

SIN6U5 : Développement web 2
(Prog. web coté serveur)
Sites statiques

Luigi Santocanale
LIF, Aix-Marseille Université

Transparents basés sur le cours de Bertrand Estellon

24 janvier 2018

Plan

Objectifs

Protocole et serveur HTTP

Génération des pages

Utilisation de PHP

La variable `$_GET`

Organisation avec un point d'entrée unique

Plan

Objectifs

Protocole et serveur HTTP

Génération des pages

Utilisation de PHP

La variable `$_GET`

Organisation avec un point d'entrée unique

BOB

CV À PROPOS BLOG CONTACT



Bonjour, je m'appelle Bob.

Je suis une personne imaginaire utilisée pour illustrer le cours de langages du Web.

MON ADRESSE

Le cours de langage du Web
Marseille, France.

MES LIENS

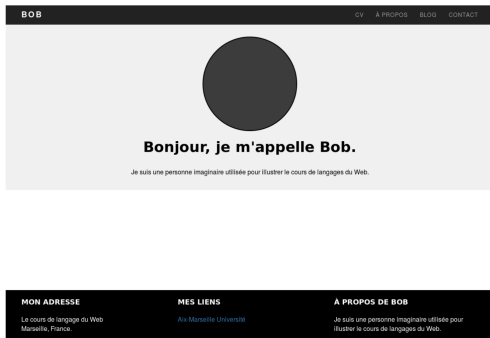
[Aix-Marseille Université](#)

À PROPOS DE BOB

Je suis une personne imaginaire utilisée pour
illustrer le cours de langages du Web.

Fonctionnalités

- Plusieurs pages
- Un menu en haut des pages
- Une bannière en bas des pages
- Une présentation homogène
- **Facilité de mise à jour**



Besoins et choix techniques

- Fournir le contenu des pages à afficher
HTML5
- Décrire le style des pages
CSS et Bootstrap
- Afficher correctement les pages sur le client
Navigateur Web
- Rendre les pages accessibles aux clients
Le réseau Internet, le protocole HTTP, un serveur HTTP

