Laboratoire: révision MVC

# Objectif(s)

**GitHub**

* Clonage
* Création de branches
* Commit/Push/Merge

**Révision 2W5**

* 1 seul projet MVC (1 couche)
* CRUD à partir de listes *ViewBag*
* FCT Create Zombie
* FCT Create Zombie Type
* Validation côté-client et côté-serveur
* Annotations des modèles
* Erreur *span*

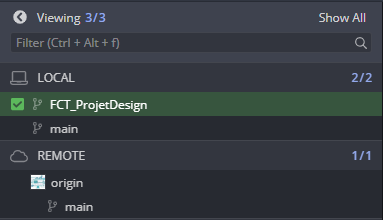
# Repository GitHub

## Clonage du repository

1. Dans Gitkraken, clonez le repository
   * Soit via URL
   * Soit via GitHub.com
2. Ouvrez le repository (OK)

## Création d’une branche pour la fonctionnalité

1. Créez une branche appelée **FCT\_ProjetDesign**
2. Assurez-vous d'être dans la branche que vous avez créée dans LOCAL

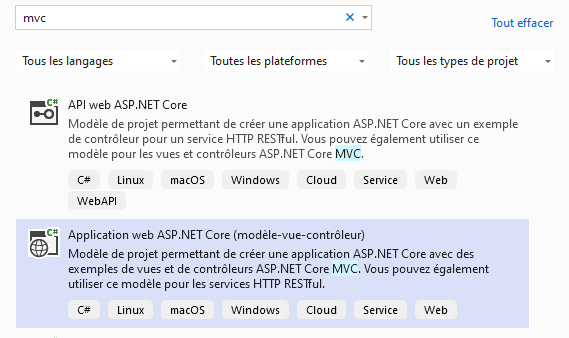


NOTE: si la branche de la séance n'apparaît que dans la section **REMOTE,** c’est qu’elle n’a pas été téléchargée: double-cliquez sur celle-ci pour qu’elle se copie dans LOCAL (voir image précédente) Vous pourrez alors sélectionner (checkout) la branche locale FCT\_ProjetDesign

# Solution/projet MVC ZombieParty

## Création du projet et de la solution ZombieParty

1. Dans Visual Studio, créez un nouveau projet **Application Web Asp Net Core (Modèle-vue-contrôleur)**



Étape 1:

* + Nom du projet: **ZombieParty**
  + Emplacement**: Dossier où vous avez cloné le repo.**
  + NE PAS mettre la solution et le projet dans le même répertoire

Étape 2:

