

Cours de Développement Web

Introduction au PHP

Frédéric Flouvat

Université de la Nouvelle-Calédonie

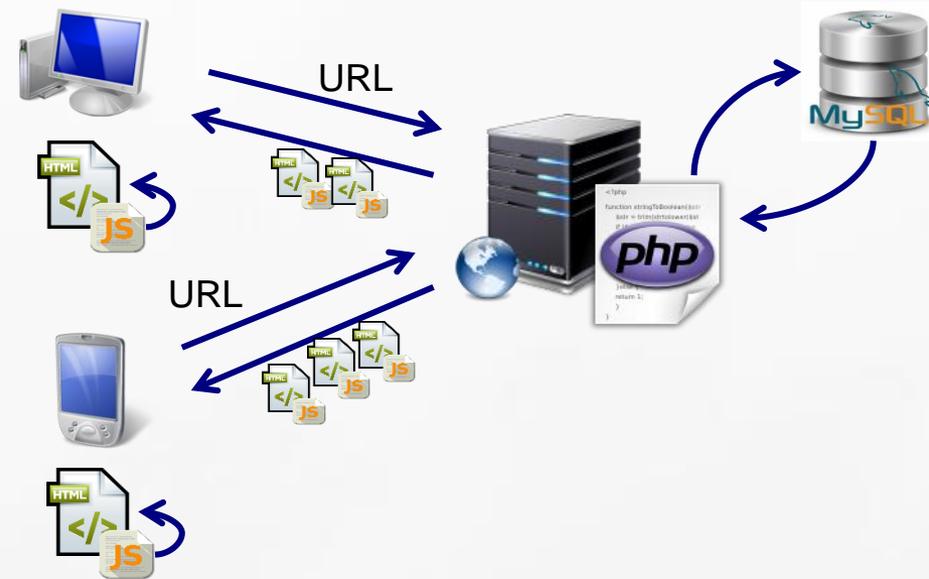
frederic.flouvat@univ-nc.nc



Rappel

- Le PHP, un langage côté serveur
 - génère dynamiquement du HTML
 - accède aux bases de données
 - totallement invisible pour les clients

- Coder en PHP
 - nécessité d'avoir un serveur web d'installé avec un module PHP
 - utiliser des environnements de développement, tel que *Xampp*, pour travailler en local (<http://localhost>)



- Exécution du code PHP
 - interprétation et exécution du code faite par un moteur PHP intégré au serveur web
 - erreurs PHP insérées dans le code HTML et affichées par le navigateur
 - possibilité d'utiliser des debugger tel que *phpdbg* (intégré par défaut à partir de PHP 5.6)

Le PHP

- Langage de programmation dont l'objectif est de générer du HTML grâce à des ***echo*** ou des ***print***
 - une syntaxe similaire au C/C++/Java, mais un fonctionnement et une philosophie différents sur certains aspects (p.ex. les variables, les tableaux)
- Fichier source PHP = fichier texte avec l'extension **.php**
 - code PHP entre balises **<?php ?>**
 - peut contenir directement du HTML

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    ...
  </head>
  <body>
    <p>
      <?php
        echo phpinfo();
      ?>
    </p>
  </body>
</html>
```

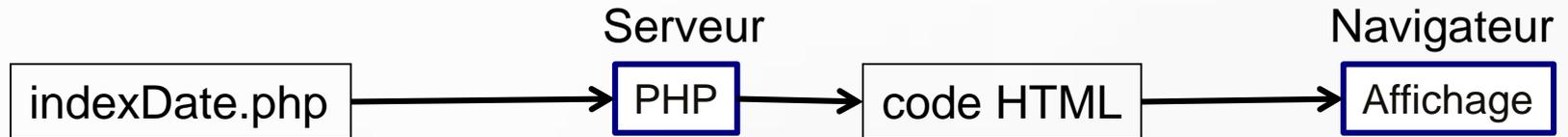
Le PHP

Autre exemple:

fonction PHP retournant une chaîne de caractères représentant la date courante

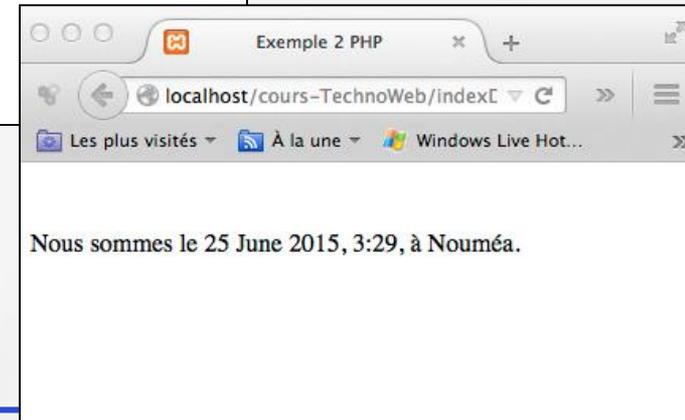
```
<!DOCTYPE html>
<html lang="fr">
  <head>
    ...
  </head>
  <body>
    <p>Nous sommes le
      <?php
        echo date("F j Y, G:i,") ;
      ?>
    à Nouméa.
  </p>
</body>
</html>
```

Fonctionnement du PHP



```
indexDate.php
indexDate.php > No Selection
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title>Exemple 2 PHP</title>
5 <meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
6 </head>
7 <body>
8 <p>Nous sommes le
9 <?php
10     echo date("j F Y, G:i,") ;
11 <?>
12     à Nouméa.
13 </p>
14 </body>
15 </html>
```

