

Exercices

Un contrôleur par exercice.

Exo 1

- Créer une nouvelle application ASP.NET Core en mode MVC
 - Explorer la nouvelle structure d'un projet ASP.NET Core.
-
1. Faire une page TestData qui
 - Affiche la valeur du ViewBag.TestKey
 - Affiche la valeur du TempData["TestKey"]
 2. Faire une action TestData en GET qui
 - Insère une valeur dans ViewBag.TestKey
 - Insère une valeur dans TempData["TestKey"]
 - Retourne la vue TestData
 3. Faire une vue TestData2 qui affiche les mêmes informations que la vue TestData
 4. Faire une action TestData2 en GET qui renvoi juste la vue TestData2
 5. Faire une action TestData3 en GET qui
 - Insère une valeur dans ViewBag.TestKey
 - Insère une valeur dans TempData["TestKey"]
 - Redirige vers l'action TestData2
 6. Faire une action TestData4 en GET qui
 - Insère une valeur dans ViewBag.TestKey
 - Insère une valeur dans TempData["TestKey"]
 - Retourne la vue TestData (comportement non recommandé)

Exo 2

Faire une classe modèle de type Message (Emetteur, Contenu et Date)

7. Faire une page contenant un formulaire qui envoi des informations à un contrôleur en POST
8. Faire une page contenant un formulaire fortement typé (type Message) qui envoi le model à un contrôleur en POST
9. Faire une page fortement typé Message qui affiche un message
10. Faire une page fortement typé liste de Message qui affiche une liste de messages
11. Faire l'envoi en post d'une liste de string
12. Faire l'envoi en post d'une liste de Message
13. Faire un lien HTML qui envoi des informations à un contrôleur en GET

Exo 3

- Faire une vue partielle qui affiche un Message

- Faire une vue qui affiche une liste de Messages en utilisant la vue partielle
- Faire un ViewComponent qui prend en paramètre un Min et Max et qui retourne une vue qui affiche le min, max et un nombre aléatoire entre ce min et max
- Créer un service qui retourne un random en fonction d'un min et max et utiliser ce service dans une action puis dans une vue
- Grâce à un middleware redirigez l'utilisateur vers une page 404 personnalisée

Exo 4

- Appelez le component Random via un clic sur un bouton (passer par une action qui retourne le component)

Exo 5

- Faire de la validation sur un formulaire qui créer un message avec des contraintes
 - Emetteur obligatoire et unique (faire un bool random pour tester)
 - Contenu de 20 caractères max

Exo 6

- Créer une base de données avec EF Core qui contient la table Message
 - Classe Message
 - Classe dbContext
 - Config service dbContext
 - Migrations
- Création d'une couche repository (MessageRepository) qui est un service
- Pages de CRUD Message List Create Read Update Delete

Exo 7

- Faire le CRUD des Messages API
- Mettre en place Swagger
- Créer une classe Utilisateur, 1 utilisateur envoi plusieurs messages et 1 message est envoyé par 1 utilisateur
- Faire la migration EF
- Faire générer le CRUD des utilisateurs (Créer un contrôleur API avec action EF)

Exo 8

- Faire une méthode de test
- Découvrez un projet Identity
- Découvrez un projet Razor Pages
- Découvrez un projet Blazor