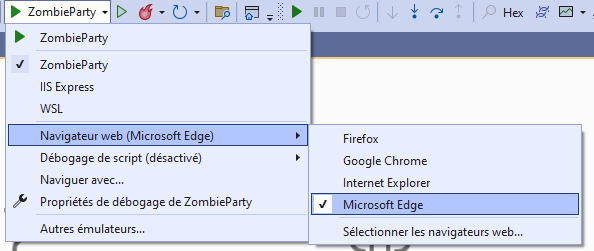
* NET 6.0

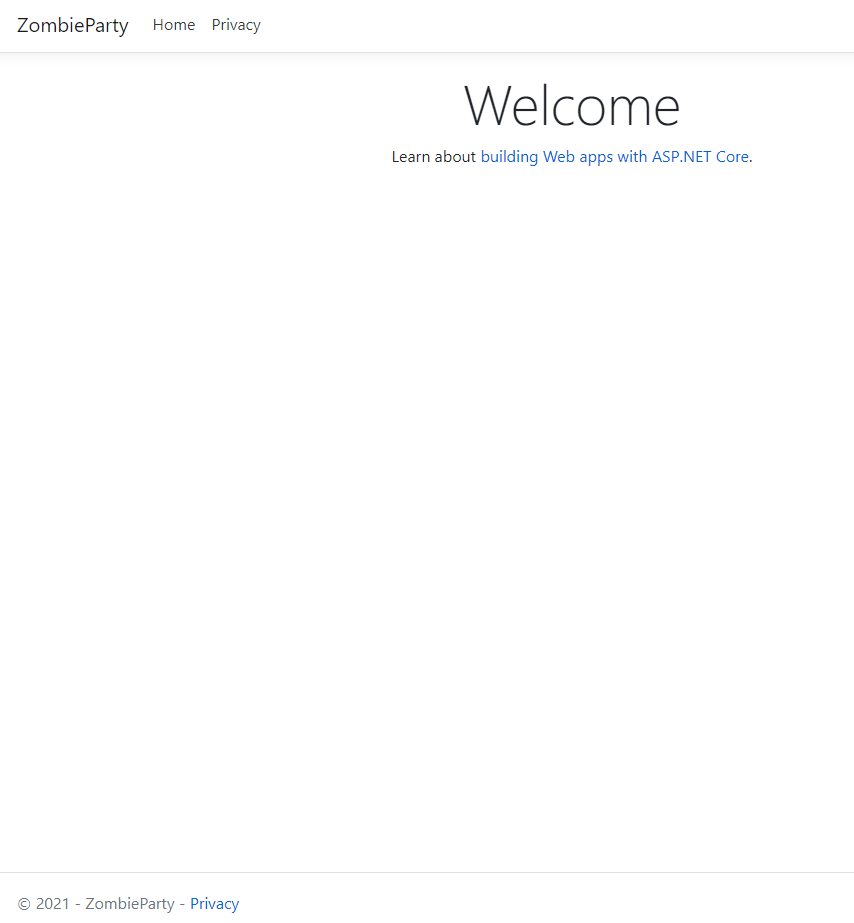
1. Attendez le chargement…soyez patient 😉
2. Vérifiez que vous êtes bien dans la bonne branche (en bas à droite de visual studio)



NOTE: Si vous voyez *MAIN*, vous n'êtes pas dans la bonne branche. Changez de branche et recréez le projet. (on corrigera le *Main* plus tard avec les *Merges*). Parfois, c’est long avant que les informations apparaissent, si c’est le cas, allez dans *Affichage-Modification git.* Vous pouvez fermer cette fenêtre toute suite après, les informations devraient être visibles.

1. Exécutez le projet: Après avoir accepté le certificat etc,, la présentation de la page Home/Index par défaut devrait apparaître. Si vous avez des problèmes de sécurité sur Localhost, changez de navigateur pour Edge.





1. Arrêtez l'exécution

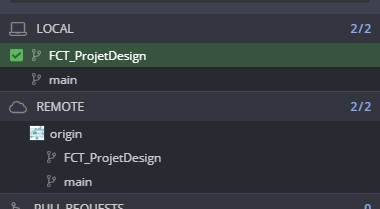
## Commit/PUSH des changements dans le code

1. Dans Gitkraken, assurez-vous d'être dans le bon dossier/repo et sur la bonne branche (**FCT\_ProjetDesign)**
2. Cliquez sur **View changes**
3. Validez la création des 76 fichiers. Cliquez sur **Stage all changes**
4. Commentez en respectant les standards proposés (vous serez noté là-dessus):
   * **Summary**: INIT Solution Projet Web
   * **Description**: Création initiale du projet MVC (1 seul projet)
5. Cliquez sur **Commit**

NOTE: lorsqu’on ne fait qu’un commit les changements sont enregistrés dans le repo LOCAL SEULEMENT. Il faut faire un push pour que ce soit accessible de Github.

1. "Poussez" (**Push)** les modifications sur le repo en ligne (**Remote)** cliquez sur 

Les changements sont maintenant sur le serveur (REMOTE)



NOTE: Vous pourriez faire plusieurs **commits,** et faire un **Push** avec tous les commits. Conseil: ne pas attendre trop longtemps avant de pousser votre code.

## Modification de la présentation du projet : Bootswatch et cie

### Ajouter Les images de présentation

1. Le Repository de départ contient un dossier dans lequel vous retrouvez les fichiers supplémentaires nécessaires au laboratoire.

