

Sites dynamiques : HTTP et PHP

Sylvain Tenier, Romain Vallée, Vincent Derrien

Département TIC - Esigelec

Semestre S7 - 2016

Plan

1 De la page au site web : un mécanisme en 3 étapes

- Accès : Internet - un système client/serveur
- Récupération : le protocole HTTP
- Interprétation : le rôle du navigateur web

2 PHP : prétraitement et interactions

- Intégration de contenu dynamique
- Composition de pages à partir de fragments
- Interactions avec l'utilisateur

3 Validation

- Prévalidation en Javascript
- Validation des éléments transmis en PHP
- Protection contre les attaques XSS

Plan

- 1 De la page au site web : un mécanisme en 3 étapes
 - Accès : Internet - un système client/serveur
 - Récupération : le protocole HTTP
 - Interprétation : le rôle du navigateur web
- 2 PHP : prétraitement et interactions
 - Intégration de contenu dynamique
 - Composition de pages à partir de fragments
 - Interactions avec l'utilisateur
- 3 Validation
 - Prévalidation en Javascript
 - Validation des éléments transmis en PHP
 - Protection contre les attaques XSS

Accès à un terminal sur un réseau

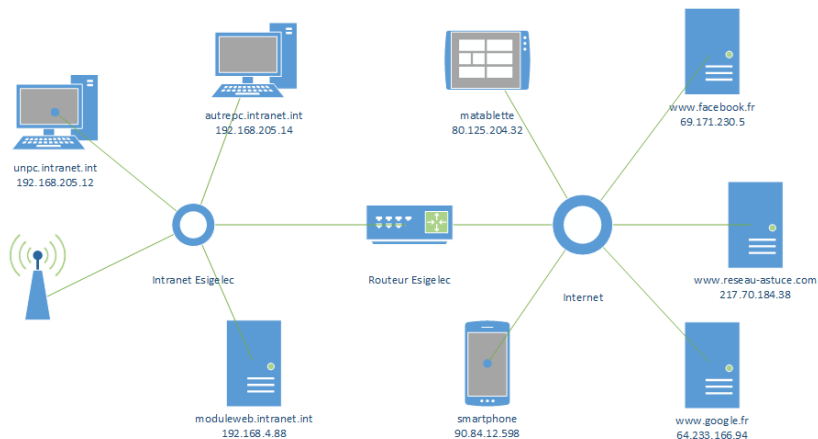


FIGURE : Terminals clients et serveurs sur réseaux Intranet et Internet

Adresse IP

- Chaque *terminal* connecté au réseau possède (au moins) une adresse IP
 - Forme : [0-255] . [0-255] . [0-255] . [0-255]
- Un terminal qui *initie* une connexion est un *client*
 - Son IP peut changer à chaque connexion à Internet
- Un terminal qui reçoit une connexion est un *serveur*
 - *internet* si son IP est publique
 - *intranet* s'il est situé dans le même réseau que le *client*
- Tout terminal peut se connecter à lui-même à l'adresse 127.0.0.1

Application 1 : sites Internet et Intranet

Opérations

- Lancez un navigateur
- Saisissez l'adresse
64.233.166.94
- Saisissez l'adresse
192.168.4.88

Depuis

- un PC
- votre téléphone
- Que constatez-vous ?

DNS : un annuaire décentralisé

- Domain Name System
- Associe une adresse FQDN (Fully Qualified Domain Name) à l'adresse IP d'un serveur
 - Le FQDN est composé du nom d'hôte suivi du nom de domaine
- Facilite la mémorisation
 - `www.google.fr` est associé à `64.233.166.94`
 - `moduleweb.intranet.int` est associé à `192.186.4.88`
 - `localhost` est associé à `127.0.0.1`
- Pour un serveur, l'accès est équivalent par l'adresse IP ou par le FQDN

Ports de connexion

- Un serveur peut fournir un ou plusieurs *services*
- Un *port* est associé à chaque service
- Les ports numérotés en dessous de 1024 sont réservés
 - serveur mail : port 25 pour l'envoi, 110 pour la réception POP
 - serveur SSH (administration à distance sécurisée) : port 22
 - serveur web : port 80 (ou > 1024)
 - serveur web sécurisé : port 443
- Les ports situés au delà sont disponibles
 - Par exemple, MAMP utilise le port 8888, Skype un port aléatoire > 1024

Accès : les points clés

- Un site web est situé sur un serveur connecté au réseau
 - ① Le réseau peut être *intranet* (même réseau que le client) ou *internet* (possédant une IP publique)
 - ② Un FQDN est généralement associé à l'adresse IP, permettant la mémorisation de l'adresse du site
 - ③ La connexion s'effectue par défaut sur le port 80 ou 443 (site sécurisé)

