**Chapitre 03**

**La transmission des données d'une vue à un contrôleur**

Ce chapitre explique comment accéder aux données d'une vue dans une méthode d'action du contrôleur. Une méthode d'action est une méthode C # simple qui peut être avec paramètres ou sans paramètres dans le contrôleur.

Les requêtes qui viennent du navigateur, utilisent soit la méthode HTTP GET ou la méthode HTTP POST. Lorsque nous appelons une méthode d'action dans l'URL d'une requête c'est la méthode HTTP GET qui sera utilisée par contre lorsque une requête est générée suite à un événement clic sur un bouton alors c'est la méthode HTTP POST qui sera appelée. Donc, dans cet article, je vais expliquer comment accéder aux données de la vue dans une méthode d'action du contrôleur lorsqu'une requête http GET ou HTTP POST est utilisée.

1. **Quand Les valeurs sont envoyées dans l'url de la requête (http get)**

On peut envoyer des valeurs en paramètres au contrôleur dans l'URL de la requête sous la forme param1=valeur&parm2=valeur&…

**Remarque** : Il faut que les noms des paramètres dans l'URL doivent être les mêmes noms que celle des paramètres de la méthode d'action

Le contenu de la méthode d'action

[HttpGet]

public ActionResult Saisie(int? id,string nom, string prenom)

{

ViewBag.Nom=nom;

ViewBag.Prenom = prenom;

return View();

}

**Le contenu de la vue qui affiche les valeurs reçues**

<head>

<meta name="viewport" content="width=device-width" />

<title>Saisie</title>

</head>

<body>

<h2>voici les paramètres envoyés sur l'url</h2>

Nom:@ViewBag.Nom<br />

Prenom: @ViewBag.Prenom

</body>

</html>

1. **Utilisation de la collection "Form" de l'objet "Request"**

Dans l'approche traditionnelle on utilise l'objet "**Request**" de la classe "HttpRequestBase". La propriété "**Form**" de l'objet "**Request**" contient les valeurs des champs d'entrées dans une paire nom/valeur dont la clé est de type "**string**" et la valeur de type "**StringValues**".

**Remarque**: Les clés à utiliser pour lire des valeurs de l'objet "Request" doivent avoir la même valeur que celle de la propriété "Name" du champ de la vue.

Quand on clique sur le bouton "**submit**" du côté client, une requête de type "POST" est créée et appelle la méthode d'action d'attribut "**HttpPost**".

Le contenu de la vue de saisie

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Saisie</title>

</head>

<body>

<div>

@using (Html.BeginForm("Reception", "Etudiant", FormMethod.Post)) {

<table>

<tr>

<td>Numéro:</td>

<td><input type="text" name="Numero" id="Numero" value=""/></td>

</tr>

<tr>

<td>Nom:</td>

<td><input type="text" name="Nom" id="Nom" value=""/></td>

</tr>

<tr>

<td>Prénom:</td>

<td><input type="text" name="Prenom" id="Prenom" value=""/></td>

</tr>

<tr>

<td>Téléphone:</td>

<td><input type="text" name="Telephone" id="Telephone" value=""/></td>

</tr>

<tr>

<td>Date de Naissance:</td>

<td><input type="text" name="DateNaissance" id="DateNaissance" value=""/></td>

</tr>

</table>

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

}

</div>

</body>

</html>

Le contenu de la méthode d'action

[HttpPost]

public ActionResult Reception()

{

Etudiant e = new Etudiant();

e.Numero=Convert.ToInt32(this.Request.Form["Numero"].ElementAt(0));

e.Nom = this.Request.Form["Nom"].ElementAt(0);

e.PreNom = this.Request.Form["PreNom"].ElementAt(0);

e.Telephone = this.Request.Form["Telephone"].ElementAt(0);

e.DateNaissance = DateTime.ParseExact(this.Request.Form["DateNaissance"].ElementAt(0),"dd/MM/yyyy",null);

ViewBag.Etudiant = e;

return View();

}

Le contenu de la vue de confirmation

@{

Layout = null;

}

@using CoursMVC.Models

@{

var etud = ViewBag.Etudiant;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Saisie</title>

</head>

<body>

<div>

<h2>Vos donnée sont:</h2>

@using (Html.BeginForm("Saisie", "Etudiant", FormMethod.Post))

{

<table>

<tr>

<td>Numéro:</td>

<td><input type="text" name="Numero" id="Numero" value="@etud.Numero" /></td>

</tr>

<tr>

<td>Nom:</td>

<td><input type="text" name="Nom" id="Nom" value="@etud.Nom" /></td>

</tr>

<tr>

<td>Prénom:</td>

<td><input type="text" name="Prenom" id="Prenom" value="@etud.PreNom" /></td>

</tr>

<tr>

<td>Téléphone:</td>

<td><input type="text" name="Telephone" id="Telephone" value="@etud.Telephone" /></td>

</tr>

<tr>

<td>Date de Naissance:</td>

<td><input type="text" name="DateNaissance" id="DateNaissace" value="@etud.DateNaissance" /></td>

</tr>

</table>

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

}

</div>

</body>

</html>

1. **Utilisation d'un objet de type "IFormCollection" comme paramètre**

On peut aussi lire les données postées en utilisant l'objet de type "iFormCollection".