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

Dans votre dossier de Solution/Projet **ZombieParty**:

1. Ouvrez le dossier *S01.1\_Lab2\_FichiersComplementaires*
2. Copiez le dossier (**images**) et son contenu dans votre dossier **wwwroot**. Il contient les images générales nécessaire au design du site :
   * **cimetierePale.png**
   * **Tile.png**
   * **Degrade.png**
   * **Fond.png**
3. Remplacez le fichier **wwwroot/favicon.ico** par celui contenu dans *S01.1\_Lab2\_FichiersComplementaires*

### Modification du layout et application d'un style bootswatch

**Style**

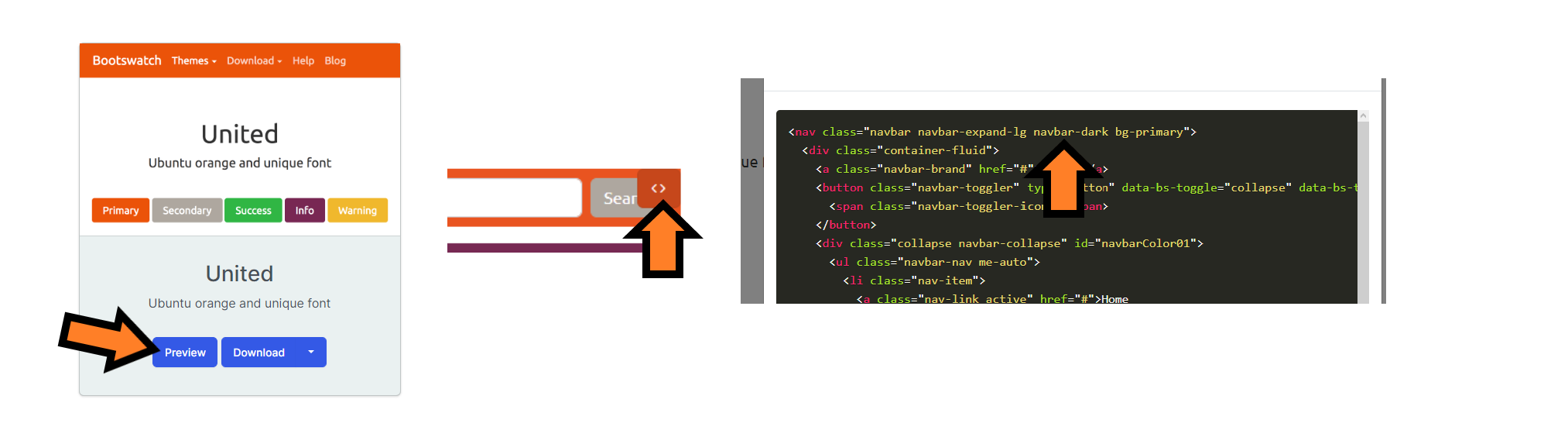
1. Dans votre dossier de Solution/Projet **ZombieParty**:
2. Remplacez le fichier fichier **wwwroot/css/site.css** par celui contenu dans *S01\_Lab2\_FichiersComplementaires*
3. Testez

Une image contenant texte, nuage, capture d’écran, nature

Description générée automatiquement

**Bootstrap/bootswatch**

1. Allez sur le site : [Bootswatch.com](https://bootswatch.com/)
2. Choisissez un thème et télécharger les CSS, **bootstrap.css** et **bootstrap.min.css** (Celui de la solution est **United**, couleurs Halloween)
3. Dans le dossier du projet, remplacez le fichier **wwwroot/lib/bootstrap/dist/css/bootstrap.css** et **bootstrap.min.css** par ceux téléchargés
4. Testez: Ce devrait être NON concluant pour les couleurs
5. Sur le site de Bootswatch, allez voir Preview du modèle choisi afin de découvrir pourquoi...