Plan

1 De la page au site web : un mécanisme en 3 étapes

- Accès : Internet - un système client/serveur
- Récupération : le protocole HTTP
- Interprétation : le rôle du navigateur web

2 PHP : prétraitement et interactions

- Intégration de contenu dynamique
- Composition de pages à partir de fragments
- Interactions avec l'utilisateur

3 Validation

- Prévalidation en Javascript
- Validation des éléments transmis en PHP
- Protection contre les attaques XSS

Les 3 étapes de la récupération

- ❶ Le client se connecte au serveur sur un port spécifique
 - 80 par défaut pour une transmission en clair
 - 443 pour un serveur sécurisé
 - personnalisé (>1024), par exemple pour un serveur de test
- ❷ Le client envoie une *requête HTTP*
- ❸ Le serveur renvoie une *réponse HTTP*

HTTP : un système de questions/réponses

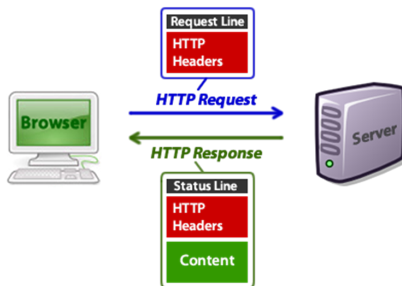
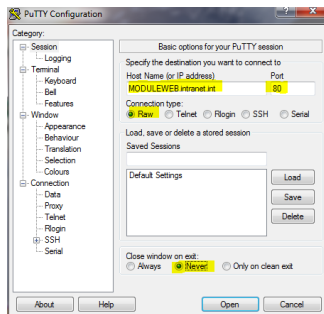


FIGURE : Accès, requête et réponse

Application 2 : interaction avec un serveur HTTP

Partie 1 : connexion au serveur

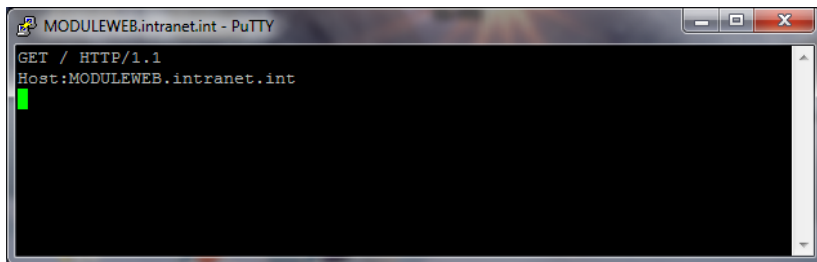
- Lancez l'application Putty
- Connectez-vous à *moduleweb.intranet.int* sur le port 80



Application 2 : interaction avec un serveur HTTP

Partie 2 : transaction HTTP

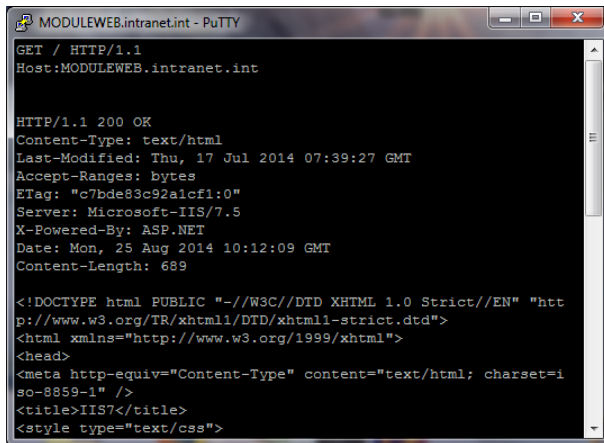
- Entrez les commandes suivantes dans le terminal de Putty



```
MODULEWEB.intranet.int - PuTTY
GET / HTTP/1.1
Host:MODULEWEB.intranet.int
█
```

- Appuyez deux fois sur *Entrée*. Qu'observez-vous ?