```
Source de : http://localhost/cours-TechnoWeb/indexDate.php
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title>Exemple 2 PHP</title>
5 <meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
6 </head>
7 <body>
8 <p>Nous sommes le
9     25 June 2015, 3:30,      à Nouméa.
10 </p>
11 </body>
12 </html>
13
```



Le langage et ses spécificités

- ☐ PHP offre les mêmes possibilités que des langages tels que C/C++/Java/JavaScript
 - des variables (locales et globales)
 - des tableaux
 - des structures de contrôle (**while**, **for**, **do**, **if**, **switch**)
 - des fonctions (avec passage des paramètres par valeur ou par adresse)
 - ...
- ☐ Mais avec
 - une gestion des types très souples (langage faiblement typé)
 - une syntaxe spécifique pour les variables et les fonctions
 - des fonctions et des opérateurs spécifiques pour manipuler les chaînes de caractères, les types, les tableaux, ...

Les similarités syntaxiques avec le C/C++/Java/JavaScript

- ☐ Instructions terminées par ;
- ☐ Délimitation des blocs avec { ... }
- ☐ Mise en commentaires avec // ou /* ... */
- ☐ Utilisation de structures de contrôles
 - **while(cond) { ... }**
 - **do{ ... } while(cond);**
 - **for(init ; cond ; modif){ ... }**
 - **if(cond){ ... } else if(cond){ ... } else { ... }**
 - **switch(expr){ case val_1: ... break; case val_2: ... break; default: ... }**
- ☐ Définition des opérateurs
 - arithmétiques: +, -, *, /, %, ++, --
 - d'affectation: =, +=, -=, *=, /=, .=
 - de comparaison: ==, <, >, <=, >=, !=
 - logiques: &&, ||, ^, !

Les variables en PHP

☐ Pas de déclaration

- type défini au moment de l'affectation

☐ Commence par \$

- ex: \$i

☐ Affectation par valeur =

- ex: \$i = 1;

```
$foo = 'original' ;  
$bar = $foo ;  
$bar = 'modifié' ;  
echo $foo ;
```

```
$foo = 'original' ;  
$bar =& $foo ;  
$bar = 'modifié' ;  
echo $foo ;
```

☐ Affectation par référence =&

- ex.: \$i =& \$tmp ;

☐ Possibilité de définir des variables de variables (variables dites dynamiques) \$\$ ou \${\$...}

- variable prenant pour valeur une variable et l'utilisant comme nom d'une autre variable

```
$var = 'bonjour' ;  
$$var = 'monde' ;  
echo $var ;  
echo ${$var} ;  
echo $$var ;  
echo $bonjour ;
```

Les chaînes de caractères en PHP

- Deux syntaxes avec des interprétations différentes
 - chaînes de caractères délimitées par des guillemets simples (p.ex. *'chaîne'*)
 - contenu non interprété
 - i.e. les variables apparaissant dans des guillemets simples ne sont pas remplacées
 - chaînes de caractères délimitées par des guillemets doubles (p.ex. *"chaîne"*)
 - contenu interprété
 - i.e. si une variable apparaît dans la chaîne, elle est remplacée par sa valeur

```
$age = 20 ;  
echo ' j'ai $age ans' ;  
echo " j'ai $age ans" ;
```

j'ai \$age ans
j'ai 20 ans

Conseil: guillemets simples ou doubles ?

- utiliser les guillemets simples au niveau PHP
- utiliser les guillemets doubles au niveau HTML

```
echo ' <div class="contenu"> blablabla </div> ' ;  
echo ' j'ai '.$age.' ans' ;
```

Les chaînes de caractères en PHP

☞ Afficher des caractères réservés: mettre \ devant

☞ Concaténer deux chaînes de caractères: .

```
echo 'j\'ai '.$age.' ans' ;  
$di = $age[0] ;
```

☞ Accéder à un caractère `var[index]`

☞ Convertir en chaîne de caractères: **(string)** devant une variable ou fonction **strval(\$var)**

☞ Autres fonctions utiles

- avoir la longueur d'une chaîne de caractères: **strlen(\$var)**
- extraire une sous-chaîne: **substr(\$var, \$start, \$len)** ou **substr(\$var, \$start)**
 - extrait la sous-chaîne de \$str commençant au caractère \$start, et faisant éventuellement \$len caractères de long

Les tableaux en PHP

- Allocation du tableau et ajout d'éléments fait dynamiquement par PHP
- Plusieurs types de tableaux
 - les tableaux indicés
 - affectation automatique d'indice si pas précisé (indice de la première case vide)

```
$tab = array() ;  
$tab[0]='hello ' ;  
$tab[] = 'world ' ;
```

- les tableaux associatifs
 - pas d'indices entiers mais une clé en chaîne de caractères

```
$mymovies = array('Vertigo'=>'Hitchcock', 'Alien'=>'Scott');  
  
$films = array();  
$films['Burton'] = 'Beetlejuice' ;  
$films['Burton'] = 'Sleepy hollow' ;
```

Les tableaux en PHP

Plusieurs types de tableaux (suite)

- les tableaux multi-dimensionnels (indicés et/ou associatifs)

```
$mymovies = array( 'Vertigo'=>array('Alfred','Hitchcock'), 'Alien'=>array('Ridley','Scott') );  
$films = array();  
$films['Burton'][1] = 'Beetlejuice' ;  
$films['Burton'][5] = 'Sleepy hollow' ;
```

Deux modes de parcours

- parcours classiques avec une boucle **for** ou **while**
- parcours spécifiques avec **foreach(...)**

```
foreach( $tab as $value )  
{  
  ...  
}
```

```
foreach( $tab as $key => $value )  
{  
  ...  
}
```

Beaucoup de fonctions mises à dispositions

- unset** (effacer), **array_diff** (faire la différence entre deux tableaux), **sort** (trier)

Les tableaux en PHP

```
<?php
$stab = array();
$stab['cle'] = 'Hello';
$stab[2] = 'World';
$stab[0] = 'Bye';
echo 'Le tableau a '.count($stab).' élément(s):<ul>';
foreach ($stab as $key=>$valeur) {
    echo '<li> '.$key.' associé à '.$valeur.'</li>';
}
echo '</ul>';
var_dump($stab);
?>
```

affiche les informations d'une variable, ici le tableau (**très utile lors du débogage**)

```
<?php
$menu = array();
$menu['Home'] = 'index.html';
$menu['News'] = 'news.html';
$menu['Contact'] = 'contact.html';
echo '<ul>';
foreach ($menu as $nom => $url ) {
    echo '<li><a href='.$url.'>$nom</a></li>';
}
echo '</ul>';
?>
```

Les fonctions en PHP

```
<?php
function somme( $a, $b){
    return $a + $b;
}

echo somme(1,2) ;
echo $a;
?>
```

pas besoin de déclarer le type de retour,
ni le type des paramètres

pas les mêmes variables !!