L'objet de type "iFormCollection" contient les données du formulaire sous la forme d'une clé/valeur comme la propriété "Form" de l'objet "Request".

Pour lire les données envoyées par la vue à partir de l'objet de type "IFormCollection", on aura besoin de le faire passer en paramètre à la méthode d'action du contrôleur.

**Le contenu de la vue de saisie**

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Saisie</title>

</head>

<body>

<div>

@using (Html.BeginForm("Reception", "Etudiant", FormMethod.Post)) {

<table>

<tr>

<td><label for="Numero">Numéro:</label></td>

<td><input type="text" name="Numero" id="Numero" value=""/></td>

</tr>

<tr>

<td><label for="Nom">Nom:</label></td>

<td><input type="text" name="Nom" id="Nom" value=""/></td>

</tr>

<tr>

<td><label for="Prenom">Prénom:</label></td>

<td><input type="text" name="Prenom" id="Prenom" value=""/></td>

</tr>

<tr>

<td><label for="Telephone">Téléphone:</label></td>

<td><input type="text" name="Telephone" id="Telephone" value=""/></td>

</tr>

<tr>

<td><label for="DateNaissance">Date de Naissance:</label></td>

<td><input type="text" name="DateNaissance" id="DateNaissance" value=""/></td>

</tr>

</table>

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

}

</div>

</body>

</html>

**Le contenu de la méthode d'action**

[HttpPost]

public ActionResult Reception(IFormCollection formcollection)

{

Etudiant e = new Etudiant();

e.Numero=Convert.ToInt32(formcollection["Numero"].ElementAt(0));

e.Nom = formcollection["Nom"].ElementAt(0);

e.PreNom = formcollection["PreNom"].ElementAt(0);

e.Telephone = formcollection["Telephone"].ElementAt(0);

e.DateNaissance = DateTime.ParseExact(formcollection["DateNaissance"].ElementAt(0), "dd/MM/yyyy", null);

this.ViewBag.Etudiant = e;

return View();

}

1. **L'utilisation d'une suite de paramètres**

On peut passer toutes les valeurs des champs de saisie du formulaire comme paramètres à une méthode d'action du contrôleur.

Remarque:

* Les noms des paramètres doivent être identiques au nom des champs du formulaire. Donc toutes les valeurs entrées par l'usager seront transmises au contrôleur à l'aide de ces paramètres.
* Les champs d'entrée contiennent des valeurs chaine de caractères donc tous les paramètres doivent être de type string.
* L'ordre des paramètres n'est pas important.

Le contenu de méthode d'action

public ActionResult Reception(int Numero, string Nom, string Prenom, string Telephone, string DateNaissance)

{

Etudiant e = new Etudiant();

e.Numero=Numero;

e.Nom = Nom;

e.PreNom = Prenom;

e.Telephone = Telephone;

e.DateNaissance =DateTime.ParseExact(DateNaissance,"dd/MM/yyyy",null);

this.ViewBag.Etudiant = e;

return View();

}

1. **L'utilisation d'un objet comme paramètre**

On peut passer un objet du modèle à une méthode d'action du contrôleur. Dans ce cas-là, on n'a pas besoin de convertir (parse) les valeurs et d'écrire du code en extra.

**Le contenu de la méthode d'action**

[HttpPost]

public ActionResult Reception(Etudiant etud)

{

this.ViewBag.Etudiant = etud;

return View();

}

**Remarque** : La vue qui appelle cette méthode doit être **fortement** **typée**.

1. **L'utilisation de la méthode TryUpdateModelAsync() du contrôleur**

**Elle met à jour l'instance du modèle spécifié en utilisant les valeurs transmises de la vue au contrôleur.**

**Le contenu de la méthode d'action**

[HttpPost]

public async Task<IActionResult> Reception()

{

Etudiant e = new Etudiant();

Await TryUpdateModelAsync<Etudiant>(e); // utilise les valeurs de l'objet Request.

return View(e);

}