Plan

Objectifs

Protocole et serveur HTTP

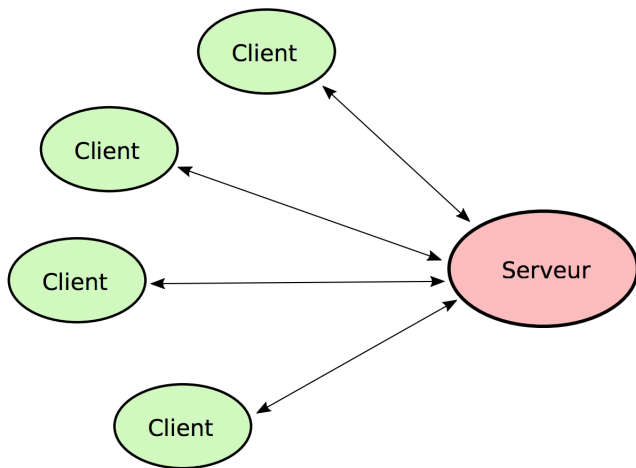
Génération des pages

Utilisation de PHP

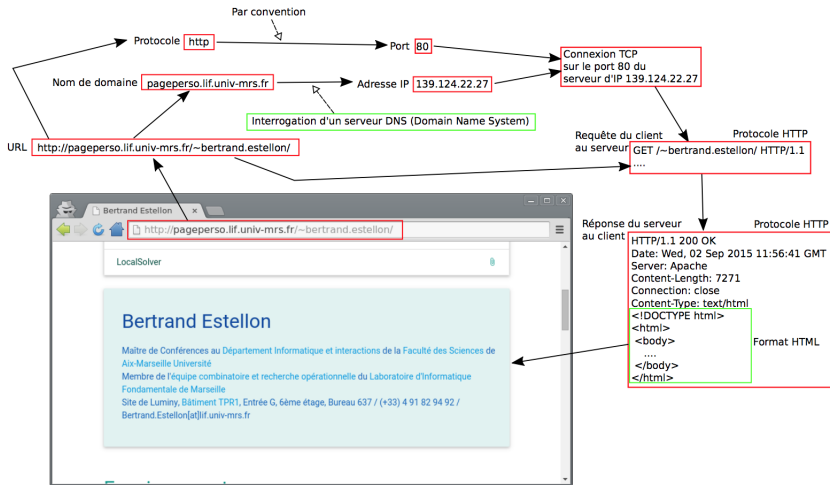
La variable `$_GET`

Organisation avec un point d'entrée unique

Architecture Client/Serveur



Affichage d'une page à partir d'une URL



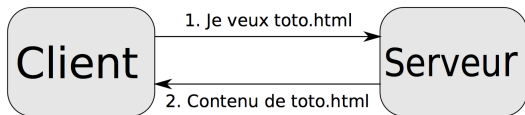
Une requête HTTP par élément à télécharger

The screenshot shows a Chrome browser window with the address bar at `pageperso.lif.univ-mrs.fr/~luigi.santocanale/`. The page content on the left shows a profile for Luigi Santocanale. The Network tab is open, displaying a waterfall chart and a table of requests.

Name	Status	Type	Initiator	Size	Time	Waterfall
~luigi.santocanale/	200	docum...	Other	5.7 KB	588 ms	
ls.css	200	stylesh...	(index)	(from d...)	4 ms	
jquery.min.js	200	script	(index)	(from d...)	6 ms	
fr.gif	200	gif	(index)	(from d...)	5 ms	
en.gif	200	gif	(index)	(from d...)	2 ms	
logoLFIarge.jpg	200	jpeg	(index)	(from d...)	50 ms	
logoAMU.jpg	200	jpeg	(index)	(from d...)	46 ms	
luigis.png	200	png	(index)	(from d...)	46 ms	
lif.ico	302	text/html	Other	208 B	336 ms	
error_404	200	text/html	lif.ico	8.3 KB	363 ms	
lif.ico	302	text/html	Other	208 B	373 ms	
error_404	200	text/html	lif.ico	7.7 KB	271 ms	

12 requests | 22.1 KB transferred | Finish: 2.02 s | DOMContentLoaded: 656 ms | Load: 665 ms

Protocole HTTP (Hypertext Transfer Protocol)



Requête :

GET /toto.html HTTP/1.0

Host : example.com

Referer : http://example2.com/

User-Agent : Mozilla/5.0 (X11 ;
Linux x86_64)

AppleWebKit/537.36 (KHTML,
like Gecko)

Chrome/45.0.2454.85

Safari/537.36

Réponse :

HTTP/1.1 200 OK

Date : Wed, 02 Sep 2015
12 :53 :36 GMT

Server : Apache

Content-Type : text/html

Content-Length : 59

Last-modified : Wed, 02 Sep
2015 10 :33 :36 GMT

<TITLE>Exemple</TITLE>

<P>page d'exemple.</P>

Serveur HTTP

Un serveur HTTP (ou serveur Web) :

- est un logiciel servant des requêtes respectant le protocole HTTP
- attend par défaut des connexions sur le port 80 (443 pour HTTPS)

Quelques serveurs HTTP connus :

- Apache HTTP Server
- IIS de Microsoft
- nginx
- GWS de Google
- lighttpd

Plan

Objectifs

Protocole et serveur HTTP

Génération des pages

Utilisation de PHP

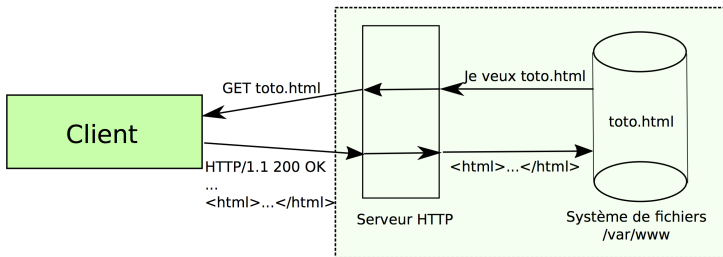
La variable `$_GET`

Organisation avec un point d'entrée unique

Pages web statiques

Le contenu d'une page web statique :