```
$a = 1;
$b = 2;
function somme(){
    global $a, $b;
    res = $a + $b;
}
echo $res;
```

Portée des variables

- locale (par défaut)
 - locale à la fonction (si dans une fonction),
locale à la page web (sinon)
- globales
 - variable accessible en dehors des fonctions
- statiques
 - portée locale mais conserve sa valeur lorsque
le script appelle la fonction plusieurs fois

```
function test()
{
    static $count = 0;
    $count++;
    echo $count;
    if ($count < 10)
        test();

    $count--;
}
```

Les fonctions en PHP

Le passage des paramètres

- par valeur (par défaut)
 - une copie des paramètres est faite au moment de l'appel
 - modifications non conservées
- par référence: **&** devant le paramètre
 - pas de recopie
 - modifications conservées

```
function add_some_extra( &$string )
{
    $string .= ', et un peu plus.';
}
$str = 'Ceci est une chaîne';
add_some_extra($str);
echo $str;
```

Définir des valeurs par défaut

function *nom_fonction*(*\$param0*, *\$param1=valDef1*, *\$param2=valDef2*, ...)

Astuces:

- possibilité d'avoir un nombre variable de paramètres en utilisant un tableau en paramètre
- si le nom d'une variable est suivi de parenthèses, PHP recherchera une fonction de même nom, et essaiera de l'exécuter → utile p.ex. pour faire des tableaux de fonctions

Variables, boucles, tableaux associatifs, fonction, API: exemple

🚦 Récupérer et afficher la liste des villes en Nouvelle-Calédonie

```
function cityNC($name)
{
    $url = 'https://data.gouv.nc/api/records/1.0/search/';
    $request_url = $url .'?dataset=communes-nc&q='. urlencode($name).'&rows=50';

    // initialisation d'une session
    $curl = curl_init($request_url);

    // spécification des paramètres de connexion
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSE);
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, FALSE);

    // envoie de la requête et récupération du résultat sous forme d'objet JSON
    $response = json_decode(curl_exec($curl));

    // fermeture de la session
    curl_close($curl);

    // stockage des villes et des codes postaux dans un tableau associatif
    foreach( $response->records as $infoville ){
        $villes[$infoville->fields->nom_minus]=$infoville->fields->code_post;
    }

    return $villes;
}
```

Récupération sur le site du gouvernement par l'intermédiaire d'une API REST

- *Application Programming Interface* : ensemble de spécifications permettant à des logiciels de s'échanger des services ou des données
- *REpresentational State Transfer* : architecture visant à créer des services sur le Web, et à interagir avec eux
 - un service = une URL;
 - une interaction = une requête (p.ex. via le protocole HTTP);
 - un résultat = un document (p.ex. un JSON)

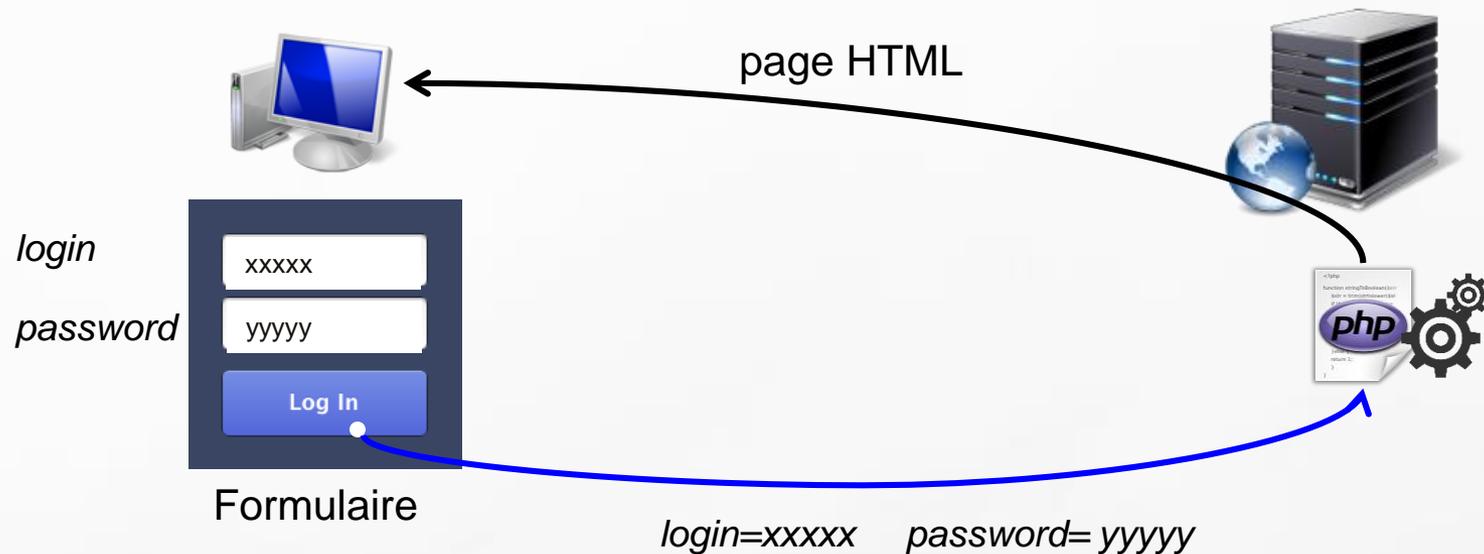
```
$villesNC = cityNC("");