**Modifier \_Layout,** afin d’utiliser les bonnes classes CSS

1. Dans **Views/Shared/\_Layout,** remplacez les deux classes de la **nav bar:**
   * navbar-light
   * bg-white

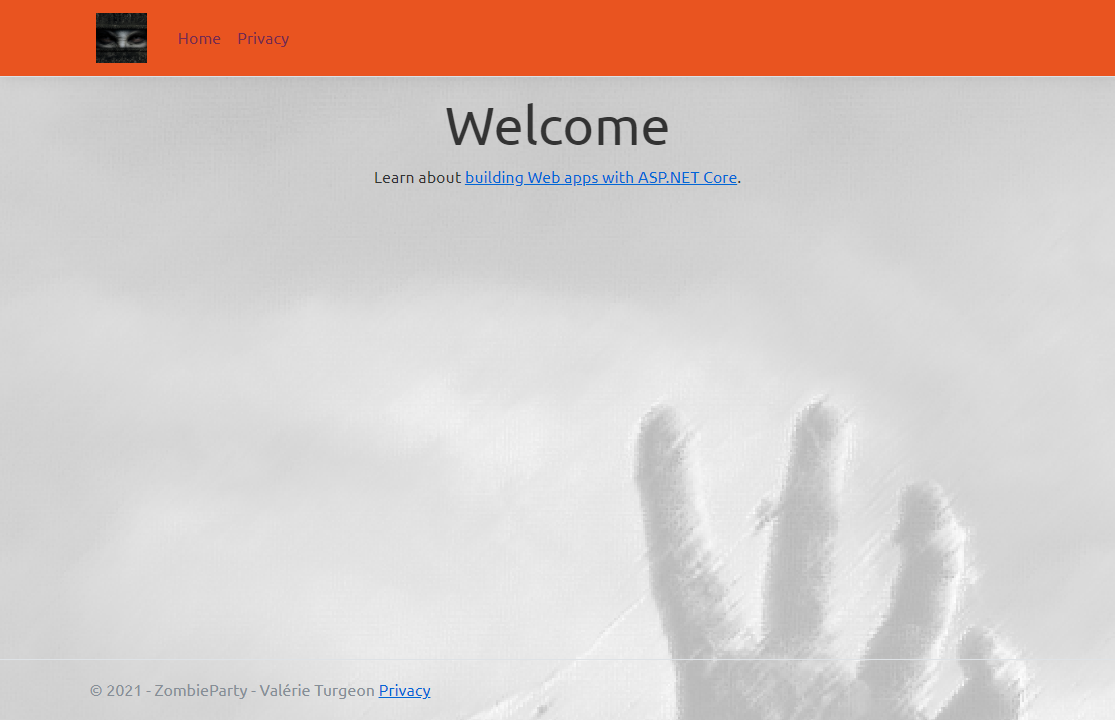
**Par:**

* + navbar-dark
  + bg-primary

1. Testez et adaptez (il sera peut-être nécessaire de vider la cache du navigateur (ctrl+F5)
2. Remplacez le nom de l’application (ZombieParty)dans le coin gauche de la **nav bar** par l'image **images/imagesLayout/tile.png** (balise img)

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index"><img src="~/images/imagesLayout/tile.png" /></a>

1. Testez
2. Ajoutez votre nom dans le pied de page (**Footer),** après le nom de l'application



## Commentaires et validation (Commit) des changements dans le code

1. Dans Gitkraken, assurez-vous d'être dans le bon dossier/repo et sur la bonne branche
2. Cliquez sur **View changes**
3. Validez les modifications: cliquez sur **Stage all changes**
4. Commentez en respectant les standards proposés (vous serez noté là-dessus):
   * **Summary**: DGN Design du Projet Web
   * **Description**: Modification du CSS, application de la librairie Bootswatch et ajout des images du design.
5. Cliquez sur **Commit**

NOTE: les changements sont enregistrés dans le repo**. LOCAL SEULEMENT**

1. Sur GitHub, dans votre Repository, allez sur la branche de la séance, ouvrez le dossier de la solution ZombieParty

Aucune modification n’est visible

1. "Poussez" (**Push)** les modifications sur le repo en ligne (**Remote).** Cliquez sur 
2. Sur GitHub, allez sur la branche de la séance, ouvrez le dossier de la solution **ZombieParty**
3. Cliquez sur **History**
4. Visualisez la liste des modifications
5. Merge **FCT\_ProjetDesign** into **Main**
6. "Poussez" (**Push)** les modifications sur le repo en ligne (**Remote).** Cliquez sur 

