Travaux pratiques LbD4

Pr. Ahmed Bentajer

TP 3: Transmission des données

Résumé cours

Lors des communications WEB, les navigateurs Web communiquent avec le serveur en utilisant une des 9 méthodes HTTP[1] (GET , POST, PUT, PATCH, DELETE, HEAD, TRACE, CONNECT). Dans ce TP on s'intéresse aux méthodes POST et GET qui permettent la transmission de l'information différemment et présentent des avantages et des inconvénients.

Notez bien que la transmission des données à travers GET et POST ne se fait pas à travers PHP mais via le protocole HTTP. PHP implémente les mécanismes nécessaires pour récupérer ses variables à travers des variables super globale GET, POST et REQUEST.

Une variable superglobale est une variable PHP particulière créée par le langage et non par le développeur. Elle a comme particularité d'être disponible partout dans le code (d'où son nom). La plupart du temps, elle commence par le symbole **$_**. Elle s'utilise comme un tableau associatif associant des clés et des valeurs.

HTTP GET

Introduction

La méthodes GET permet de transmettre les données dans l'URL votre site sous la forme

http://www.site.com/destination-get.php?nom=bentajer&prenom=ahmed

Dans cette exemple la page **destination-get.php** est appelée et recevra les données **bentajer** et **ahmed** qui sont stockées respectivement dans les variables **nom** et **prénom.** L'opérateur **&** permet de séparer les différentes paires. La page et l'information codées sont séparées par **?.**

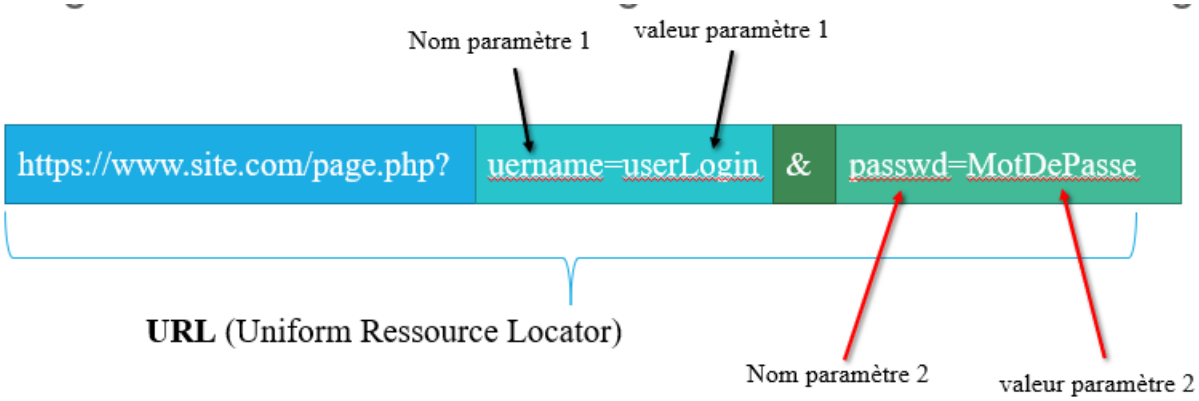
Dans une page HTML/PHP l'envoi des données avec GET peut se faire avec deux manières différentes :

- En utilisant l'attribut <a href='destination.php?key1=value1&key2=value2'>…
- Via les formulaires ayant l'attribut **method='GET'**

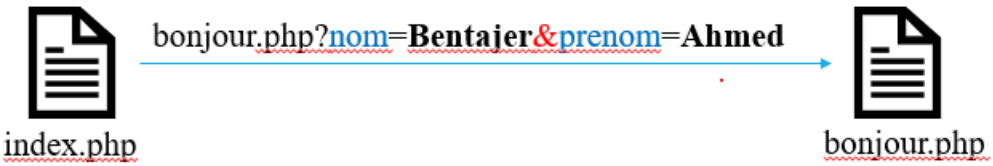
[1] https://www.rfc-editor.org/rfc/rfc9110.html

Transmission via URL

Dans la page de destination pour récupérer les données envoyées à travers GET on utilisera la variable super globale $_GET qui est un tableau associatif ayant comme clef **Nom Paramètre** et comme valeur **Valeur paramètre** (Voir figure ci-dessous)



Dans la figure ci-dessous, les clefs sont ***nom*** et ***prenom*** et leurs valeurs sont respectivement ***Bentajer*** et ***Ahmed***



Dans ce cas, la page **bonjour.php** recevra ces paramètres dans la variable superglobale $_GET et défini automatiquement par PHP :

- $_GET['nom'] aura pour valeur Bentajer ;
- $_GET['prenom'] aura pour valeur Ahmed.

Transmission via le formulaire

Les formulaires sont le moyen le plus pratiques permettant à un utilisateur d'échanger les données avec un site web.