echo '<ul>';
foreach( $villesNC as $nom => $cp ){
    echo '<li>.$nom.' ('.$cp.')</li>';
}
echo '</ul>';
```

Exercices

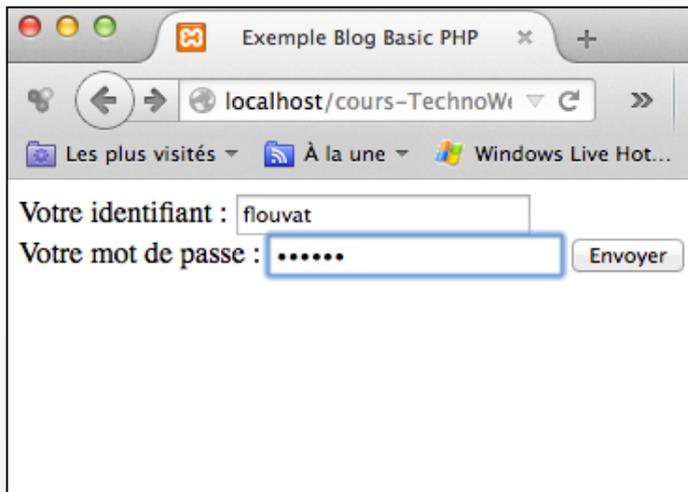
- Créer une fonction retournant la liste des pays
- Créer une fonction retournant la liste des offres d'emploi contenant un mot passé en paramètre
- Indication: utiliser pour cela l'API data.gouv.nc

PHP et formulaires HTML: principe



- Saisie des données par un utilisateur dans un formulaire **HTML** et transmission au serveur web (pour traitement) au moment de la validation du formulaire
 - différents composants de saisie: champs textuels, listes déroulantes, cases à cocher, ...
 - validation formulaire = clic bouton soumission du formulaire (*submit*)
- Traitement des données saisies en PHP

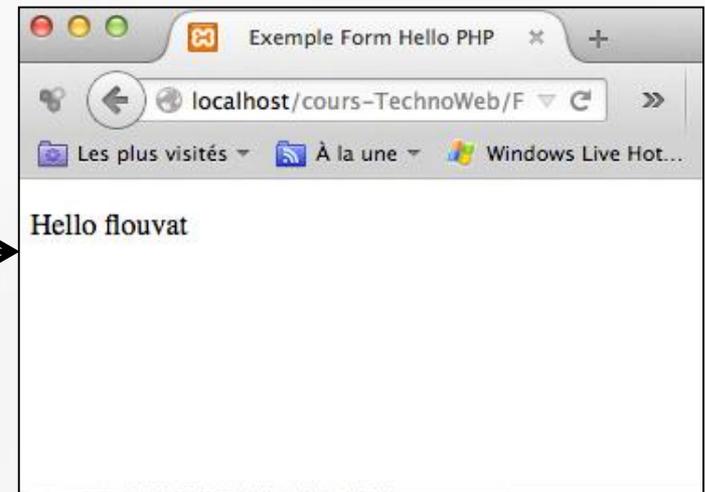
PHP et formulaires HTML: exemple



A screenshot of a web browser window titled "Exemple Blog Basic PHP". The address bar shows "localhost/cours-TechnoWi". The page content includes a form with two input fields: "Votre identifiant : flouvat" and "Votre mot de passe :". A button labeled "Envoyer" is positioned to the right of the password field.

index.html

login,password



A screenshot of a web browser window titled "Exemple Form Hello PHP". The address bar shows "localhost/cours-TechnoWeb/F". The page content displays "Hello flouvat".

annonces.php

PHP et formulaires HTML: exemple

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <title>Exemple Blog Basic PHP</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>
  <form method="post" action="annonces.php">

    <label for="idlogin"> Votre identifiant </label> :
    <input type="text" name="login" id="idlogin" placeholder="defaut" />

    <br />

    <label for="idpassword"> Votre mot de passe </label> :
    <input type="password" name="password" id="idpassword" />

    <input type="submit" value="Envoyer">

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

index.html

login
password

annonces.php

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <title>Exemple Form Hello PHP</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>

  <p> Hello <?php echo $_POST['login']; ?> </p>

</body>
</html>
```

PHP et formulaires HTML: les balises HTML et leurs attributs

- La balise principale : **<form> ... </form>**
 - url de la page web recevant les données:
attribut **action**
 - méthode de transmission des données:
attribut **method="POST" ou "GET"**
- Attention:** impossible d'inclure un formulaire dans un autre
- Par contre, possible d'avoir plusieurs formulaires dans une même page

```
<form method="..." action="..." >
  <label for="...">... </label>
  <input type="..." name="..." id="..." ... />
  <label for="...">... </label>
  <textarea name="..." id="..." ... >
    ...
  </textarea>
  <label for="...">... </label>
  <select name="..." id="..." ... >
    <option> ... </option>
    <option> ... </option>
    ...
  </select>
  <button type="..." name="..." id="..." ... >
    ...
  </button>
</form>
```

PHP et formulaires HTML: les balises HTML et leurs attributs

Les sous-balises de **form**

- les principaux composants de saisie

<input type="..." ... />

- avec **type**=*text, checkbox, radio, file, image, password, hidden, tel, email, search, number, datetime, date, submit, reset, button...*

<textarea> ... </textarea> (zone de texte)

<select> ... </select> (liste déroulante)

et leurs attributs

- name**: nom de la variable dans laquelle la valeur saisie va être stockée
- id**: identifiant de la sous-balise
- required, placeholder, autofocus, autocomplete, pattern, list, maxlength,**
...

```
<form method="..." action="..." >

  <label for="...">... </label>
  <input type="..." name="..." id="..." ... />

  <label for="...">... </label>
  <textarea name="..." id="..." ... >
    ...
  </textarea>

  <label for="...">... </label>
  <select name="..." id="..." ... >
    <option> ... </option>
    <option> ... </option>
    ...
  </select>

  <button type="..." name="..." id="..." ... >
    ...
  </button>

</form>
```

