

Cours-TP-3 : POST, les cookies

Le contenu de ce TP sera à rendre avec le suivant

Avant-propos : premières manipulation des requêtes en POST

Les questions sont faciles : ce n'est pas un piège, il s'agit de s'habituer à manipuler les outils.

Question 1 : Complétez le point d'entrée vers l'url <http://localhost:8080/helloWorld> en POST, qui écrit "Bonjour Console" sur la sortie standard du serveur et envoie "bonjour" au client. Vous pouvez vous inspirer du code de myFirstEndPoint.

Question 2 : Utiliser insomnia (ou un autre outil permettant d'envoyer des requêtes HTTP) pour tester votre point d'entrée précédent. Pourquoi ne peut-on pas utiliser directement un navigateur ?

Question 3 : Tester le point d'entrée <http://localhost:8080/helloWorldRedirect> en POST, avec et sans corps pour la requête. Que constatez-vous ?

Question 4 : En vous inspirant de showParamFromGet, complétez la méthode showParamFromPost pour que les paramètres de la requête du client soient affichés sur la sortie standard du serveur. Avez-vous besoin d'écrire un code différent ?

Question 5 : Tester votre point d'entrée. Comment indiquer les paramètres de la requête en POST ?

Personnages et cookies

Dans cette partie, les modifications du personnage courant seront gérées directement dans le contrôleur (sans passer par la création d'une instance dans le modèle)

Question 6 : observez le contenu du modèle. Où appeler Universe.creation(); pour que les éléments soient créés au lancement du serveur ? Que signifie de static devant l'attribut characters de la classe Universe ?

Question 7 : Compléter la méthode choiceCharacter pour que celle-ci récupère le paramètre "name" et affiche sur la sortie standard sa valeur. Testez-la.

Question 8 : Modifiez `choiceCharacter` pour chercher dans `Universe.getCharacters()` un personnage de même nom, et, dans le cas où celui-ci existe, afficher ses caractéristiques sur la sortie standard du serveur. Testez

Question 9 : Modifiez `choiceCharacter` pour attacher à la réponse des cookies contenant des liens entre "name", "HP" et "attack" et leurs valeurs correspondante pour le personnage choisi, dans le cas où celui-ci existe. Testez.

Question 10 : Ecrivez la méthode `showCharacter` pour que celle-ci affiche (en texte brut dans le corps de la réponse) le contenu des cookies "name", "attack" et "HP" si ceux-ci existent (peu importe l'ordre)

Question O1 (optionnelle) : faites en sorte d'avoir un affichage similaire à celui donné par le `toString()` de `Character`

Question 11 : Complétez la méthode `takeDammage` pour qu'un appel à <http://localhost:8080/takeDammage?dammage=16> réduise de 16 (ou du paramètre donné) les HP du personnage, avec un minimum de 0, dans le cookie approprié. La méthode redirigera ensuite vers la la méthode `showCharacter`.

Question O2 (optionnelle) : ajouter la gestion du cas où les cookies appropriés n'existent pas, à la manière de votre choix. (par exemple, redirection vers une page indiquant qu'aucun personnage n'a été sélectionné).