

Développer un site web dynamique en PHP

**Module 6 : Les échanges de données
entre le client et le serveur**



Objectifs

- Comprendre le protocole de communication HTTP
- Effectuer une redirection vers une autre page web
- Déposer, consulter et supprimer un cookie
- Gérer une session
- Mettre en place une communication asynchrone

Les échanges de données entre le client et le serveur

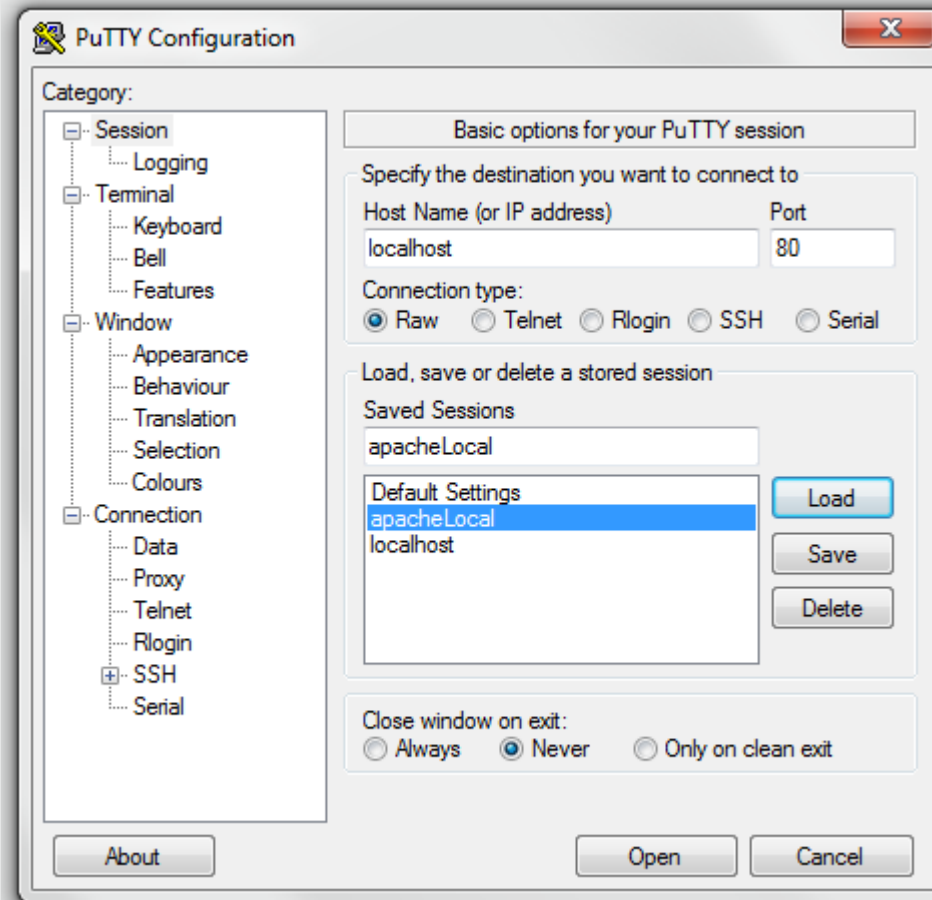
Le protocole HTTP

- Langage de communication entre le client et le serveur
- Commandes :
 - GET : récupération d'une page
 - HEAD : récupération uniquement de l'entête
 - POST : envoi de données en vue d'un traitement
 - OPTIONS : obtention des option de communication
 - CONNECT : utilisation d'un proxy comme tunnel de communication
 - TRACE : répétition de ce qui a été envoyé
 - PUT : envoi d'une ressource sur le serveur
 - PATCH : modification d'une ressource sur le serveur
 - DELETE : suppression d'une ressource sur le serveur