PHP et formulaires HTML: les balises HTML et leurs attributs

- Les sous-balises de **form** (suite)
 - Les étiquettes **<label> ... </label>**
 - attribut **for**: identifiant du composant de saisie
 - <button type="..."> ... </button>** versus **<input type="..." />**
 - avec **type=submit, reset, button**
 - tous les deux, un bouton
 - seule différence: plus de possibilités pour modifier le style avec **button**
 - changer le texte, faire un bouton à partir d'une image, ...
 - attention: des problèmes avec certaines versions de IE (p.ex. IE6) et button

```
<form method="..." action="..." >
  <label for="...">... </label>
  <input type="..." name="..." id="..." ... />

  <label for="...">... </label>
  <textarea name="..." id="..." ... >
    ...
  </textarea>

  <label for="...">... </label>
  <select name="..." id="..." ... >
    <option> ... </option>
    <option> ... </option>
    ...
  </select>

  <button type="..." name="..." id="..." ... >
    ...
  </button>

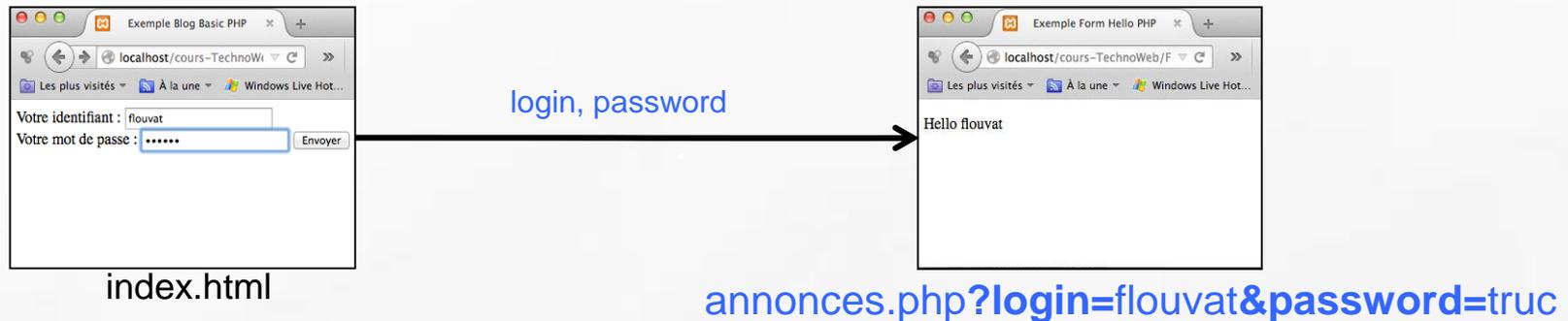
</form>
```

PHP et formulaires HTML: retour sur la transmission des paramètres

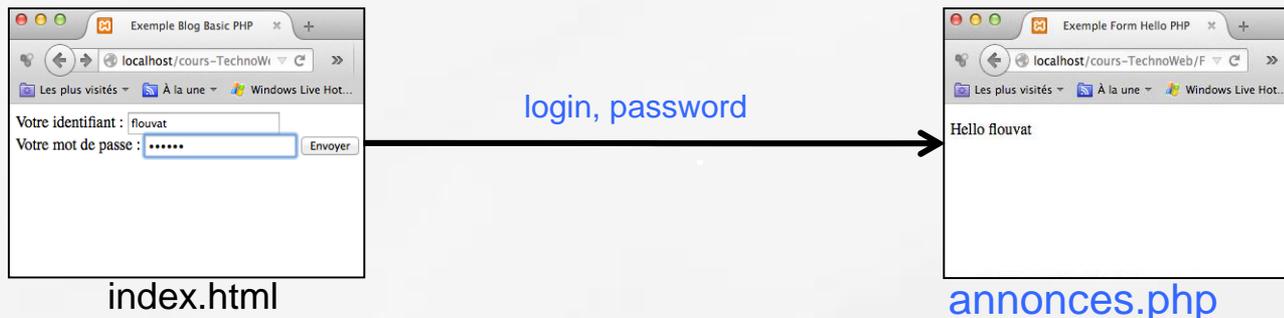
☐ Deux méthodes de transmission des paramètres `<form method="..." ... >...</form>`

- **get** : transmission des données dans l'adresse de la page

url?var1=val1&var2=val2 ...



- **post** : transmission des données séparément de l'adresse web
➔ permet de cacher les informations transmises



PHP et formulaires HTML: récupération des données en PHP

Principe:

- l'interpréteur PHP initialise un tableau associatif qui associe à chaque nom de paramètre sa valeur

Le nom du tableau initialisé dépend de la méthode:

- **\$_POST['...']** pour la méthode **post**
- **\$_GET['...']** pour la méthode **get**

données transmises
en mode post

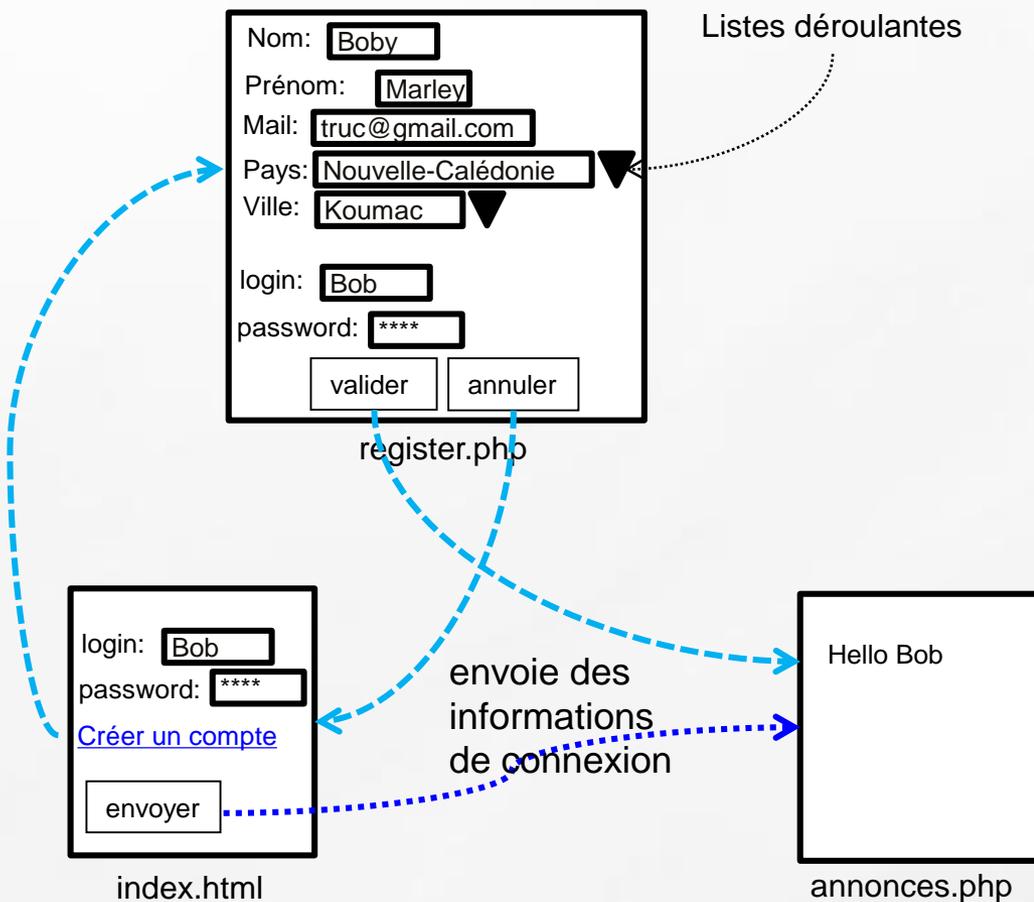
```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Exemple Form Hello PHP</title>
    <meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
  </head>
  <body>

    <p> Hello <?php echo $_POST['login']; ?> </p>

  </body>
</html>
```