Les 3 parties d'une transaction HTTP



```
MODULEWEB.intranet.int - PuTTY
GET / HTTP/1.1
Host:MODULEWEB.intranet.int

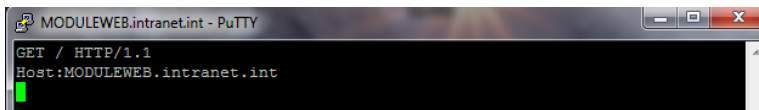
HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Thu, 17 Jul 2014 07:39:27 GMT
Accept-Ranges: bytes
ETag: "c7bde83c92a1cf1:0"
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Mon, 25 Aug 2014 10:12:09 GMT
Content-Length: 689

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>IIS7</title>
<style type="text/css">
```

FIGURE : Requête, en-tête et corps de réponse

Requête HTTP : composants obligatoires

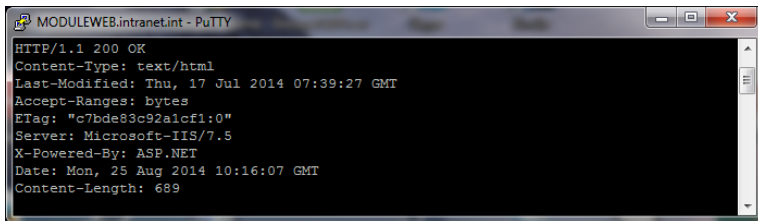
- La première ligne indique la méthode, la ressource à récupérer et la version du protocole
 - 1 Méthodes courantes : GET, POST, HEAD
 - 2 Ressource : chemin en notation Unix
 - / récupère la page d'accueil ou le contenu du dossier racine
 - /lapage.html récupère la ressource (fichier) lapage.html
 - 3 Version : HTTP/1.1 principalement
- La seconde ligne indique le FQDN de l' hôte



A screenshot of a PuTTY terminal window titled "MODULEWEB.intranet.int - PuTTY". The terminal displays an HTTP request: "GET / HTTP/1.1" on the first line and "Host:MODULEWEB.intranet.int" on the second line. A green cursor is visible at the end of the second line.

En-têtes “Headers” de la réponse HTTP

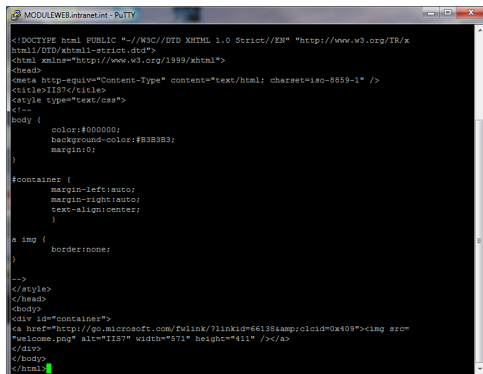
- La première ligne est la *ligne de statut*
 - 1 Version du protocole (HTTP/1.1)
 - 2 Code de statut
 - 2xx indique une requête réussie
 - 3xx indique que la requête doit être redirigée
 - 4xx indique une erreur dans la requête
 - 5xx indique une erreur du serveur
- Les autres lignes sont relatives au serveur ou à la requête



```
MODULEWEB.intranet.int - PuTTY
HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Thu, 17 Jul 2014 07:39:27 GMT
Accept-Ranges: bytes
ETag: "c7bde83c92a1cf1:0"
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Mon, 25 Aug 2014 10:16:07 GMT
Content-Length: 689
```

Corps "Body" de la réponse HTTP

- Le body est présent si le code de statut est 200
- Le type de contenu est indiqué par l'en-tête Content-type
 - Dans le cas d'une page HTML on a text/html



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>IIS7</title>
<style type="text/css">
<!--
body {
    color:#000000;
    background-color:#B3B3B3;
    margin:0;
}

#container {
    margin-left:auto;
    margin-right:auto;
    text-align:center;
}

a img {
    border:none;
}

-->
</style>
</head>
<body>
<div id="container">
<a href="http://go.microsoft.com/fwlink/?linkid=66138&cid=0x409"><img src=
"welcome.png" alt="IIS7" width="571" height="411" /></a>
</div>
</body>
</html>
```

Récupération : les points clés

- La récupération d'une ressource s'effectue en 3 étapes
 - ❶ Connexion au serveur hôte en spécifiant le FQDN et le port
 - ❷ Envoi d'une requête HTTP contenant la méthode, le chemin vers la ressource, la version du protocole et le FQDN de l'hôte
 - ❸ Réception d'une réponse HTTP contenant
 - des en-têtes, incluant notamment un code de statut
 - un corps dont le contenu correspond à l'en-tête Content-type si le code de statut est 200
- Référence : <http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html>

Plan

1 De la page au site web : un mécanisme en 3 étapes

- Accès : Internet - un système client/serveur
- Récupération : le protocole HTTP
- Interprétation : le rôle du navigateur web

2 PHP : prétraitement et interactions

- Intégration de contenu dynamique
- Composition de pages à partir de fragments
- Interactions avec l'utilisateur

3 Validation

- Prévalidation en Javascript
- Validation des éléments transmis en PHP
- Protection contre les attaques XSS

Navigateurs web et HTTP

- Un navigateur web est un *Client HTTP*
 - Il s'agit d'un logiciel qui automatise la connexion au serveur et la récupération des pages web par HTTP
 - Il existe d'autres types de clients HTTP, comme les robots chargés d'indexer le Web ou de récupérer de l'information automatiquement
- En entrée, l'utilisateur fournit un *URI*
 - Exemple : `http://www.rfc-base.org/rfc-3987.html`
 - Cet URI est constitué de 3 parties
 - ① Le protocole `http://`
 - ② Le FQDN de l'hôte `www.rfc-base.org`
 - ③ La ressource à récupérer `rfc-3986.html`

De l'URI à la requête

- Le navigateur utilise l'URI pour :
 - 1 se connecter à l'hôte (sur le port 80 si le port n'est pas précisé),
 - 2 générer la requête HTTP.
- Exemples (sur fond bleu l'URI, sur fond gris la requête)

