

Programmation Web client riche

TP2 : Ajax

Introduction :

AJAX (acronyme de *Asynchronous Javascript and XML*) est une technique utilisée pour créer des pages web dynamiques. C'est la combinaison de technologies telles que Javascript, CSS, XML, le DOM et le XMLHttpRequest.

Elle permet la mise à jour des pages web, sans devoir les recharger, en échangeant d'une manière asynchrone des données avec le serveur web.

Exemples de sites utilisant Ajax : Google Maps, youtube, Facebook, ...

Principe :

Suite à un événement, une requête HTTP sera envoyée d'une manière asynchrone au serveur HTTP (la requête peut comprendre des données insérées par l'internaute). Le serveur va traiter cette requête et y répondre d'une manière asynchrone. Une fois la réponse est reçue par le client, elle lancera la mise à jour de la page web affichée dans le navigateur du client sans devoir recharger la page web.

Les étapes nécessaires pour envoyer et recevoir une requête Ajax :

1. Créer un objet XMLHttpRequest : `xmlhttp=new XMLHttpRequest();`
2. Définir les instructions à exécuter lorsque la réponse est retournée au client :
`xmlhttp.onreadystatechange=function(){ document.write(xmlhttp.responseText); }`
3. Configurer la requête : `xmlhttp.open("méthode","URL",mode d'envoi);`
Méthode : **GET** ou **POST**
URL : adresse sur le serveur, par exemple : `web.php`. Avec la méthode GET on peut envoyer des données, par exemple : `web.php?q=10`, pour envoyer 10 dans la variable q.
Mode d'envoi : **true** pour asynchrone ou **false** pour synchrone (false)
4. Envoyer la requête : `xmlhttp.send()`. Avec la méthode POST, on peut envoyer des données avec la méthode `send()`, par exemple : `xmlhttp.send("q=10&nom=dupont")`

Exemple :

Description : Afficher les prénoms stockés sur le serveur et qui commencent par les lettres introduites dans le champ de texte.

Code :

Page HTML :

```
<html>
  <head>
    <script type="text/javascript">
      function showHint(data){
        if (data.length==0){
          document.getElementById("txtHint").innerHTML="";
          return;
        }
        xmlhttp=new XMLHttpRequest();
        xmlhttp.onreadystatechange=function(){
          if (xmlhttp.readyState==4 && xmlhttp.status==200)
            document.getElementById("txtHint").innerHTML=
              xmlhttp.responseText;
        }
        xmlhttp.open("GET", " /api/hint?data="+data, true);
        xmlhttp.send();
      }
    </script>
```

```

</head>
<body>
  <h3>Commencer à taper un nom dans le champs :</h3>
  Nom: <input type="text" id="txt1" onkeyup="showHint (this.value)" />
  <p>Suggestions: <span id="txtHint"></span></p>
</body>
</html>

```

Script python:

```

from flask import request
@app.route('/api/hint', methods=['GET'])
def getHint():
    names=["Anna", "Brittany", "Cinderella", "Diana", ... , "Vicky"]
    data=request.args["data"]
    hint=""
    for name in names:
        if data.lower()==name[0:len(data)]:
            if hint=="":
                hint=name
            else:
                hint=hint+" , "+name
    return hint

```

Pour plus d'informations, consulter le diaporama Ajax.

Exercice :

Créez dans votre base de données MySQL les 3 tables (clients, services, carte de réduction) en important le script sql fourni. Puis, créez un formulaire qui permet de sélectionner le numéro de référence d'un client. Puis avec ajax, récupère automatiquement de la base de données les informations (civilité, prénom, nom, adresse, code postale, ville, carte) concernant ce client et les affiche dans le formulaire. Puis dans le tableau réservé aux services commandés, l'utilisateur peut sélectionner le nom du produit d'une liste déroulante (les produits sont ajoutés à la liste lors du chargement du formulaire), les informations concernant le service sont affichées et le montant total de la facture est calculé en fonction des services choisies et la carte de réduction du client. Le formulaire doit ressembler à la figure suivante :

**ski club
à votre service**

12 impasse des moulinet
90000 Belfort
3 84 56 47 89
6 54 89 57 23

17/2/2011

Facture

Votre code client

Monsieur Bertin Théophile
12 rue des près
90000 Belfort

code prestation	désignation	prix	quantité	montant
1	raquettes	3 €	<input type="text" value="3"/>	9 €
3	skis paraboliques	3.25 €	<input type="text" value="5"/>	16.25 €
	vide	0 €	<input type="text" value="0"/>	0 €
	vide	0 €	<input type="text" value="0"/>	0 €
	vide	0 €	<input type="text" value="0"/>	0 €
	vide	0 €	<input type="text" value="0"/>	0 €
	vide	0 €	<input type="text" value="0"/>	0 €
somme				25.25 €
remise				10%
à régler				22.725 €

votre carte est verte

Résoudre l'exercice en effectuant les étapes suivantes :

1. Importez le script sql dans votre base de données MySQL.
2. Développer une fonction JavaScript qui prend une chaîne de caractère contenant des données séparées par des séparateurs, par exemple : « jean,dupont,rue de la fontaine,92000,Paris », et un séparateur (, ou ;) en paramètre et permet d'extraire ces données et les stocker dans un vecteur. Utiliser la méthode split de la classe String.
3. Créer une application qui récupère avec une requête Ajax les données concernant l'identifiant du client saisi dans un champ de texte et les affiche dans une page web. Utiliser la fonction développée dans l'étape précédente pour extraire les données de la réponse à la requête Ajax.
4. Créer une page web qui permet de récupérer avec une requête Ajax, les noms des produits et les afficher dans une liste déroulante au chargement (événement onload) de la page web.
5. Modifier la page web précédente afin de pouvoir récupérer avec Ajax les informations concernant un produit sélectionné dans la liste déroulante et les afficher.
6. Créer la facture affichée dans la figure ci-dessus. Avec javascript, écrire toutes les fonctions nécessaires pour calculer le montant total de la facture en fonction des produits sélectionnés.
7. Ajouter les fonctionnalités développées en 3, 4 et 5 à la facture.
8. Ajouter la fonctionnalité de remise selon la carte de fidélité du client.
9. Ajouter deux boutons qui permettent d'augmenter le nombre de lignes dans la commande ou de supprimer des lignes de la commande.
10. Créer les tables suivantes : Commande(numCommande, prixTotal, dateCommande,
 codeClient*)
 Contient(numCommande, codeProduit, quantité)
11. Ajouter un bouton « Sauvegarder » qui permet en utilisant Ajax de sauvegarder dans la base de données la commande saisie.
12. Ajouter un bouton « Créer nouvelle facture » qui permet d'effacer toutes les données insérées par l'utilisateur et de commencer la saisie d'une nouvelle facture.
13. Ajoutez une liste déroulante qui s'affiche lorsque le code d'un client est inséré et permet la sélection et l'affichage des détails d'une commande déjà passée. La liste déroulante doit être remplie dynamiquement par une requête Ajax. On pourra par suite modifier une commande déjà effectuée et la sauvegarder.