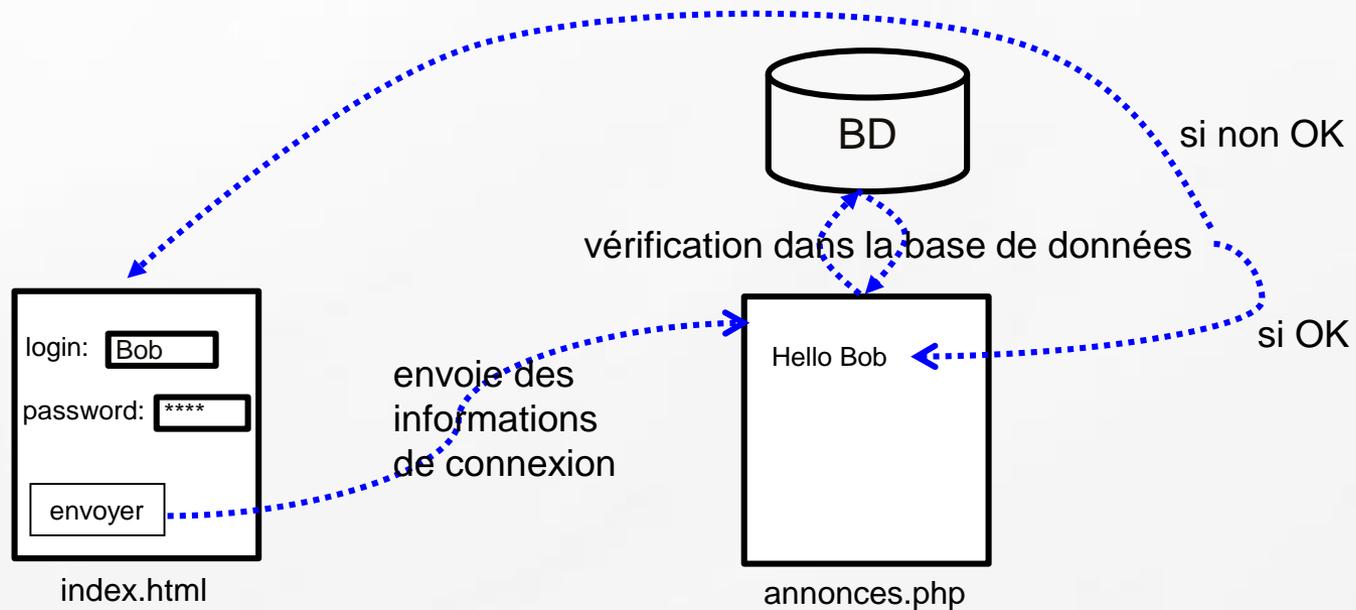
Exercice

- Créer un formulaire permettant à un utilisateur de saisir les informations lui permettant de créer un compte



Bases de données et PHP: exemple

- Accès à une bases de données pour vérifier un login et un mot de passe



Bases de données et PHP: exemple

```
<?php
$link = mysqli_connect('localhost', 'root', '', 'jobstage_db');
$query = 'SELECT login FROM Users WHERE login="' . $_POST['login'] . '" and password="' . $_POST['password'] . '"';
$resultlogin = mysqli_query($link, $query );

if ( ! mysqli_num_rows( $resultlogin ) ) {
    mysqli_free_result( $resultlogin );
    header( "refresh:5;url=index.html" );
    echo 'Erreur de login et/ou de mot de passe (redirection automatique dans 5 sec.)';
    exit;
}

?>

<!DOCTYPE html>
<html lang="fr">
<head>
<title>Exemple Form Hello PHP</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>

<p> Hello <?php echo $_POST['login']; ?> </p>

</body>
</html>

<?php mysqli_close( $link ); ?>
```

1

2

3

4

- 1. Connexion à la base de données
- 2. Envoi d'une requête SQL
- 3. Récupération du nombre de résultats
 - Si ! OK alors
 - libération de la mémoire occupée par les résultats
 - redirection vers la page de connexion et affichage d'un message d'erreur
 - Sinon
 - affichage du reste de la page
- 4. Fermeture de le connexion

annonces.php

Bases de données et PHP: les étapes

1) Connexion au SGBD et sélection de la base de données

- fonction **mysqli_connect(\$server, \$user, \$pwd, \$database)** retourne la connexion à la base de données

```
$link = mysqli_connect('localhost', 'root', '', 'jobstage_db');
```

1) Envoi d'une requête SQL

- fonction **mysqli_query(\$link, \$query)** retourne le résultat de la requête dans une structure de données **mysqli_result** (si requête SELECT) ou true/false (si requête INSERT, UPDATE, DELETE)
 - utiliser **mysqli_error(\$link)** pour récupérer les messages d'erreurs de MySQL

```
$resultall = mysqli_query($link, 'SELECT id, title FROM Post');
```

