

Partage de Photos

Partie 4

Ajout du contrôleur Commentaires

1. Ajoutez Un contrôleur nommé Commentaires en utilisant le modèle « Contrôleur MVC 5 avec vues, utilisant Entity Framework »
2. Supprimez les actions Create, Edit et Details

Ajouter un contrôleur

Classe de modèle :

Classe de contexte de données : +

Utiliser les actions asynchrones du contrôleur

Vues :

Générer des vues

Bibliothèques de scripts de référence

Utiliser une page de disposition : ...

(Laissez vide s'il est défini dans un fichier Razor _viewstart)

Nom du contrôleur :

```
public class CommentairesController : Controller
{
    private GaleriePhotosDb db = new GaleriePhotosDb();

    // GET: Commentaires/Delete/5
    0 références
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Commentaire commentaire = db.Commentaires.Find(id);
        if (commentaire == null)
        {
            return HttpNotFound();
        }
        return View(commentaire);
    }

    // POST: Commentaires/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    0 références
    public ActionResult DeleteConfirmed(int id)
    {
        Commentaire commentaire = db.Commentaires.Find(id);
        db.Commentaires.Remove(commentaire);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    7 références
    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}
```

3. Supprimez toutes les vues du dossier « Views/Commentaires » à l'exception de la vue Delete.cshtml



Action CommentairesPhoto

4. Ajoutez une action partielle qui retourne les commentaires associés à une photo
 - Nom: `_CommentairesPhoto`
 - Portée: public
 - Type de retour: `PartialViewResult`
 - Annotation: `ChildActionOnly`
 - Paramètre: `PhotoId` (l'id de la photo)
 - L'action doit aussi stocker l'id de la photo dans la collection `ViewBag`, qui sera utilisé par le formulaire d'ajout d'un commentaire.

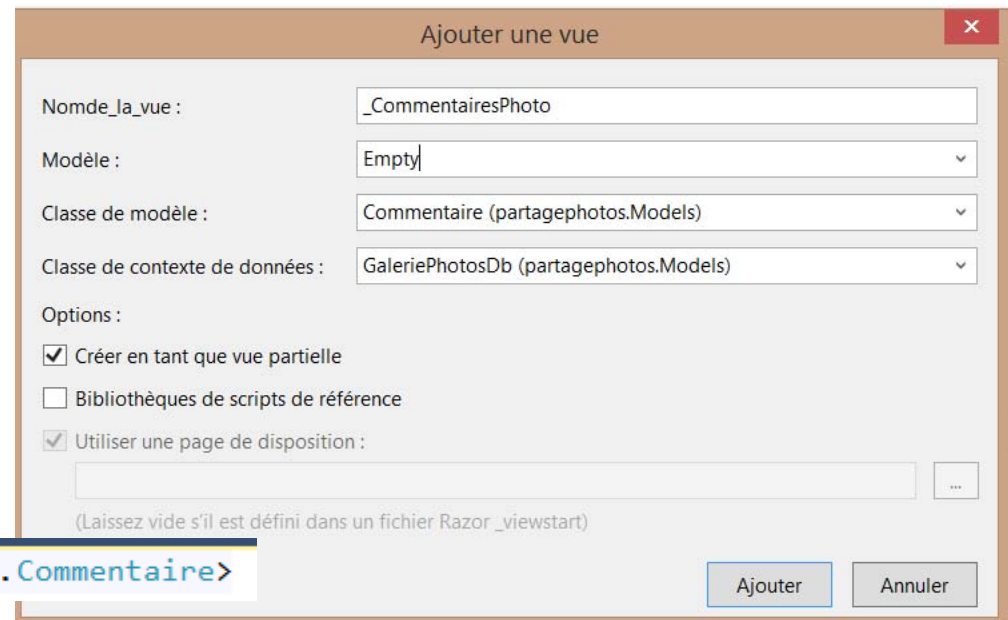
5. Cliquez sur l'action `_CommentairesPhoto` avec le bouton droit et sélectionner la commande « Ajouter une vue »

6. Déplacez la vue dans le dossier Shared.

7. Remplacez la première ligne par:

3

```
@model IEnumerable<partagephotos.Models.Commentaire>
```



8. Ajoutez le code suivant:

```
<h3>Commentaires</h3>
<div id="commentaires_boite">
  <div id="commentaires"></div>
</div>
```

9. Affichez la liste des commentaires dans le div

« commentaires »

- Chaque commentaire doit être affiché dans un div
- l'utilisateur, le sujet et le texte du commentaire doivent être aussi affichés dans des div.