- ne varie pas en fonction des caractéristiques de la demande
- correspond souvent au contenu d'un fichier présent sur le serveur



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Ma page personnelle</title>
    <link href="css/bootstrap.css" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">
  </head>
  <body>
    <!-- menu --><?php include 'menu.html.php' ?>
    <!-- corps de la page index.html --><?php include
→ 'corps.html.php' ?>
    <!-- pied de page --><?php include 'footer.html.php' ?>
  </body>
</html>
```

Menu

```
<div class="navbar navbar-inverse navbar-static-top">
  <div class="container">
    <div class="navbar-header">
      <a class="navbar-brand" href="index.html">BOB</a>
    </div>
    <div class="navbar-collapse collapse">
      <ul class="nav navbar-nav navbar-right">
        <li><a href="cv.html">CV</a></li>
        <li><a href="apropos.html">À propos</a></li>
        <li><a href="blog.html">Blog</a></li>
        <li><a href="contact.html">Contact</a></li>
      </ul>
    </div>
  </div>
</div>
```



```
<div id="hello">
  <div class="container">
    <div class="row">
      <div class="col-lg-8 col-lg-offset-2 centered">
        
        <h1>Bonjour, je m'appelle Bob.</h1>
        <p>Je suis une personne imaginaire...</p>
      </div>
    </div>
  </div>
</div>
<div id="emptybody" class="container"></div>
```

Pied de page

```
<div id="footer">
  <div class="container">
    <div class="row">
      <div class="col-lg-4">
        <h4>Mon adresse</h4>
        <p>...</p>
      </div>
      <div class="col-lg-4">
        <h4>Mes Liens</h4>
        <p><a href="#">Aix-Marseille Université</a></p>
      </div>
      ...
    </div>
  </div>
</div><!-- footer -->
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Ma page personnelle -- CV</title>
    <link href="css/bootstrap.css" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">
  </head>
  <body>
    <!-- menu -->
    <!-- corps de la page cv.html -->
    <!-- pied de page -->
  </body>
</html>
```

Maintenabilité du code

Que doit-on faire pour ajouter une nouvelle page ?

- Créer un fichier et copier le menu et le pied de page
- Écrire le contenu de la page
- Mettre à jour le menu dans toutes les pages
- Faire en sorte que le titre de la page soit cohérent avec le menu

Problèmes de conception :

- Code dupliqué (exemple : si je modifie un morceau du pied de page, je dois modifier tous le pieds de page)
- L'ajout d'une page entraîne la modification de toutes les pages

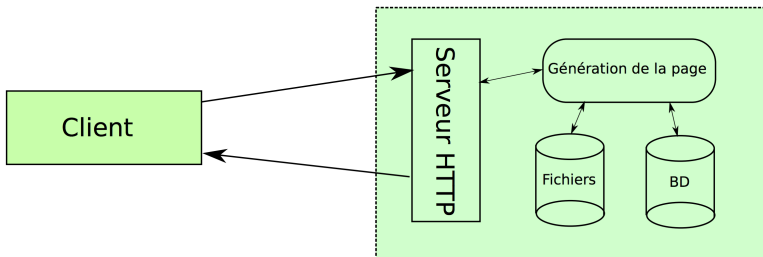
Conséquences :

- Ajouter une page demande des efforts importants
- Le risque d'ajouter un bug dans le site est important

Pages web dynamiques

Le contenu d'une page web dynamique :

- varie en fonction des caractéristiques de la demande
- est généré par le serveur



Plan

Objectifs

Protocole et serveur HTTP

Génération des pages

Utilisation de PHP

La variable `$_GET`

Organisation avec un point d'entrée unique

Inclusion de fichiers

contact.php

```
<?php
// contact.php
include 'header.inc.php';
include 'menu.inc.php';
include 'contact.inc.php';
include 'footer.inc.php';
```

header.inc.php

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Contact</title>
    <link href="..." rel="...">
  </head>
  <body>
```

footer.inc.php

```
  <div id="footer">
    <div class="container">
      <div class="row"> ...
    </div>
  </div>
</body>
</html>
```

contact.inc.php :

```
<div class="container"> ....
</div>
```

Les variables PHP

Quel est le titre de la page `cv.php` suivante ?

```
<?php
// contact.php
include 'header.inc.php';
include 'menu.inc.php';
include 'contact.inc.php';
include 'footer.inc.php';
```

Problématique : Comment factoriser le code de l'en-tête sans perdre la possibilité de changer le titre de la page ?