```
http://www.rfc-base.org/rfc-3986.html
```

```
GET /rfc-3986.html HTTP/1.1
```

```
Host: www.rfc-base.org
```

```
http://localhost:8080/unepage.html
```

```
GET /unepage.html HTTP/1.1
```

```
Host: localhost:8080
```

Application 3 : inspection de requête

- Instructions
 - 1 Démarrez google chrome
 - 2 Faites un clic droit "inspecter l'élément" puis cliquez sur "Network"
 - 3 Tapez l'adresse `www.esigelec.fr` dans la barre d'adresse
- Que constatez-vous ?

Une page web est constituée de plusieurs ressources

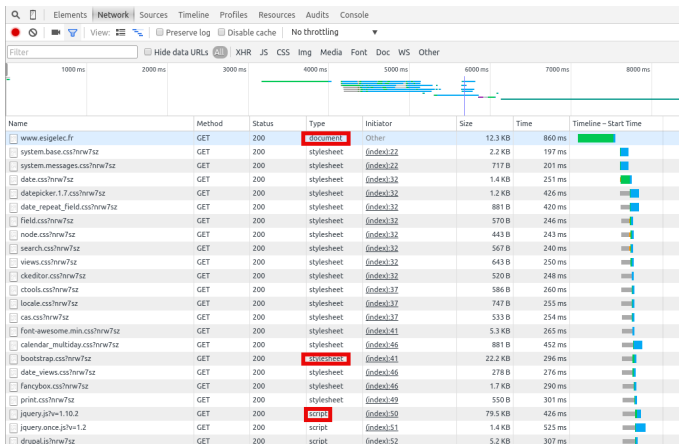


FIGURE : Ensemble des requêtes effectuées par le navigateur

Types de ressources composant une page web (rappel)

- Une page web est composée à partir de 3 langages

HTML (HyperText Markup Language)

Définit la *structure* de la page sous la forme de balises autour du contenu

CSS (Cascading StyleSheets)

Définit la *présentation* de la page (positionnement, couleurs, polices, ...)

Javascript

Permet de *manipuler* la page une fois chargée (animations, gestion d'événements, modifications, ...)

- L'accès à une URI donnée génère une requête HTTP par le navigateur web pour chaque ressource à récupérer composant la page web identifiée par l'URI

Récupération par HTTP d'une page web

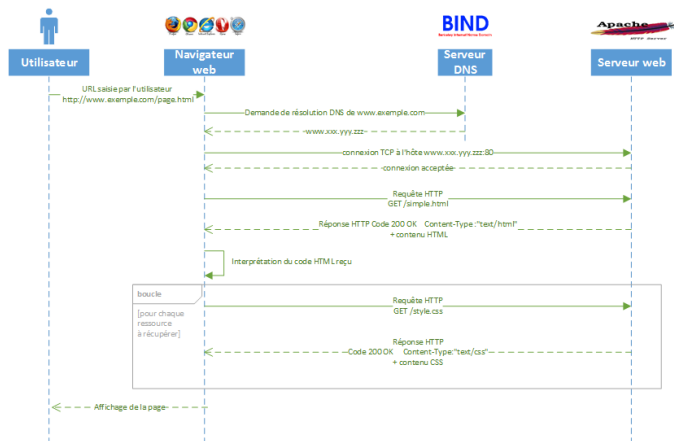


FIGURE : Requêtes et réponses HTTP

Autres types de ressources

- HTTP permet de récupérer d'autres types de ressources que des pages web et leurs composants
- Exemple d'un document PDF

```
http://www.esigelec.fr/sites/default/files/documents/  
studentguide-2015_web.pdf
```

Requête

```
1 GET /sites/default/files/documents/studentguide-2015_web.pdf  
  HTTP/1.1  
2   Host: www.esigelec.fr  
3
```

En-tête de la réponse