Les échanges de données entre le client et le serveur

Le protocole HTTP

- Test avec PuTTY



Les échanges de données entre le client et le serveur

Le protocole HTTP



- GET : récupération d'une page

```
GET /CoursPHP/3-formulaires/traitementAPart/index.php HTTP/1.0

HTTP/1.1 200 OK
Date: Fri, 01 Sep 2017 08:15:15 GMT
Server: Apache/2.4.23 (Win64) PHP/7.0.10
X-Powered-By: PHP/7.0.10
Content-Length: 4542
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <title>Formulaire traitement à part</title>
  ...
```

Les échanges de données entre le client et le serveur

Le protocole HTTP



- GET pour une page inexistante

```
GET /inexistante.php HTTP/1.0
```

```
HTTP/1.1 404 Not Found
```

```
Date: Fri, 01 Sep 2017 08:32:03 GMT
```

```
Server: Apache/2.4.23 (Win64) PHP/7.0.10
```

```
Content-Length: 297
```

```
Connection: close
```

```
Content-Type: text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

```
    <html><head>
```

```
        <title>404 Not Found</title>
```

```
    </head><body>
```

```
        <h1>Not Found</h1>
```

```
        <p>The requested URL /inexistante.php was not found on this server.</p>
```

```
        <hr>
```

```
        <address>Apache/2.4.23 (Win64) PHP/7.0.10 Server at localhost Port 80</address>
```

```
    </body></html>
```

Les échanges de données entre le client et le serveur

Le protocole HTTP

- Codes de statut
 - Information 1xx
 - Succès 2xx
 - 200 : OK : Requête traitée avec succès
 - Redirection 3xx
 - 301 : Moved Permanently : Document déplacé de façon permanente
 - 302 : Moved Temporarily : Document déplacé de façon temporaire
 - Erreur client 4xx
 - 403: Forbidden : Le serveur a compris la requête, mais refuse de l'exécuter.
(Problème de droits)
 - 404 : Not Found : Ressource non trouvée
 - Erreur serveur 5xx
 - 500 : Internal Server Error : Erreur interne du serveur
 - 503 : Service Unavailable : Service temporairement indisponible ou en maintenance

Les échanges de données entre le client et le serveur

Le protocole HTTP

- HEAD : récupération uniquement de l'en-tête



```
HEAD /CoursPHP/3-formulaires/traitementAPart/index.php HTTP/1.0

HTTP/1.1 200 OK
Date: Fri, 01 Sep 2017 08:17:00 GMT
Server: Apache/2.4.23 (Win64) PHP/7.0.10
X-Powered-By: PHP/7.0.10
Connection: close
Content-Type: text/html; charset=UTF-8
```


Les échanges de données entre le client et le serveur

Le protocole HTTP



- GET avec transmission de données

```
GET /CoursPHP/3-formulaires/traitementAPart/formulaireTraitement.php?nom=a&saisie%5Bage%5D=14
&saisie%5Bemail%5D=gege%40truc.com&saisie%5Bmdp%5D=123AZerty&ok=Valider HTTP/1.0
```

```
HTTP/1.1 200 OK
```

```
Date: Fri, 01 Sep 2017 08:49:24 GMT
```

```
Server: Apache/2.4.23 (Win64) PHP/7.0.10
```

```
X-Powered-By: PHP/7.0.10
```

```
Connection: close
```

```
Content-Type: text/html; charset=UTF-8
```

```
<!DOCTYPE html>
```

```
<html lang="fr">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Formulaire traitement à part</title>
```

```
...
```

Le protocole HTTP



- POST : envoi de données en vue d'un traitement

```
POST /CoursPHP/3-formulaires/traitementAPart/formulaireTraitement.php HTTP/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 83

nom=a&saisie[age]=14&saisie[email]=gege%40truc.com&saisie[mdp]=123AZErtY&ok=Valider
HTTP/1.1 200 OK
Date: Fri, 01 Sep 2017 09:00:15 GMT
Server: Apache/2.4.23 (Win64) PHP/7.0.10
X-Powered-By: PHP/7.0.10
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <title>Formulaire traitement à part</title>
  ...
```

Les échanges de données entre le client et le serveur

La fonction `header()`

- La fonction `header()` permet d'envoyer une en-tête HTTP
 - Exemple