# FonctionnalitéS Zombie et ZOMBIE TYPE

## Création d’une branche pour la fonctionnalité

1. Créez une branche appelée **FCT\_Zombie** à partir de la branche **Main** locale

## Affichage de la liste des Zombies

### Création du *Modèle* Zombie

1. Dans **Models**, créez la classe publique Zombie
2. Ajoutez les propriétés suivantes à la classe:
   * Name (string)
   * Type (string)
   * Point (int)

## Commentaires et validation (Commit) des changements dans le code

1. Dans Gitkraken, assurez-vous d'être dans le bon dossier/repo et sur la bonne branche
2. Cliquez sur **View changes**
3. Validez les modifications: cliquez sur **Stage all changes**
4. Commentez en respectant les standards proposés (vous serez noté là-dessus):
   * **Summary**: FCT Zombie classe
   * **Description**: Ajout de la classe Zombie
5. Cliquez sur **Commit**

NOTE: les changements sont enregistrés dans le repo. LOCAL SEULEMENT

### Création du contrôleur Zombie

1. Ajoutez le contrôleur **Zombie** (Controller MVC – vide)
2. Dans l’action **Index**, ajoutez une nouvelle liste de Zombies dans le **ViewBag**

public IActionResult Index()

{

this.ViewBag.MaListe = new List<Zombie>()

{

new Zombie(){Name= "LeChuck", Point=5, Type="Fiction",},

new Zombie(){Name= "Lenore", Point=4, Type="Fiction",},

new Zombie(){Name= "Draugr", Point=2, Type="Légendaire",},

new Zombie(){Name= "Ragamuffin", Point=5, Type="Légendaire",},

new Zombie(){Name= "Taxidermy", Point=1, Type="Légendaire",},

new Zombie(){Name= "chien de l'enfer", Point=7, Type="Fiction",},

new Zombie(){Name= "Avogadro", Point=9, Type="Fiction",}

};

return View();

}

1. Ajoutez un Point d’arrêt (BreakPoint) sur **Index**
2. Démarrez l’application
3. Dans le navigateur, **LocalHost :nnnnn/Zombie**
4. Dans VS, allez au pas principal suivant (**F10)** etconsultez le contenu du ViewBag
5. Continuez, arrêtez l’application (la vue Index n’existe pas)

### Création de la vue Zombie/Index

1. Créez la vue **Index** partir du Contrôleur (bouton droit sur Index()), View Razor vide
2. Remplacez le contenu par défaut de la vue par le contenu du fichier *S01\_lab2\_FichiersComplementaires/* *View Zombie Index.txt*
3. Testez avec le URL **/Zombie**
4. Ajoutez le code nécessaire pour afficher les informations de chaque élément de la liste (NAME et POINT)

<div class="container p-3">

<div class="row">

@foreach (var item in this.ViewBag.MaListe)

{

<div>

<h3> @item.Name</h3>

<h5> @item.Point</h5>

</div>

}

</div>

</div>

1. Testez avec le URL **/Zombie**

Le contenu de la liste devrait apparaître avec une présentation un peu moche

1. Ajoutez des classes bootstrap et des div afin de présenter les Zombies dans des **Card** Bootstrap ex. :

<div class="container p-3">

<div class="row">

@foreach (var item in this.ViewBag.MaListe)

{

<div href="#" class="card border m-3 text-center bg-warning" style="width: 20rem; background-image: url('/images/imagesLayout/Fond.png'); ">

<div class="card-body text-warning">

<h3 class="card-title"> @item.Name </h3>

<h5> @item.Point </h5>

</div>

</div>

}

</div>

</div>

1. Tester avec le URL **/Zombie**

### Ajouter un menu deroulant pour atteindre la vue Index de Zombie

1. Sur le site de bootstrap, copiez le contenu de la **Navbar, (**seulement la partie du <li> de la classe **nav-item dropdown)**

<https://getbootstrap.com/docs/5.0/components/navbar/>

1. Ajoutez (collez) le contenu dans \_**Layout,** après **Privacy**
2. Changez le nom du menu pour **Zombies** et la 1ère Action pour **Manage** Zombies, qui affiche la vue Index de Zombie. On aura deux actions dans le menu pour ce laboratoire.