Solution : Utiliser une variable

Passage du titre via une variable

Le fichier `header.inc.php` :

```
<title><?=$title?></title>
```

Affectation de la variable `$title` avant d'inclure le fichier `header.inc.php` dans le fichier `cv.php` :

```
<?php
// cv.php
$title='Mon CV';
include 'header.inc.php';
include 'menu.inc.php';
include 'cv.inc.php';
include 'footer.inc.php';
```

Passage du titre via une variable

Le code du fichier `header.inc.php` peut être utilisé sans problème pour la page `contact.php` :

```
<?php
// contact.php
$title='Contact';
include 'header.inc.php';
include 'menu.inc.php';
include 'contact.inc.php';
include 'footer.inc.php';
```

Avantages :

- Un seul fichier `header.inc.php` pour tout le site
- Le code est donc plus facilement maintenable

Point d'entrée unique

contact.php :

```
<?php
// contact.php
$title='Contact';
include 'header.inc.php';
include 'menu.inc.php';
include 'contact.inc.php';
include 'footer.inc.php';
```

aboutme.php :

```
<?php
// aboutme.php
$title='À propos';
include 'header.inc.php';
include 'menu.inc.php';
include 'aboutme.php';
include 'footer.inc.php';
```

Comment garantir que toutes les pages respectent ce format ?

```
<?php
// ...php
$title=...;
$content=...;
include 'header.inc.php';
include 'menu.inc.php';
include $content;
include 'footer.inc.php';
```

Données optionnelles dans l'URL

Une URL avec des données optionnelles :

```
http://dii.univ-mrs.fr/ade.php?cal=M2_CCI&getdate=20150908
```

Les données optionnelles sont :

- des couples clé=valeur
- séparées entre elles par des &
- séparées du début de l'URL par ?

Les clés et les valeurs sont des chaînes de caractères

Dans l'exemple, nous avons deux couples :

- la valeur M2_CCI est associée à la clé cal
- la valeur 20150908 est associée à la clé getdate

Plan

Objectifs

Protocole et serveur HTTP

Génération des pages

Utilisation de PHP

La variable `$_GET`

Organisation avec un point d'entrée unique

La variable `$_GET`

Filtrage des entrées

```
<?php  
$page_num = filter_input(INPUT_GET, 'page_num');  
?>
```

Après l'exécution de cette ligne, la variable `$page_num` vaut :

- NULL si `$_GET['page_num']` n'est pas définie
- la valeur de `$_GET['page_num']` sinon

Filtrage des entrées

```
<?php
$page_num = filter_input(
    INPUT_GET,
    'page_num',
    FILTER_VALIDATE_INT);
?>
```

Après l'exécution de ces lignes, la variable `$page_num` vaut :

- NULL si `$_GET['page_num']` n'est pas définie
- false si la valeur ne passe pas le filtre
- la valeur de `$_GET['page_num']` convertie en entier sinon

Filtrage des entrées

```
<?php
$options=array(
    'options'=>array(
        'default'=>5, 'min_range'=>0, 'max_range'=>9));
$page_num = filter_input(
    INPUT_GET,
    'page_num',
    FILTER_VALIDATE_INT,
    $options);
?>
```

Après l'exécution de ces lignes, la variable `$page_num` vaut :

- l'entier 5 si `$_GET['page_num']` n'est pas définie
- l'entier 5 si la valeur ne passe pas le filtre
- la valeur de `$_GET['page_num']` convertie en entier, sinon