- **attention:** pas de point-virgule à la fin de la requête

Bases de données et PHP: les étapes

3) Récupération et utilisation du résultat (un tuple à la fois)

- fonction **mysqli_fetch_assoc(\$result)** retourne un tableau associatif qui correspond à la ligne récupérée (ligne en cours) ou NULL s'il n'y a plus de ligne
- nécessité d'utiliser une boucle pour parcourir s'il y a plusieurs tuples dans le résultat

```
<ul>
  <?php while ( $row = mysqli_fetch_assoc($resultall) ): ?>
    <li>
      <?php echo $row['title']; ?>
    </li>
  <?php endwhile ?>
</ul>
```

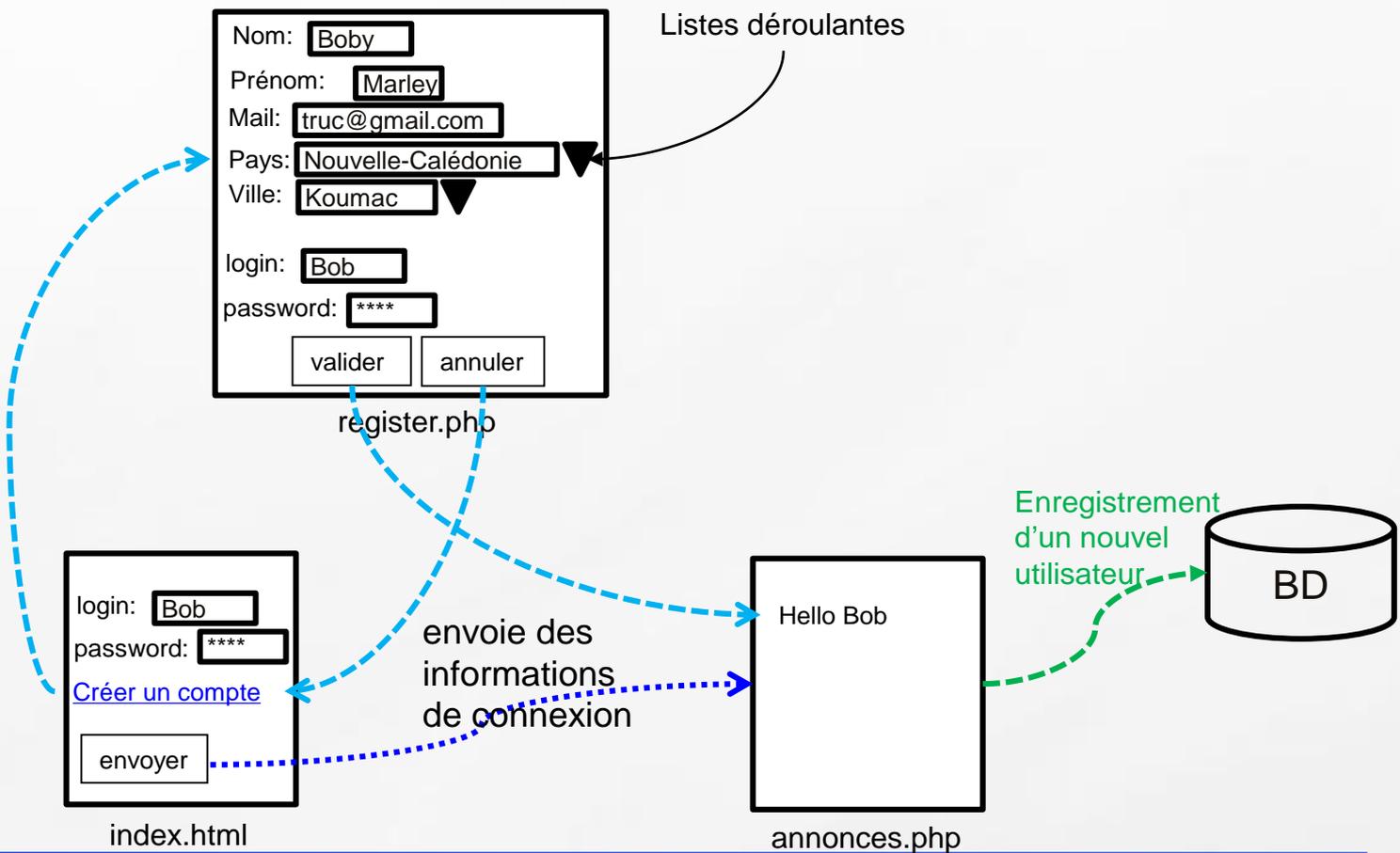
3) Fermeture de la connexion

- fonction **mysqli_close(\$link)** retourne true si la connexion a été fermée
 - automatique en fin de page

```
mysqli_close( $link );
```

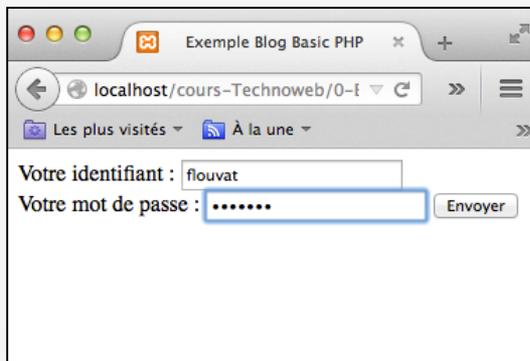
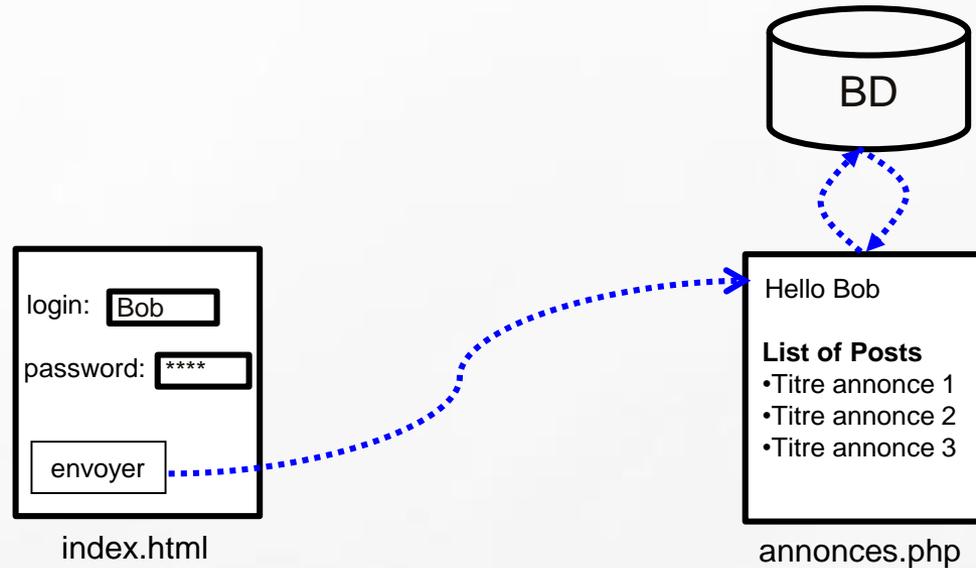
Exercice

