

## Programmation Web

### TP n° 3b : Formulaire en express

Les données d'une requête en POST émises par un formulaire web peuvent être récupérées par un serveur Express, à condition de faire suivre la création du serveur d'une invocation de la forme ci-dessous :

```
1 const express = require('express');
2 const server = express();
3 server.use(express.urlencoded({extended: true}));
```

Dans un gestionnaire de requêtes en POST, `req.body` devient alors une référence vers un objet contenant une propriété par `input` nommé du formulaire, le nom de la propriété étant le nom de l'`input`, et sa valeur celle de l'`input` au moment de l'envoi de la requête.

**Exemple.** Un formulaire contient trois `input` de type `text`, de noms `prenom`, `nom`, `adresse`. Sa méthode est `"post"`, son action `"http://localhost:8080/"`.

Un utilisateur remplit les deux premiers champs du formulaire par John et Doe, laisse vide le champ `adresse`, puis soumet ces données au serveur. Si le code du serveur contient un élément de route défini par :

```
1 server.post('/', function (req, res) {
2     console.log(req.body);
3     // ...
4 });
```

le message affiché dans le terminal sera : `{ prenom: 'John', nom: 'Doe', adresse: ''}`. Autrement dit, `req.body.prenom`, `req.body.nom` et permet `req.body.adresse` permettent de récupérer les valeurs des trois champs.

**Remarque.** Si l'on se sert de EJS et si l'on souhaite dans un modèle de page laisser le serveur spécifier la valeur d'un `input` – par exemple pour préremplir le formulaire – il est impossible de le faire via des balises `<%= ... %>` écrites directement dans la balise `<input>`. En revanche, rien n'empêche d'écrire par exemple dans le modèle (à la toute fin de `<body>`), et avec des guillemets) :

```
1 <script>
2     document.getElementById("nom").value = "<%= nom %>";
3 </script>
```

où `nom` entre les balises `<%=` et `%>` est une propriété de l'objet externe fourni par le serveur au moment du rendu de la page (`res.render`).

**1. Formulaire d'inscription** On souhaite créer un serveur simulant l'inscription d'utilisateurs associés à un mot de passe dans une base de données. La simulation sera simpliste : les couples d'utilisateurs et de mots de passe seront simplement accumulés dans un unique tableau d'objets.

Créer un modèle de page web `form.ejs` contenant un formulaire à deux champs : un champ nom, un champ mot de passe (par exemple sur le modèle de l'Exercice 2 du TP 1). L'en-tête de la page (`<h3>`) et les valeurs des deux champs seront fournis par le serveur (*c.f.* la remarque ci-dessus). La méthode du formulaire est `"post"`, son action `"http://localhost:8080/"`. Écrire ensuite un serveur de comportement suivant :

1. A la première requête `GET` vers `/`, le serveur renvoie une page de modèle `form.ejs` avec l'en-tête `"Veuillez remplir ce formulaire"` et des valeurs de champs vides.
2. A la réception d'une requête `POST` vers `/`, le serveur examine les données reçues :
  - (a) Si l'un des deux champs n'est pas rempli, le serveur renvoie une nouvelle page de même modèle avec l'en-tête `"Veuillez remplir tous les champs"`, en inscrivant dans cette page les valeurs déjà écrites dans la précédente. Sinon, le serveur consulte sa base de données.
  - (b) Si l'utilisateur est déjà inscrit, le serveur met à jour son mot de passe, puis renvoie une page web d'un second modèle avec un message de la forme : `"Bonjour nom de l'utilisateur, votre mot de passe a été mis à jour."`
  - (c) Sinon, le serveur ajoute le nom de l'utilisateur et son mot de passe à sa base de données, puis renvoie une page web d'un troisième modèle, avec un message de la forme : `"Bonjour nom de l'utilisateur, vous avez été enregistré."`

## 2. QCM

Écrire un serveur couplé à une modèle de page web permettant de répondre à un QCM d'un nombre quelconque de questions à 3 réponses.

Les questions sont numérotées à partir de 1. Lorsqu'une question est posée pour la première fois à l'utilisateur, celui-ci reçoit une page web affichant le numéro de cette question, le nombre total de questions (*e.g.* `"Question 3 sur 42"`), ainsi qu'un formulaire lui permettant de choisir une réponse à la question courante. Lorsqu'une réponse de l'utilisateur est incorrecte, la même question lui est reposée, précédée de l'indication `"Erreur, recommencez."`. Lorsque l'utilisateur a répondu à toutes les questions, la page se contente d'afficher le texte `"Bravo !"`.

*Remarque.* Pour cet exercice, vous êtes volontairement moins guidé. Un seul modèle de document suffit, et il n'a pas à contenir plus d'un formulaire. A vous de trouver quelles sont les informations qu'il faut lui communiquer.