```
1 HTTP/1.1 200 OK  
2 Content-Type: application/pdf  
3
```

Plan

- 1 De la page au site web : un mécanisme en 3 étapes
 - Accès : Internet - un système client/serveur
 - Récupération : le protocole HTTP
 - Interprétation : le rôle du navigateur web
- 2 PHP : prétraitement et interactions
 - Intégration de contenu dynamique
 - Composition de pages à partir de fragments
 - Interactions avec l'utilisateur
- 3 Validation
 - Prévalidation en Javascript
 - Validation des éléments transmis en PHP
 - Protection contre les attaques XSS

Plan

- 1 De la page au site web : un mécanisme en 3 étapes
 - Accès : Internet - un système client/serveur
 - Récupération : le protocole HTTP
 - Interprétation : le rôle du navigateur web
- 2 PHP : prétraitement et interactions
 - Intégration de contenu dynamique
 - Composition de pages à partir de fragments
 - Interactions avec l'utilisateur
- 3 Validation
 - Prévalidation en Javascript
 - Validation des éléments transmis en PHP
 - Protection contre les attaques XSS

Exemple : intégration d'éléments dynamiques

Version de PHP et heure courante

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title> Page générée avec
6     <?php echo 'php'. phpversion(); ?>
7   </title>
8 </head>
9 <body>
10  <?php echo '<p> Bienvenue : il
11    est ' . date("H:i") . '</p>'; ?>
12  </body>
13 </html>
14
```

Points clés

- extension de fichier .php
- balises <?php et ?>
- un code PHP n'est *pas interprétable* par le navigateur web
- tout code HTML est un code PHP valide

Séquence de traitement d'une ressource PHP

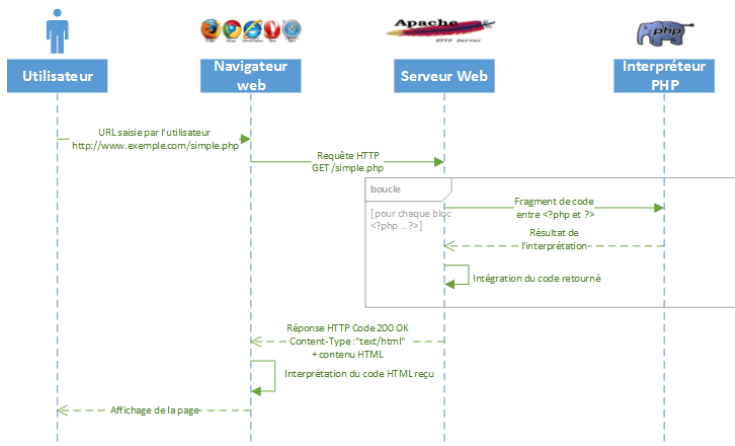


FIGURE : Prétraitement de la page par l'interpréteur PHP

Le navigateur n'interprète pas PHP



Rappel

- Le navigateur web est capable d'interpréter HTML, CSS et Javascript
- Le serveur *transforme* le PHP en HTML
- Le navigateur web *ne reçoit jamais* de PHP

Séquence de traitement d'une ressource PHP (2)

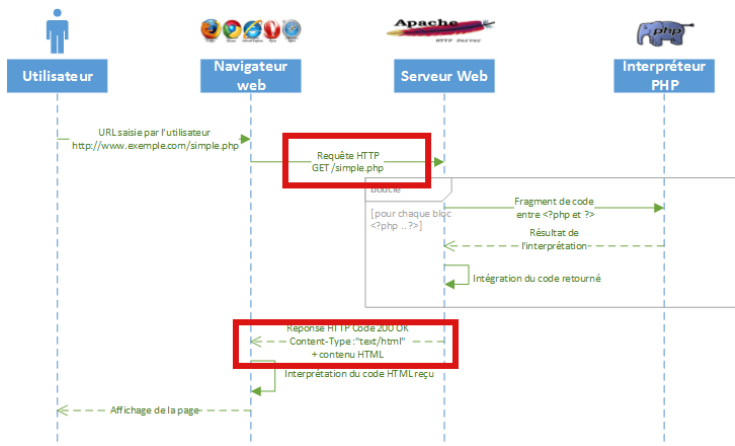


FIGURE : Prétraitement de la page par l'interpréteur PHP

Du PHP au HTML

Contenu de la ressource PHP réclamée par la requête HTTP

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title> Page générée avec
6     <?php echo 'php '. phpversion();
7     ?>
8   </title>
9 </head>
10 <body>
11 <?php echo '<p>Bienvenue : il
12   est ' . date("H:i") . '</p>'; ?>
13 </body>
14 </html>
```

Contenu HTML de la réponse HTTP envoyée au navigateur

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title> Page générée avec php
6     5.5.8</title>
7   </head>
8   <body>
9     <p> Bienvenue : il est 11:15</p>
10  </body>
11 </html>
```