<li class="nav-item">

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

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

Zombies

</a>

<ul class="dropdown-menu" aria-labelledby="navbarDropdown">

<li><**a** class="dropdown-item" **asp-controller**="Zombie" **asp-action**="Index">Manage Zombies</**a**></li>

<li><a class="dropdown-item" href="#">Another action</a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="#">Something else here</a></li>

</ul>

</li>

</ul>

1. Tester le menu

## Commentaires et validation (Commit) des changements dans le code

1. Dans Gitkraken, assurez-vous d'être dans le bon dossier/repo et sur la bonne branche
2. Cliquez sur **View changes**
3. Validez les modifications: cliquez sur **Stage all changes**
4. Commentez en respectant les standards proposés (vous serez noté là-dessus):
   * **Summary**: FCT Zombie Index
   * **Description**: Ajout du contrôleur Zombie et de la vue Index associée affichant la liste des zombies. Ajout du menu de gestion avec Bootstrap.
5. Cliquez sur **Commit**

NOTE: les changements sont enregistrés dans le repo. LOCAL SEULEMENT

1. "Poussez" (**Push)** les modifications sur le repo en ligne (**Remote).** Cliquez sur 

## Créer et Afficher ZombieType

### Création du Modèle ZombieType

1. Créez le modèle ZombieType
   * Id (int)
   * TypeName (string)

## Commentaires et validation (Commit) des changements dans le code

1. Dans Gitkraken, assurez-vous d'être dans le bon dossier/repo et sur la bonne branche
2. Cliquez sur **View changes**
3. Validez les modifications: cliquez sur **Stage all changes**
4. Commentez en respectant les standards proposés (vous serez noté là-dessus):
   * **Summary**: FCT ZombieType classe
   * **Description**: Ajout de la classe ZombieType
5. Cliquez sur **Commit**

NOTE: les changements sont enregistrés dans le repo. LOCAL SEULEMENT

### Création du Contrôleur ZombieType

1. Créer le contrôleur
2. Dans l’action **Index**, ajoutez une nouvelle liste de ZombieType dans le **ViewBag**

public IActionResult Index()

{

this.ViewBag.MaListe = new List<ZombieType>()

{

new ZombieType(){TypeName= "Virus", Id=1},

new ZombieType(){TypeName= "Contact", Id=2}

};

return View();

}

### Création de la vue ZombieType/Index

1. Créez la vue **Index** partir du Contrôleur (bouton droit sur Index()), View Razor vide
2. Remplacez le contenu par défaut de la vue par le contenu du fichier *S01.1\_Lab2\_FichiersComplementaires/* *View ZombieType Index.txt*
3. Testez dans le navigateur, **/ZombieType**
4. Ajoutez le code nécessaire pour afficher les informations de chaque élément de la liste de ZombieType
5. Testez dans le navigateur, **/ZombieType**

### Ajouter dans le menu ZombieType

1. Modifiez le \_Layout
2. Changez la 2e action pour **Manage Types**, pour qu’elle affiche la vue **Index** de **ZombieType**.
3. Testez le menu

## Commentaires et validation (Commit) des changements dans le code

1. Dans Gitkraken, assurez-vous d'être dans le bon dossier/repo et sur la bonne branche
2. Cliquez sur **View changes**
3. Validez les modifications: cliquez sur **Stage all changes**
4. Commentez en respectant les standards proposés (vous serez noté là-dessus):
   * **Summary**: FCT ZombieType Index
   * **Description**: Ajout du contrôleur ZombieType et de la vue Index associée affichant la liste des types de zombies. Ajout du menu de gestion avec Bootstrap.
5. Cliquez sur **Commit**

NOTE: les changements sont enregistrés dans le repo. LOCAL SEULEMENT

## Create ZombieType

### Ajouter l’action CREATE au Controller

1. Dans le contrôleur ZombieType, ajoutez l’action Create qui retourne la vue du même nom

//GET CREATE

public IActionResult Create()

{

return View();

}

1. À partir de la vue ZombieType/**Index,** modifiez le bouton **Create New Type** afin qu’il redirige à la vue ZombieType/**Create**