Plan

Objectifs

Protocole et serveur HTTP

Génération des pages

Utilisation de PHP

La variable `$_GET`

Organisation avec un point d'entrée unique

Point d'entrée unique

Un seul point d'entrée :

```
http://...../index.php
```

Les différentes pages seront accessibles via ces URL :

```
http://...../index.php?page=aboutme
```

```
http://...../index.php?page=cv
```

```
http://...../index.php?page=blog
```

```
http://...../index.php?page=contact
```

Point d'entrée unique

Le fichier pages.inc.php :

```
<?php
$pages = array(
    'aboutme' => array(
        'title'=>'À Propos',
        'content'=>'aboutme.inc.php',
        'menu'=>'À Propos'),
    'cv' => array(
        'title'=>'Mon CV',
        'content'=>'cv.inc.php',
        'menu'=>'CV'),
    'blog' => array(
        'title'=>'Mon Blog',
        'content'=>'blog.inc.php',
        'menu'=>'Blog'),
    'contact' => array(
        'title'=>'Contact',
        'content'=>'contact.inc.php',
        'menu'=>'Contact');
);
$default_page = 'aboutme';
?>
```

Point d'entrée unique

Le fichier menu.inc.php :

```
<div class="navbar navbar-inverse navbar-static-top">
  <div class="container">
    <div class="navbar-header">
      <a class="navbar-brand" href="index.html">BOB</a>
    </div>
    <div class="navbar-collapse collapse">
      <ul class="nav navbar-nav navbar-right">
        <?php foreach ($pages as $key=>$page): ?>
          <li>
            <a href="?page=<?= $key ?>">
              <?=$page['menu'] ?>
            </a>
          </li>
        <?php endforeach: ?>
      </ul>
    </div>
  </div>
</div>
```

Point d'entrée unique

Le fichier index.php :

```
<?php
include 'pages.inc.php';

function get_page($page_id) {
    global $pages;
    if(array_key_exists($page_id, $pages))
        return $pages[$page_id];
    else null;
}
```

Point d'entrée unique

Le fichier index.php :

```
<?php
include 'pages.inc.php';

function get_page($page_id) {
}

$page = filter_input(INPUT_GET,
                    'page',
                    FILTER_CALLBACK,
                    array('options' => 'get_page'));

if ($page==null)
    $page = $pages[$default_page];
```

Point d'entrée unique

Le fichier index.php :

```
<?php

if ($page==null)
    $page = $pages[$default_page];

$title = $page['title'];
$content = $page['content'];

include 'header.inc.php';
include 'menu.inc.php';
include $content;
include 'footer.inc.php';
?>
```

Réécriture des URL

- Nous souhaitons que les différentes pages aient les URL suivantes :

```
http://...perso.luminy.univ-amu.fr
```

```
http://...perso.luminy.univ-amu.fr/aboutme.html
```

```
http://...perso.luminy.univ-amu.fr/cv.html
```

```
http://...perso.luminy.univ-amu.fr/blog.html
```

```
http://...perso.luminy.univ-amu.fr/contact.html
```

- Nous souhaitons également qu'il ne soit pas possible d'exécuter ou d'accéder aux autres fichiers du répertoire

Les fichiers `.htaccess` de configuration d'Apache

Si le module de réécriture des règles est installé, on peut écrire dans le fichier `.htaccess` les lignes suivantes :

```
RewriteEngine On
RewriteRule ^$ index.php [END,QSA]
RewriteRule ^(.*)\.html$ index.php?page=$1 [END,QSA]
RewriteRule ^(.*)$ - [L,R=404]
```

La syntaxe :

```
RewriteRule Modèle Substitution [drapeaux]
```

Dans les sous-répertoires :

```
RewriteEngine Off
```

Les fichiers `.htaccess` de configuration d'Apache

La syntaxe :

```
RewriteRule Modèle Substitution [drapeaux]
```

Le modèle est une expression régulière :

- `^` : début de ligne
- `$` : fin de ligne
- `.` : n'importe quel caractère
- `()` : pour définir les `$x` dans la substitution

Les drapeaux :

- `END` : termine le processus de réécriture pour ce répertoire
- `QSA` : conserve les chaîne de paramètres de l'URL de départ
- `L` : termine le processus de réécriture
- `R=xxx` : force une redirection avec un code de statut HTTP optionnel