Exercices

# Exercice 1 – Formulaire en GET, outils développeurs des navigateurs

Pour débuter sur les formulaires, nous allons créer un premier écran permettant de saisir une addition. Cet écran tout simple contiendra deux champs permettant de saisir l’opérande gauche et droite, puis un bouton “calculer” qui déclenchera le calcul.

Pour ce faire, nous allons :

* Créer une nouvelle action appelée “FormulaireSaisieAddition” dont la seule responsabilité sera d’afficher un formulaire de saisie d’addition à l’utilisateur.
* Cette action ne fera que renvoyer la vue
* Créer la vue correspondante qui contiendra ;
  + Un formulaire paramétré en GET (pour le moment), dont l’action pointera vers une nouvelle action appelée “EnvoiFormulaireAddition” (qui n’existe pas encore, c’est normal)
  + Le formulaire contiendra deux champs de type texte, un pour l’opérande gauche, et l’autre pour l’opérande droite, ainsi qu’un bouton de soumission appelé “Calculer”. Pensez à bien donner un nom unique à chacun de vos champs

Une fois la vue réalisée :

* Lancez l’application, puis naviguez vers cette action via Chrome.
* Saisissez une opérande droite et gauche, mais avant de valider, ouvrez les “outils développeurs” dans Chrome (cf. Google :) ), puis allez sur l’onglet “Réseau”.
* Avec l’onglet “réseau” ouvert, validez le formulaire en cliquant sur le bouton “Calculer” : l’application web renverra une erreur (car l’action pointée n’existe pas), mais surtout, une ligne devrait apparaitre dans l’outil développeur : c’est la requête appelée par le formulaire.
* Analysez cette requête, notamment son adresse de cette requête : il s’agit bien d’une adresse classique telle qu’on aurait pu la saisir dans le navigateur. On a déjà au dernier cours, écrit des actions capables de répondre sur des adresses spécifiées comme telles. A partir de ces informations, vous devriez être capable d’écrire l’action correspondante, avec ses paramètres.
* Avant de continuer, profitez-en pour vous familiariser avec les les outils développeur de Firefox et de Microsoft Edge en faisant de même.

Nous allons donc créer une nouvelle action correspondant à l’adresse appelée par le formulaire, qui créera l’addition en BDD, puis affichera le résultat. Vous avez déjà une action qui fait exactement cela : “Addition”. Mais pour l’exercice, nous allons faire cela en deux temps :

* Tout d’abord, réécrivez-là de zéro, comme si vous ne disposiez pas de l’action “Addition” (voyez cela comme un exercice, rien de plus)
* Puis une fois que cette première étape a été réalisée et fonctionne, pour l’exercice toujours, débrouillez-vous pour que le formulaire appelle maintenant l’opération “Addition” déjà existante.

# Exercice 2 – formulaires en POST et refactorisation

Nous allons maintenant nous intéresser à l’envoi des formulaires en POST.

Sur votre vue qui fonctionne, modifiez la méthode du formulaire pour la positionner en « POST ». Relancez l’application, et revalidez le formulaire en analysant la requête dans les outils développeurs. Vous devriez voir un changement au niveau de :

* L’adresse appelée
* Du passage des paramètres

Prenez le temps de bien retrouver les informations importantes.

En plus de cela, nous allons profiter de cet exercice pour proposer un unique formulaire qui permettrait de créer n’importe quelle opération.

Il s’agira d’effectuer les opérations suivantes :

* Tout d’abord, renommez l’action « FormulaireSaisieAddition » en « FormulaireSaisieOperation »
* Ajoutez dans ce formulaire une liste de sélection HTML (balise « select », cf. votre moteur de recherche préféré) qui permettra de choisir le type d’opération souhaitée
* Renommez l’action « EnvoiFormulaireAddition » en « EnvoiFormulaireOperation » en rajoutant le paramètre qui permettra de récupérer la valeur de la sélection
* Modifiez le code de cette action pour réutiliser ce paramètre lors de la création de l’opération
* Testez le bon fonctionnement de tout cela 😊

# Exercice 3 – Création d’une action de suppression

Nous allons maintenant rajouter une action permettant de supprimer une opération en BDD.

Tout d’abord, sur l’écran de listing des opérations, rajoutez un lien « Supprimer » qui pointera vers une nouvelle action.

Cette action pourra par exemple s’appeler « SupprimerOperation » et prendra en paramètre l’identifiant en BDD de l’opération à supprimer. Le corps de la fonction supprimera l’opération en BDD puis redirigera le navigateur vers l’écran de listing des opérations.

Testez le code développé, et notez que le code de l’action de suppression reste clair.

# Exercice 4 – Refactoring de la création

Maintenant que l’on sait comment séparer le code de son action en plusieurs étapes, nous allons faire la même chose sur les différentes actions d’ajout d’opération que nous avons fait jusque-là.

Jusqu’à présent, le code des actions « Addition », « Suppression » et autres réalisaient deux tâches distinctes :

* Construction puis sauvegarde en BDD de l’opération demandée
* Affichage de l’opération nouvelle créée

Nous allons supprimer cette seconde partie mais rediriger le navigateur vers l’action d’affichage de l’opération créée.

Vous aurez probablement besoin de modifier l’opération SQL d’insertion car en plus d’insérer la donnée en BDD, cette instruction devra vous renvoyez la clé primaire que vous venez d’insérer. Rendez-vous sur votre moteur de recherche préféré pour savoir quel code SQL Server exécuter pour cela 😉. Au final, l’algorithme de votre fonction de création devrait être de la sorte :

* Création d’une instance d’opération à partir des informations saisies par l’utilisateur
* Ajout de l’opération en BDD, récupération immédiate de la clé primaire insérée
* Renvoi du code http 301 – Redirect vers l’action d’affichage de l’opération nouvelle crée (à partir de son identifiant)

# Exercice 5 – Formulaire d’édition

Nous allons maintenant rajouter une action permettant de modifier une opération déjà existante. Sur le listing, rajoutez un nouveau lien vers cette action, et développez là 😊

# Exercice 6 – Formulaire de contact

Nous allons maintenant rajouter une nouvelle page permettant à un utilisateur de simuler l’envoi d’un mail de contact. La page demandera à l’utilisateur de renseigner un message, une adresse mail de contact, un nom et un prénom.

* Réalisez déjà l’action permettant d’afficher le formulaire en question
* Puis, réalisez l’action pointée par le formulaire en permettant de simuler l’envoi du mail
  + Simulez l’envoi du mail soit en sauvegardant le contenu du mail dans un fichier, en BDD, ou pour les plus téméraires, en envoyant réellement le fameux mail.
  + Passez les paramètres en utilisant une classe plutôt qu’un ensemble de paramètres séparés.