```
<?php  
header("HTTP/1.0 404 Not Found");
```



Doit être appelée
avant l'envoi de
tout contenu

La redirection

- La redirection vers un chemin relatif

```
<?php  
header('Location: ../../index.php');
```

- La redirection vers un chemin absolu

```
<?php  
header('Location: /CoursPHP/index.php');
```

- La redirection vers un autre site web

```
<?php  
header('Location: http://www.eni-ecole.fr/');
```

Les échanges de données entre le client et le serveur

La redirection

- La redirection permanente
 - Permet aux moteurs de recherche de modifier l'adresse dans leurs bases de données

```
<?php  
header('Status: 308 Permanent Redirect', false, 308);  
header('Location: ../../index.php');
```

La redirection

- La redirection conditionnelle

```
<?php
$nom = filter_input(INPUT_POST, 'nom', FILTER_SANITIZE_STRING);
if(!$nom) {
    header('Location: saisieNom.php');
    exit();
}
?>
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <title>Les redirections</title>
    </head>
    <body>
        <h1>Bonjour <?= $nom ?></h1>
    </body>
</html>
```



Saisie du nom

Nom :



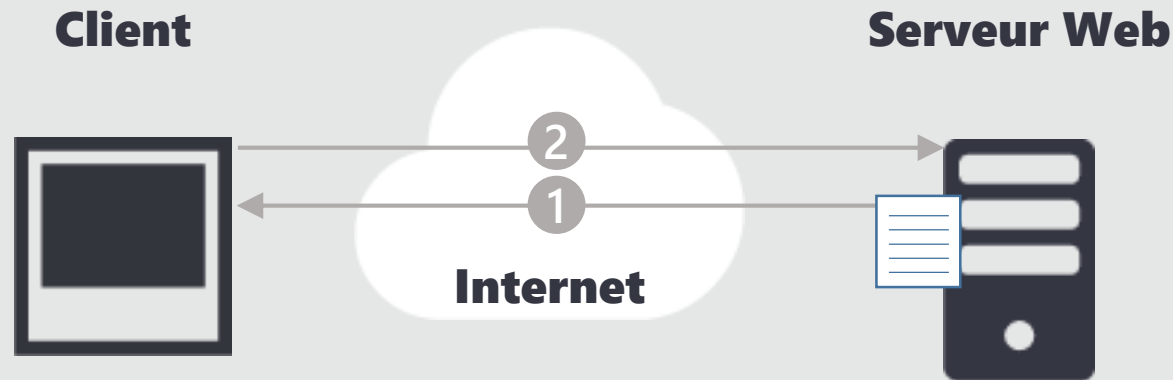
Bonjour Julie

Les échanges de données entre le client et le serveur

Les cookies

« Un cookie est défini par le protocole de communication HTTP comme étant une suite d'informations envoyée par un serveur HTTP à un client HTTP, que ce dernier retourne lors de chaque interrogation du même serveur HTTP sous certaines conditions. »

Source : Wikipédia



Les échanges de données entre le client et le serveur

Le dépôt et la suppression d'un cookie

- Déposer un cookie

- Pour la durée de la session

```
setcookie('nom_du_cookie', 'valeur du cookie')
```

- Pour une durée déterminée

```
setcookie('nom_du_cookie', 'valeur du cookie ', time() + (30 * 24 * 60 * 60))
```

30 jours

- Suppression d'un cookie

```
setcookie('nom_du_cookie')
```

La valeur vide entraîne la suppression du cookie

Consultation d'un cookie

- La variable superglobale `$_COOKIE`

```
<?php
if (isset($_COOKIE['nom_du_cookie']))
    echo '<p><code>$_COOKIE[\'nom_du_cookie\'] = </code>' .
        $_COOKIE['nom_du_cookie'] . '</p>';
else
    echo '<p>Pas de cookie "nom_du_cookie"</p>';
```

- La fonction `filter_input()`