NOTE : One ne fait que le GET, le POST est plus loin

### Ajouter la vue ZombieType/CREATE

1. Créez la vue **Create** àpartir du contrôleur (bouton droit sur Create()), View Razor vide
2. Remplacez le contenu par défaut de la vue par le contenu du fichier *S01.1\_Lab2\_FichiersComplementaires/* *View ZombieType Create.txt*
3. Testez
4. Ajoutez la référence et l’injection au modèle **ZombieType** en haut de la vue Create
5. Ajoutez les TagHelpers pour le formulaire, les libellés et les boîtes d’entrée
6. Ajoutez le TagHelper pour le bouton « Back To List » qui va nous ramener à la vue **Index**
7. Testez

## Commentaires et validation (Commit) des changements dans le code

1. Dans Gitkraken, assurez-vous d'être dans le bon dossier/repo et sur la bonne branche
2. Cliquez sur **View changes**
3. Validez les modifications: cliquez sur **Stage all changes**
4. Commentez en respectant les standards proposés (vous serez noté là-dessus):
   * **Summary**: FCT Zombie Types Create
   * **Description**: Ajout de la vue Create permettant de créer un type de zombies
5. Cliquez sur **Commit**

NOTE: les changements sont enregistrés dans le repo. LOCAL SEULEMENT

1. Poussez (**Push)** les modifications sur le repo en ligne (**Remote).** Cliquez sur 

# Validation règles du modèle

### Ajouter la validation du modèle Côté Serveur

**Petit rappel de 2W5 :**

1. Ajoutez des annotations au modèle **ZombieType** pour l’affichage et la validation

|  |  |
| --- | --- |
| **Élément** | **Règle** |
| TypeName | Dois afficher « Type Name » |
| TypeName | Obligatoire  Message d’erreur : "Type Name have to be fill." |

1. Ajoutez les TagHelpers et les balises nécessaires à la validation du modèle dans la vue ZombieType/**Create**
2. Ajoutez l’action **Create** POST dans le contrôleur

//POST

[HttpPost]

public IActionResult Create(Models.ZombieType zombieType)

{

if (ModelState.IsValid)

{

// Ajouter à la BD

}

return this.View(zombieType);

}

1. Ajoutez un point d’arrêt (BreakPoint) dans le contrôleur **ZombieType**
2. Ajoutez dans la vue les balises pour la validation

@model ZombieParty.Models.ZombieType

<**form** method="post" **asp-action**="Create">

<div class="border p-3">

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

<div class="form-group row">

<h2 class="text-info pl-3">Zombie Type</h2>

<hr />

</div>

<div class="row">

<div class="col-8">

<div class="form-group row">

<div class="col-2">

<**label** **asp-for**="TypeName"></**label**>

</div>

<div class="col-8">

<**input** **asp-for**="TypeName" class="form-control" />

<**span** **asp-validation-for**="TypeName" class="text-danger"></**span**>

</div>

</div>

…

}

1. Testez des données Valides et des données invalides (évidemment, rien n’est enregistré, mais est-ce que les messages d’erreur apparaissent?)
2. En bas de la vue **ZombieType/Create**, ajouter l’appel à la vue partielle (Partial View) de validation côté client

@section Scripts{

@{ <**partial** **name**="\_ValidationScriptsPartial" /> }

}

1. Testez des données Valides et des données invalides

## Commentaires et validation (Commit) des changements dans le code

1. Dans Gitkraken, assurez-vous d'être dans le bon dossier/repo et sur la bonne branche
2. Cliquez sur **View changes**
3. Validez les modifications: cliquez sur **Stage all changes**
4. Commentez en respectant les standards proposés (vous serez noté là-dessus):
   * **Summary**: FCT ZombieType Create et Validation
   * **Description**: Ajout de la validation côté client et côté serveur pour ZombieType.
5. Cliquez sur **Commit**

NOTE: les changements sont enregistrés dans le repo LOCAL SEULEMENT

1. "Poussez" (**Push)** les modifications sur le repo en ligne (**Remote).** Cliquez sur 
2. C’est la fin de la fonction Zombie
3. Merge la branche **FCT\_Zombie** dans la branche **Main.**

Fin du laboratoire