**TP Airport Management**

**Partie 8: ASP.NET MVC**

**I  Création du projet web**

1. Ajouter le projet **AM.UI.WEB** de type Application web **ASP.NET MVC (6.0)**.
2. Dans le projet Web installer les composants suivants :
   * **EntityFramework.Design**
   * **VisualStudio.Web.CodeGeneration.Design**
3. Définir le projet **AM.UI.WEB** comme projet de démarrage.
4. Lancer l’application et analyser le résultat.

**III Création d’un contrôleur**

1. Dans le projet **Web**, référencer les projets **Core** et **Infrastructure**.
2. Au niveau du dossier **Controllers**, ajouter un contrôleur nommé **FlightController** contenant par défaut les actions de lecture/écriture.
3. Dans la classe **FlightController**, injecter le service **IServiceFlight**.

IServiceFlight sf;

public FlightController(IServiceFlight sf)

{

this.sf = sf;

}

1. Dans la classe **program.cs**, injecter les dépendances en utilisant le conteneur d’injection .Net.

// Injection de dépendance

builder.Services.AddDbContext<DbContext, AMContext>();

builder.Services.AddScoped<IUnitOfWork, UnitOfWork>();

builder.Services.AddScoped<IServiceFlight, ServiceFlight>();

builder.Services.AddScoped<IServicePlane, ServicePlane>();

builder.Services.AddSingleton<Type>(t => typeof(GenericRepository<>));

**IV Création d’une vue Index**

L’objectif est de retourner une vue affichant la liste des vols.

1. Dans **FlightController**, modifier l’action **Index** afin de récupérer la liste de vols et la passer en paramètre à la vue.

// GET: FlightController

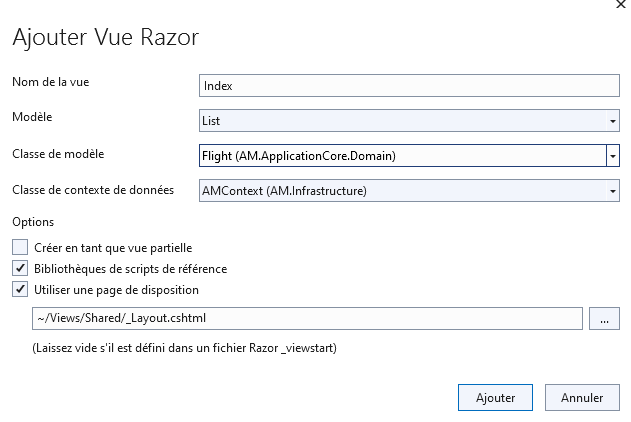
public ActionResult Index()

{

return View(sf.GetMany());

}

1. Créer la vue **Index** : faites un clic droit sur le nom de l’action **Index,** cliquez sur **AddView** et remplir la fenêtre.



1. Exécuter l’application et afficher la vue **Index**.

**V Ajout d’une barre de recherche**

On se propose d’ajouter une barre de recherche qui permet de filtrer les vols par date de départ.

1. Insérer le code html nécessaire dans la vue **Index**.

<**form** **asp-action**="index">

<fieldset>

<legend> Recherche</legend>

Saisir une date de départ : <input type="date" name="dateDepart" />

<input type="submit" value="Serach" />

</fieldset>

</**form**>

1. Modifier l’action **Index** afin de filtrer les vols affichés en fonction du filtre saisi.

// GET: FlightController

public ActionResult Index(DateTime? dateDepart)

{

if (dateDepart == null)

return View(sf.GetMany());

else

return View(sf.GetMany(f => f.FlightDate.Date.Equals(dateDepart)));

}

1. Lancer l’application et tester la recherche

**VII Utilisation d’un fichier layout**

1. Modifier le fichier **\_Layout.cshtml** afin d’ajouter un lien pour l’action **Index** et tester.

<**a** class="nav-link text-dark" **asp-area**="" **asp-controller**="Flight" **asp-action**="Index">Flight</**a**>

**VI Création de la vue Create**

L’objectif est de créer un formulaire d’ajout d’un avion.

1. Ajouter un contrôleur nommé **PlaneController**.
2. Faire les injections nécessaires.

IServicePlane sp;

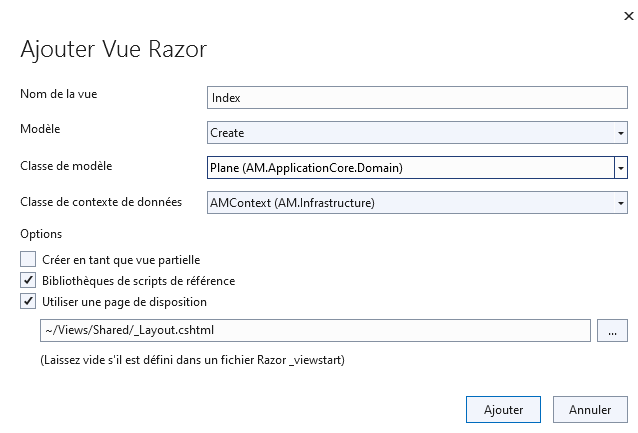
public PlaneController(IServicePlane sp)

{

this.sp = sp;

}

1. Faire un clic droit sur le nom de l’action **Create,** choisir **AddView** et remplir la fenêtre.



1. Adapter le code de la vue pour sélectionner le type de l’avion.

<**select** **asp-for**="PlaneType" class="form-control" **asp-items**="Html.GetEnumSelectList<AM.ApplicationCore.Domain.PlaneType>()"></**select**>

1. Modifier l’action **Create POST** afin d’insérer un avion dans la BDD.

// POST: PlaneController/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(Plane collection)

{ try

{

sp.Add(collection);

sp.Commit();

return RedirectToAction(nameof(Index));

}

catch

{

return View();

}

}

L’objectif est de créer un formulaire d’ajout d’un vol.

1. Ajouter et configurer la vue **Create**.
2. Sélection de l’avion affecté au vol:
   1. Afficher tous les détails des avions dans la liste déroulante. Pour cela, on va ajouter la propriété Information Non Mappée suivante dans la classe Plane :

[NotMapped]

public string Information { get { return PlaneId + " " + ManufactureDate + " " + Capacity; } }

* 1. Dans la classe FlightContoller, ajouter le code nécessaire qui permet de récupérer la liste des avions.

// GET: FlightController/Create

public ActionResult Create()

{

ViewBag.PlaneFk =

new SelectList(sp.GetMany(),"PlaneId", "Information");

return View();

}

* 1. Adapter la vue **Create** pour afficher la liste des avions.

<**select** **asp-for**="PlaneFk" class ="form-control" **asp-items**="ViewBag.PlaneFk"></**select**>

1. Modifier l’action **Create POST** afin d’insérer le vol dans la BDD.

// POST: FlightController/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(Flight collection)

{ try

{ sf.Add(collection);

sf.Commit();

return RedirectToAction(nameof(Index));

}

catch

{ return View();

} }