```
$coo = filter_input(INPUT_COOKIE, 'nom_du_cookie', FILTER_SANITIZE_STRING);
if($coo)
    echo '<p><code>$coo = </code>' . $coo . '</p>';
else
    echo '<p>Pas de cookie "nom_du_cookie"</p>';
```

Les échanges de données entre le client et le serveur

Les cookies

Démonstration



Les échanges de données entre le client et le serveur

Les sessions

- Création d'une session

```
<?php
```

```
// Création de la session si elle n'a pas encore été créée ou récupération  
de celle existante
```

```
session_start();
```

- Génération d'un identifiant unique

- Exemple: **ri8v6lep61uvti7fqbu5tfsoh4**

Les échanges de données entre le client et le serveur

Les sessions - la variable superglobale \$_SESSION

- Stocker des données pour la session

```
<?php
// Création de la session si elle n'a pas encore été créée ou récupération de
celle existante
session_start();
// Enregistrer deux informations dans la session.
$_SESSION['nom'] = 'Shannon';
```

- Consulter des données stockées pour la session

```
<?php
session_start();
if (isset($_SESSION['nom']))
    echo '<p>$_SESSION[\'nom\'] = ' . $_SESSION['nom'] . '</p>';
else
    echo '<p>Pas de valeur pour "nom" dans la session</p>';
```

La fonction `filter_input()` ne peut être utilisée car même s'il existe une constante `INPUT_SESSION`, cela n'a pas encore été implémenté

Les échanges de données entre le client et le serveur

Les sessions

Démonstration



Les échanges de données entre le client et le serveur

AJAX

- Objectif : appel au serveur sans recharger toute la page
 - Un page PHP côté serveur pour répondre à la demande
 - Peut renvoyer les informations selon différents formats :
 - Texte
 - XML
 - JSON
 - ...
 - Un script JavaScript côté client
 - Pour envoyer la demande
 - Pour traiter la réponse à son retour

Les échanges de données entre le client et le serveur

AJAX

- La page PHP

```
<?php
$valeurs = [
    '67000' => ['Strasbourg'],
    '67100' => ['Strasbourg'],
    '67200' => ['Strasbourg'],
    '67110' => ['Dambach', 'Gumbrechtshoffen'],
    '67120' => ['Altorf', 'Avolsheim', 'Dachstein', 'Dorlisheim']];

$cp = filter_input(INPUT_GET, 'cp', FILTER_SANITIZE_NUMBER_INT);
if(isset($valeurs[$cp]))
    echo implode('|', $valeurs[$cp]);
```

Les échanges de données entre le client et le serveur

AJAX

- Le script JavaScript

```
window.onload = init;
var inputCP;

function init() {
    inputCP = document.getElementById("cp");
    inputCP.onblur = chargerVilles;
}

function chargerVilles() {
    // Création d'une instance permettant d'envoyer des requêtes au serveur
    var req = new XMLHttpRequest();
    // Création de la requête à envoyer
    req.open("GET", "GetVillesTxt.php?cp=" + inputCP.value, true);
    // Choix de la méthode appelée automatiquement lors de l'arrivée de la réponse
    req.onload = insererChoixVilles;
    // Envoi de la requête
    req.send();
}
```


Les échanges de données entre le client et le serveur

AJAX

```
function insererChoixVilles() {  
    if (this.status === 200) { // Si le code de statut est à Succès  
        // this.responseText permet d'obtenir le contenu retourné par le serveur  
        villes = this.responseText.split("|");  
        selectVille = document.getElementById("ville");  
        selectVille.innerHTML = "";  
        if (villes[0] === "") {  
            inputCP.title = "Ce code postal n'existe pas !";  
            inputCP.style.backgroundColor = "pink";  
        } else {  
            for (v of villes) {  
                optionVille = document.createElement("option");  
                optionVille.value = v;  
                optionVille.innerHTML = v;  
                selectVille.appendChild(optionVille);  
            }  
        }  
    }  
}
```

Les échanges de données entre le client et le serveur

AJAX

Démonstration