Plan

- 1 De la page au site web : un mécanisme en 3 étapes
 - Accès : Internet - un système client/serveur
 - Récupération : le protocole HTTP
 - Interprétation : le rôle du navigateur web
- 2 PHP : prétraitement et interactions
 - Intégration de contenu dynamique
 - Composition de pages à partir de fragments
 - Interactions avec l'utilisateur
- 3 Validation
 - Prévalidation en Javascript
 - Validation des éléments transmis en PHP
 - Protection contre les attaques XSS

Squelette de base d'une page HTML

Exemple avec les balises sémantiques HTML 5

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title> Squelette de page avec balises sémantiques</title>
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body>
9     <div id="wrapper">
10      <header> Bannière ou en-tête du contenu</header>
11      <nav><ul><li> Menu de navigation</li></ul></nav>
12      <section id="content">
13        <article> Premier article</article>
14      </section>
15      <aside> Barre sur le côté droit</aside>
16      <footer> Pied de page du contenu</footer>
17    </div>
18  </body>
19 </html>
20
```

Squelette dynamique en PHP

Les éléments communs sont importés depuis des fichiers `.inc.php`

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title> Squelette de page dynamique</title>
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body>
9     <div id="wrapper">
10       <?php include('header.inc.php'); ?>
11       <?php include('nav.inc.php'); ?>
12       <section id="content">
13         <article> Premier article</article>
14       </section>
15       <?php include('cotedroit.inc.php'); ?>
16       <?php include('footer.inc.php'); ?>
17     </div>
18   </body>
19 </html>
```

header.inc.php

```
1 <header> Bannière ou
   en-tête du
   contenu</header>
```

nav.inc.php

```
1 <nav><ul><li> Menu de
   navigation</li></
   ul></nav>
```

...

Plan

- 1 De la page au site web : un mécanisme en 3 étapes
 - Accès : Internet - un système client/serveur
 - Récupération : le protocole HTTP
 - Interprétation : le rôle du navigateur web
- 2 PHP : prétraitement et interactions
 - Intégration de contenu dynamique
 - Composition de pages à partir de fragments
 - Interactions avec l'utilisateur
- 3 Validation
 - Prévalidation en Javascript
 - Validation des éléments transmis en PHP
 - Protection contre les attaques XSS

Demande ou envoi ?



Recherche Google

J'ai de la chance

Step 1: Your details

Name

First and last name

Email

example@domain.com

Phone

Eg. +447500000000

Step 2: Delivery address

Address

Post code

Country

Step 3: Card details

Choix de la méthode HTTP

Méthode GET

- *Demande* de documents
 - Recherche, filtrage, ...
- Passe par l'URL de la page
- Ne doit pas modifier le contenu

Méthode POST

- *Envoi* de données à traiter
 - Inscription, connexion, participation, ...
- Encodage des données dans le corps de la requête HTTP
- Peut modifier la base de données

Formulaire minimal

Formulaire avec les attributs sémantiques HTML 5

Votre nom Votre Email
☒ Oui, envoyez-moi du spam !

```
1 <!DOCTYPE html>
2 <html>
3   <head><meta charset="utf-8"><title> Formulaire minimal</title></head>
4   <body>
5     <form method="post" action="spamme.php">
6       <label for="nom">Votre nom</label>
7       <input type="text" id="nom" name="nom" placeholder="Votre
nom..." required>
8       <label for="mail"> Votre Email</label>
9       <input type="email" id="mail" name="mail" placeholder="xxx@yyy.zzz
" required><br>
10      <input type="checkbox" id="check" name="check" value="spamok"
checked>
11      <label for="check"> Oui, envoyez-moi du spam !</label>
12      <button name="envoi" value="ok">Spammez-moi !</button>
13      <input type="reset" value="Effacer">
14    </form>
15  </body>
16 </html>
```

Attributs remarquables

Attributs de l'élément `form`

`method=["get"|"post"]` méthode de requête HTTP

`action=[path|uri]` adresse du script de traitement du formulaire

Attributs remarquables de l'élément `input`

`type=[text|password|checkbox|radio|submit|reset|
file|hidden|image|button|...]` choix du type de contrôle

`name=[chaine]` nom de la variable à traiter par le script

`value=[chaine]` valeur de la variable envoyée au script

Contrôles sémantiques HTML 5

- HTML 5 introduit de nouveaux contrôles
- Valeurs possibles de l'attribut `type` de l'élément `input` :
 - `email` le champ requiert un contenu au format d'adresse électronique
 - `url` le champ accueille des URL valides
 - `tel` le champ est destiné aux numéros de téléphone
 - `number` le champ accepte uniquement les caractères numériques
 - `color` le champ est prévu pour les chaînes représentant une valeur de couleur
- Le navigateur peut afficher des interfaces spécifiques et des validations préventives
- Si un contrôle n'est pas reconnu, il est considéré comme `text`

Exploitation des contrôles sémantiques

Exemple du contrôle `input="email"`

Personnalisation clavier Android



Validation de formulaire

Une idée ? Un projet ?