La balise *HTML form*, possède deux attributs essentiels :

- **Action :** permet de définir l'URL cible qui recevra/traitera les information envoyées par le formulaire
- **Method :** Permet de définir la requête HTTP utilisée pour envoyer les données vers la page renseignée dans l'attribut **action.** Dans le cas de la figure ci-dessous les données seront transmises en utilisant la méthode http GET

Dans le cas où un formulaire est utilisé pour envoyer les données en utilisant la méthode GET, la clef est la valeur de l'attribut nom, et sa valeur est le texte saisi par l'utilisateur.

```
<form action="destination-get.php" method="get">
    <input type="text" name="fname"><br><br>
    <input type="text" name="lname"><br><br>
    <input type="submit" value="Submit">
</form>
```

A l'intérieur d'un formulaire, les balises HTML <input> permettent de définir des champs de saisie pour l'utilisateur. L'attribut **name** d'une balise <input> définit le nom de la variable qui contiendra la valeur saisie.

Suivant la méthode de soumission utilisée GET ou POST (Dans notre cas la méthode GET), les données envoyées via un formulaire sont ajoutées dans le corps de la requête HTTP et se retrouvent dans un tableau associatif nommé $_GET défini automatiquement par PHP (figure ci-dessous).

```
$nom_visiteur = $_GET['nom'];
$prenom_visiteur = $_GET['prenom'];
```

Dans les balises HTML tel (select, <input type='radio' … >, ….) la valeur récupérée et celle de l'attribut value de la balise (figure ci-dessous).

```
<form action="destination-get.php" method="get">
    <select name="niveau" size="1">
        <option value="CP1"> 1er année </option>
        <option value="CP2"> 2eme année</option>
    </select>

    <input type="radio" name="appelation" value="Mme" /> Madame<br />
    <input type="radio" name="appelation" value="M" /> Monsieur<br />
    <input type="submit" value="Submit">
</form>
```

Ainsi pour récupérer ses valeurs dans la page ***destination-get.php*** on utilisera le code de la figure ci-dessous :

```php
$niveau_utilisateur = $_GET['niveau'];
$appelation = $_GET['appelation'];
```

Les valeurs de niveau peuvent être soit CP1 ou CP2 ainsi que les valeurs du select peuvent être soit Mme ou M.

HTTP POST

La transmission de données par la méthode *POST* se fait généralement à travers les formulaires dans le script sans passer par l'URL, il s'agit de la méthode la plus couramment utilisée.

La variable superglobal correspondante est $_POST.

La même logique décrite pour $_GET s'applique sur $_POST, sauf pour la partie de la transmission des données via les URL

Informations complémentaires

Avant de manipuler les paramètres transmis dans l'URL il faudra s'assurer que le paramètres est défini ou non à l'aide de la fonction PHP **isset().**

```php
if(isset($_GET['nom']) && isset($_GET['prenom'])){
    // TRAITEMENT A FAIRE
}
```

Pour se prémunir contre ces risques, il faut appliquer un principe très important ***Never Trust User's Input***. Un développeur est obligé a effectué des contrôles sur les saisies des utilisateurs pour ne pas prendre le risque d'exposer son site à des attaques très faciles à réaliser, tel :

- L'injection XSS (Cross-Site Scripting).
- L'injection SQL dans une base données,

Pour réaliser une injection XSS, un utilisateur pourra saisir code JavaScript au lieu d'une valeur "normale".

<script type="text/javascript">alert(Coucou !')</script>

Si la valeur saisie est directement affichée par la page PHP, le résultat HTML généré contiendra un script JavaScript imprévu qui sera exécuté par le navigateur client. Cela peut potentiellement causer de gros problèmes de sécurité.

Un premier niveau de sécurité, qui doit devenir un réflexe, consiste à "nettoyer" toute donnée externe avant de l'utiliser pour générer une page Web. Il existe plusieurs fonctions PHP qui limitent fortement le risque d'injection de code JavaScript. Le choix le plus fréquent est la fonction **htmlspecialchars**. Cette fonction remplace certains caractères spéciaux par des entités HTML.

```php
$nom_visiteur = htmlspecialchars($_GET['nom']);
$prenom_visiteur = htmlspecialchars($_GET['prenom']);

$niveau_utilisateur = htmlspecialchars($_GET['niveau']);
$appelation = htmlspecialchars($_GET['appelation']);
```

Exercices

Les CSS réalisé dans ce TP doivent se faire avec le framework bootstrap

Exercice 1

En se basant sur l'exercice 8 du TP 2 sur les tableaux, créez une pages index contenant des liens vers la page **show_student_infos.php**. Ces liens vont transmettre à la page **show_student_infos.php** le code de l'étudiant (ex. ET123, …) que par la suite vous devez utiliser pour afficher les informations de l'étudiant dans une table HTML. (Utiliser le code HTML et CSS nécessaire)

Valider votre page sur : https://validator.w3.org/

Exercice 2

En se basant sur l'exercice 6 du TP 2, créer un formulaire permettant à l'utilisateur de choisir un domaine qu'il désire apprendre (WEB, BD, Réseau, ….) puis afficher les informations sur les sites web recommandées. Le résultat doit être sous format d'une table HTML/CSS

Exercice 3

Dans une page **index.php** créer un formulaire d'envois de courriels.