(Remarque: utiliser `@Html.LabelFor` et `@Html.DisplayFor` pour chaque attribut).

- Générer un lien avec `@Html.ActionLink` vers l'action Delete (il faut passer à l'action le paramètre `CommentaireID`)
- Déplacez la vue partielle dans le dossier shared.

Vue partielle _commentairesPhoto

```
@model IEnumerable<partagephotos.Models.Commentaire>
<h3>Commentaires</h3>
<div id="commentairesboite">
  <div id="commentaires">
    @foreach (var item in Model)
    {
      <div>
        <div>
          @Html.LabelFor(i=>item.UserName)
          @Html.DisplayFor(i => item.UserName)
        </div>
        <div>
          @Html.LabelFor(i => item.Sujet)
          @Html.DisplayFor(i => item.Sujet)
        </div>
        <div>
          @Html.LabelFor(i => item.Texte)
          @Html.DisplayFor(i => item.Texte)
        </div>
        @Html.ActionLink("Supprimer", "Delete", new { id = item.CommentaireID })
      </div>
    }
  </div>
</div>
```

Formulaire d'ajout d'un commentaire

Action Creer

L'action `_Creer` affiche le formulaire d'ajout d'un commentaire

- Ajouter l'action partielle `_Creer` dans le contrôleur `Commentaires`
 - Paramètre : id de la photo
 - L'action crée un nouveau commentaire et définit `PhotoId` pour ce commentaire.
 - Retourne la vue partielle `_CreerCommentaire`

Action CommentairesPhoto (HttpPost)

L'action `_CommentairePhoto` permet d'enregistrer un nouveau commentaire dans la base de données et retourne la liste des commentaires mise à jour.

- Paramètres: `Commentaire commentaire`
`int PhotoId`
- Retourne la vue partielle `_CommentairesPhoto` et lui passe la liste des commentaires

Dans la vue CommentairesPhoto

- Dans la vue partielle `_CommentairesPhoto` et sous l'élément `<h3>`, ajoutez un formulaire Ajax pour créer un commentaire, utilisez le helper Ajax avec les paramètres suivants:

- Action: `_CommentairesPhoto`
- L'id de la photo (qui est stocké dans `ViewBag.PhotoId`)
- L'option ajax : `UpdateTargetId: commentairesboite`
- Avant le div fermant de la vue ajoutez (vue partielle `_Creer`):

```
| <div id="ajoutCommentaire" class="add-comment-box">  
|     @Html.Action("_Creer", "Commentaires", new { PhotoID = ViewBag.PhotoId })  
| </div>
```

Vue partielle

CreerCommentaire

La vue partielle `_Creer` affiche les champs du formulaire d'ajout

- Dans le dossier `shared` ajouter une nouvelle vue partielle nommée « `_CreerCommentaire` »

```
@model partagephotos.Models.Commentaire

@Html.ValidationSummary(true)

<div>
  <div>
    <span>
      @Html.LabelFor(model => model.UserName)
    </span>
    <span>
      @Html.EditorFor(model => model.UserName)
      @Html.ValidationMessageFor(model => model.UserName)
    </span>
  </div>
  <div>
    <span>
      @Html.LabelFor(model => model.Sujet)
    </span>
    <span>
      @Html.EditorFor(model => model.Sujet)
      @Html.ValidationMessageFor(model => model.Sujet)
    </span>
  </div>
  <div>
    <span class="editor-label">
      @Html.LabelFor(model => model.Texte)
    </span>
    <span class="editor-field">
      @Html.EditorFor(model => model.Texte)
      @Html.ValidationMessageFor(model => model.Texte)
    </span>
  </div>
  <p>
    <input type="submit" value="Ajouter" />
  </p>
</div>
```