Repositories une classe qui implémente l'interface nommée EmployeeRepository et donner le code des méthodes en utilisant LINQ.

```
public class EmployeeRepository : IRepository<Employee>
{
    List<Employee> lemployees;

    public EmployeeRepository()
    {
        lemployees = new List<Employee>()
        {
            new Employee {Id=1,Name="Sofien ben ali", Departement= "comptabilité",Salary=1000},
            new Employee {Id=2,Name="Mourad triki", Departement= "RH",Salary=1500},
            new Employee {Id=3,Name="ali ben mohamed", Departement= "informatique",Salary=1700},
            new Employee {Id=4,Name="tarak aribi", Departement= "informatique",Salary=1100}
        };
    }

    public void Add(Employee e)
    {
        //ajouter un employé dans la liste
    }

    public void Delete(int id)
    {
        //Supprime l'employé ayant l'Id passé en paramètre
    }
}
```

```

    public Employee FindByID(int id)
    {
        //retourne l'employé ayant cet id
    }

    public List<Employee> GetAll()
    {
        // Retourne la liste des employés ;
    }

    public List<Employee> Search(string term)
    {
        //Recherche des employés dont leur noms contiennent la chaîne term
    }

    public void Update(int id, Employee newemployee)
    {
        //Modifier un employé
    }
}

```

- Dans le dossier Controllers, Créer un nouveau contrôleur (avec read/write actions) permettant de gérer les opérations sur les employés.
- Donner le code des méthodes d'actions de EmployeeController.
- Générer les vues nécessaires à chaque méthode d'action du contrôleur avec la technique de scaffolding de MVC.
- N'oubliez pas d'inscrire la dépendance de la classe EmployeeRepository dans la méthode ConfigureServices de la classe Startup.cs :

```

public void ConfigureServices(IServiceCollection services)
{
    services.AddControllersWithViews();
    services.AddSingleton<IRepository<Employee>, EmployeeRepository>();
}

public class EmployeeController : Controller
{
    readonly IRepository<Employee> employeeRepository;

    //injection de dépendance
    public EmployeeController(IRepository<Employee> empRepository)
    {
        employeeRepository = empRepository;
    }

    public ActionResult Index()
    {
    }

    public ActionResult Details(int id)
    {
    }

    public ActionResult Create()
    {
    }

    [HttpPost]
    public ActionResult Create(Employee e)
    {
    }
}

```

```
public ActionResult Edit(int id)
{
}

[HttpPost]
public ActionResult Edit(int id, Employee newemployee)
{
}

public ActionResult Delete(int id)
{
}

[HttpPost]
public ActionResult Delete(int id, Employee e)
{
}

public ActionResult Search(string term)
{
}
}
```

Gestion des Employés Home Liste Employés

## Liste des Employés

[Ajout Employé](#)

Search

Id	Name	Departement	Salary	
1	Sofien ben ali	comptabilité	1000	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
2	Mourad triki	RH	1500	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
3	ali ben mohamed	informatique	1700	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
4	tarak aribi	informatique	1100	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

[Afficher tous les employés](#)

© 2020 New software