Le formulaire doit contenir les champs : Nom, Prénom, email, Objet, une liste pour sélectionner si l'utilisateur veut contacter le service après-vente ou techniques et enfin le corps de Message.

Afficher ensuite les informations saisies par l'utilisateur dans la page destination.

Exercice 4

Refaire l'exercice 3 mais cette fois en n'utilisant que la page **index.php** pour afficher le formulaire et les informations que l'utilisateur a saisis.

Exercice 5

Écrire un formulaire **calculatrice.php** avec 2 cases pour la saisie des opérandes, un groupe de 4 cases à cocher pour le choix de l'opération, puis afficher le résultat de l'opération dans la même page.

Exercice 6

L'objectif ici est de contrôler la saisie des utilisateurs

On souhaite créer un formulaire d'inscription **inscription.php** des visiteurs qui contient les champs suivants:

- Civilité: liste de sélection qui contient les options Mlle, Mme et M.
- Deux champs texte (type="text") avec les labels nom et prénom
- Un champ date (type="date") avec label Date de naissance
- Un groupe de 10 cases à cocher (type="checkbox" name="formations[]) contenant les formations proposées
- Un champ mot de passe (type="password") avec label Mot de passe
- Un champ Récrire mot de passe (type="password") avec label Réécrire mot de passe
- Un bouton de soumission du formulaire (type="submit")

La méthode POST sera utilisée pour l'envoi du formulaire.

L'attribut action du formulaire devra être vide.

Le script vérifie l'existence des données et les affichant dans un tableau présenté en HTML et CSS.

L'utilisateur doit au moins sélectionner une case à cocher.

Si par exemple le client laisse un champ vide on affichera un message "Champs laissé vide" (ou champs doit être remplace par le nom du champs) en rouge, ou si les deux mots de passes renseignés ne sont pas identique le message "Mots de passes non identiques" sera affiché.

Si tous les champs sont valides, alors on affichera les informations saisis par le client.

Dans le cas où un champ n'a pas été renseigne ou il y a un problème dans les mots de passes, il faudra conserver les valeurs déjà saisis au sein de formulaires.

Exercice 7

Un site bancaire une solution de plan de financement pour l'achat d'un bien immobilier.

Pour cela on vous demande de créer une page information.html contenant un formulaire permettant au client de saisir les informations nécessaires pour le calcule du plan de financement, ses informations sont :

- Une liste déroulante contenant les informations suivantes : Salarié, Fonctionnaire, Profession libérale

- Des champs permettant la saisie du nom, prénom,
- Un champ montant de financement
- Un champ duré en mois (max 300 mois).
- Case à cocher permettant à l'utilisateur s'il veut une assurance ou non (Par défaut la case Oui est sélectionnée)

Suivant la fonction du client les taux suivants s'appliquent :

- 4% pour les fonctionnaires (soit 4/100/12 donc 0.4% par mois)
- 5% pour les salariés
- 6% pour la profession libérale.

Si l'option assurance est OUI alors un taux de 0.04% du capitale devra être ajouter aux mensualités.

1. Valider la saisie des données par l'utilisateur
    a. Nom et Prénom doivent être non vide
    b. Champs montant doit être un nombre réel
    c. Champ durée doit être un entier entre 6 et 300
2. Afficher le tableau d'amortissement (ci-dessous un exemple de tableau d'amortissement)

Date Ech	Mensualité (MAD)	Intérêt (MAD)	Amortissement (MAD)	Cap. restant dû (MAD)
01/03/2023	2 922,95	2 083,33	839,62	499 160,38
01/04/2023	2 922,95	2 079,83	843,12	498 317,27
01/05/2023	2 922,95	2 076,32	846,63	497 470,64
01/06/2023	2 922,95	2 072,79	850,16	496 620,48
01/07/2023	2 922,95	2 069,25	853,7	495 766,79
01/08/2023	2 922,95	2 065,69	857,26	494 909,53
01/09/2023	2 922,95	2 062,12	860,83	494 048,7

Exercice 8

La banque veut mettre à jour sa solution, pour cela elle vous demande d'ajouter les champs suivants au formulaire :

- Un champ date de naissance,
- Un champ pour saisir le salaire du visiteur
- Un champ pour saisir le montant mensuel total des crédits que le client paye

Avant d'afficher le tableau d'amortissement, vous devez :

- Calculer l'âge de l'emprunteur et s'assurer que la durée qu'il demande se trouve dans l'intervalle de fonction (retraite a 62 ans)

- Si la mensualité du crédit demandé + total des mensualités des autres crédits > 50% du salaire alors refuser le crédit mais en proposant le montant max que peut avoir l'emprunteur de tel sorte que la somme des deux mensualités soit <= 52% du salaire