Votre nom

Votre e-mail

Votre message

ENVOYER

Application 4 : envoi de formulaire avec la méthode POST

Actions

- ➊ Récupérez le formulaire minimal sur ENT et déployez le dans EasyPHP
- ➋ Affichez l'inspecteur d'éléments, onglet Network
- ➌ Remplissez et validez le formulaire en
 - ➊ laissant les champs vides
 - ➋ saisissant des chaînes de caractères aléatoires
 - ➌ remplissant correctement les champs

Questions

- ➊ À quel niveau se passe la détection de validité ?
- ➋ Comment sont transmises les informations saisies ?
- ➌ Que contient la réponse HTTP ?
 - Comment corriger le problème ?

Envoi de données par POST

← → ↻ 127.0.0.1/projects/Seance3/app11/spamme.php ☆

Objet non trouvé!

L'URL demandée n'a pas pu être trouvée sur ce serveur. La référence sur [la page citée](#) semble être erronée ou perimée. Nous vous prions d'informer l'auteur de [cette page](#) de cette erreur.

Si vous pensez qu'il s'agit d'une erreur du serveur, veuillez contacter le [webmestre](#).

Error 404

The screenshot shows a web browser with a 404 error message. Below the error message, the Network tab is open, showing details for a POST request to `127.0.0.1/projects/Seance3/app11/spamme.php`. The request status is 404 Not Found. The request headers include `Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8`, `Accept-Encoding: gzip, deflate`, `Accept-Language: fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4`, `Cache-Control: max-age=0`, `Connection: keep-alive`, `Content-Length: 52`, `Content-Type: application/x-www-form-urlencoded`, `Host: 127.0.0.1`, `Origin: http://127.0.0.1`, and `Referer: http://127.0.0.1/projects/Seance3/app11/formminimal.php`. The Form Data section shows the following data: `nom: Joe`, `mail: joe@example.com`, and `check: spamok`.

FIGURE : Encodage des données dans une requête POST

Traitement de formulaire en PHP

Exemple de traitement du formulaire minimal

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Traitement</title>
6   </head>
7   <body>
8     <?php echo ' <p>Bonjour
9       '. $_POST['nom'] .
10      ' <br> Merci de nous avoir
11      transmis votre adresse '
12      . $_POST['mail'] . ' </p>';
13     ?>
14   </body>
15 </html>
```

Points clés

- Une variable php débute par un \$
- Les éléments du formulaire sont récupérés dans un tableau associatif \$_POST
- Les clés d'accès du tableau correspondent aux attributs `name=` du formulaire

Application 5 : traitement de formulaire

Actions

- ❶ Copiez le code de traitement dans un fichier *spamme.php*
- ❷ Chargez la page du formulaire dans le navigateur
- ❸ Remplissez et validez le formulaire
- ❹ Modifiez le formulaire et le script pour utiliser la méthode GET

Questions

- ❶ Quelle différence constatez-vous dans la requête HTTP ?
- ❷ Quelle méthode faut-il privilégier ?

Plan

- 1 De la page au site web : un mécanisme en 3 étapes
 - Accès : Internet - un système client/serveur
 - Récupération : le protocole HTTP
 - Interprétation : le rôle du navigateur web
- 2 PHP : prétraitement et interactions
 - Intégration de contenu dynamique
 - Composition de pages à partir de fragments
 - Interactions avec l'utilisateur
- 3 Validation
 - Prévalidation en Javascript
 - Validation des éléments transmis en PHP
 - Protection contre les attaques XSS

Plan

- 1 De la page au site web : un mécanisme en 3 étapes
 - Accès : Internet - un système client/serveur
 - Récupération : le protocole HTTP
 - Interprétation : le rôle du navigateur web
- 2 PHP : prétraitement et interactions
 - Intégration de contenu dynamique
 - Composition de pages à partir de fragments
 - Interactions avec l'utilisateur
- 3 Validation
 - Prévalidation en Javascript
 - Validation des éléments transmis en PHP
 - Protection contre les attaques XSS

Validation intégrée au navigateur

- Évite de transmettre un formulaire manifestement erroné
 - Gain de temps et de ressources
- N'est *PAS* un élément de sécurisation du site
 - Tout ce qui est transmis peut être modifié par l'utilisateur
- Exemples
 - ➊ Attribut `required` de l'élément `input`
`<input type="text" ... required>`
 - ➋ Contrôles sémantiques HTML 5 validés par le navigateur
`<input type="email" ...>`

Validation avec Javascript

Principes

- Interception du clic sur le `submit` pour empêcher l'envoi
- Validation de chaque élément du formulaire
 - Si correct, transmission du formulaire
 - Sinon, affichage d'un message d'erreur
- Le plugin jQuery Validation Plugin permet de simplifier la validation
- La validation en Javascript n'est *PAS* un élément de sécurité

Plan

- 1 De la page au site web : un mécanisme en 3 étapes
 - Accès : Internet - un système client/serveur
 - Récupération : le protocole HTTP
 - Interprétation : le rôle du navigateur web
- 2 PHP : prétraitement et interactions
 - Intégration de contenu dynamique
 - Composition de pages à partir de fragments
 - Interactions avec l'utilisateur
- 3 Validation
 - Prévalidation en Javascript
 - Validation des éléments transmis en PHP
 - Protection contre les attaques XSS

Validation du contenu par fonctions PHP

- Toute donnée fournie par l'utilisateur doit être contrôlée côté serveur
- PHP fournit des fonctions de gestion des variables
 - `is_null($var)` : \$var est vide
 - `is_numeric($var)` : \$var est numérique
 - `is_string($var)` : \$var est une chaîne de caractères
 - `isset($var)` : \$var est définie
- Exemple de validation d'un champ `age`

```
1 <?php>
2     if (!is_numeric($_POST['age'])) {
3         $_SESSION['msg'] = 'Veuillez corriger le champ âge';
4         header('Location: ' . $_SERVER['HTTP_REFERER']);
5     }
6     ?>
7
```

Validation du contenu par filtres PHP

Utilisation de la fonction `filter_var`

- renvoie FALSE si la validation échoue
- 6 filtres de validation peuvent être passés en paramètre :
 - 1 `FILTER_VALIDATE_BOOLEAN`
 - 2 `FILTER_VALIDATE_EMAIL`
 - 3 `FILTER_VALIDATE_FLOAT`
 - 4 `FILTER_VALIDATE_INT`
 - 5 `FILTER_VALIDATE_REGEXP`
 - 6 `FILTER_VALIDATE_URL`

- Exemple de validation d'email par filtre

```
1 <?php
2     if (!filter_var($_POST['mail'],FILTER_VALIDATE_EMAIL)){
3         $_SESSION['msg']='Veuillez corriger l'email';
4         header('Location: ' . $_SERVER['HTTP_REFERER']);
5     }?>
6
```

Plan

- 1 De la page au site web : un mécanisme en 3 étapes
 - Accès : Internet - un système client/serveur
 - Récupération : le protocole HTTP
 - Interprétation : le rôle du navigateur web
- 2 PHP : prétraitement et interactions
 - Intégration de contenu dynamique
 - Composition de pages à partir de fragments
 - Interactions avec l'utilisateur
- 3 Validation
 - Prévalidation en Javascript
 - Validation des éléments transmis en PHP
 - Protection contre les attaques XSS

Application 6 : exemple d'attaque XSS

Cross Site Scripting

Actions

- 1 Charger la page de formulaire dans Internet Explorer
- 2 Remplir le formulaire de la manière suivante :
Dans le champ email saisir un email valide
Dans le champ nom saisir le code suivant :
`</p><script>alert('XSS!!');</script><p style="color:red;">`

- Que constatez-vous ?

Empêcher les attaques XSS

- Protéger l'affichage avec html_entities

- Exemple

```
1 <?php echo ' <p>Bonjour'. htmlentities($_POST['nom'])  
2 . ' <br> Merci de nous avoir transmis votre adresse '  
3 . htmlentities($_POST['mail']) . ' </p>';  
4 ?>  
5
```

- Appliquer un filtre de *nettoyage* avec `filter_var`

- `https:`

`//php.net/manual/fr/filter.filters.sanitize.php`

Références

Formulaires HTML <http://www.w3.org/TR/2011/WD-html5-20110525/forms.html>

PHP <http://php.net/manual/fr/index.php>

Vulnérabilités XSS <http://www.cert.ssi.gouv.fr/site/CERTA-2002-INF-001/>

XSS et PHP http://talks.php.net/show/fossin_xss/0