- Modifier la page `annonces.php` pour inclure l'enregistrement des informations d'un utilisateur nouvellement inscrit dans la base de données



Bases de données et PHP: exemple

- Afficher la liste des annonces



Bases de données et PHP: exemple

```
<?php
$link = mysqli_connect('localhost', 'root', '', 'jobstage_db');
$query= 'SELECT login FROM Users WHERE login="'. $_POST['login'].'" and password="'. $_POST['password'].'';
$resultlogin = mysqli_query($link, $query );

if( mysqli_num_rows( $resultlogin )){
    mysqli_free_result( $resultlogin );
    $resultall = mysqli_query($link, 'SELECT id, title FROM Post');
}
else{
    header( "refresh:5:url=index.html" );
    echo 'Erreur de login et/ou de mot de passe (redirection automatique dans 5 sec.)';
    exit;
}
?>

<!DOCTYPE html>
<html lang="fr">
<head>
...
</head>
<body>
<p> Hello <?php echo $_POST['login']; ?> </p>
<h1>List of Posts</h1>
<ul>
    <?php while ($row = mysqli_fetch_assoc($resultall)): ?>
        <li><?php echo $row['title']; ?> </li>
    <?php endwhile ?>
</ul>
</body>
</html>

<?php  mysqli_close( $link );?>
```

Modifier
annonces.php

Transmettre des données d'une page à l'autre via un lien hypertexte

Transmettre des variables et leurs valeurs directement dans un lien hypertexte

- *page "origine"*: créer un simple lien hypertexte et ajouter les variables à la fin de l'url
 - équivalent à une transmission de données via un formulaire en **get**

```
<a href="article.php?nom=John&prenom=Do">lien</a>
```

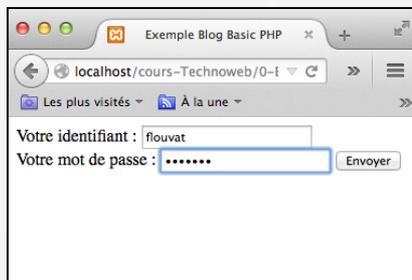
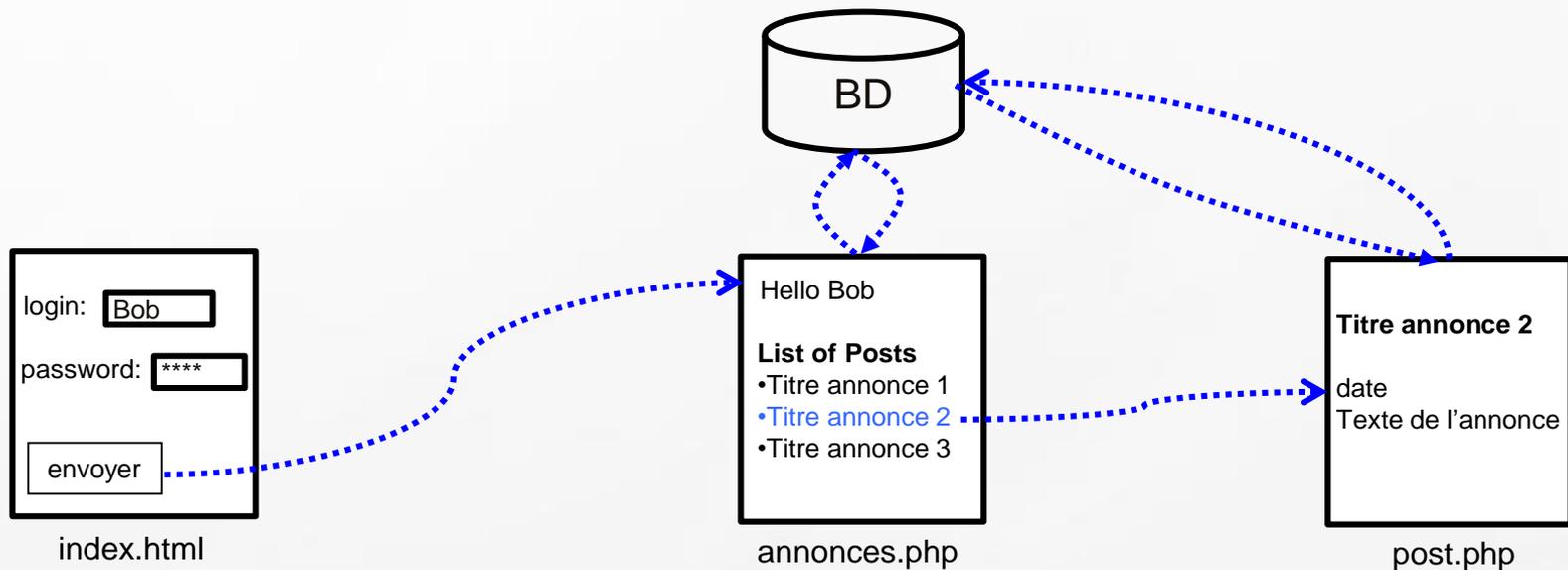
- *page "destination"*: utiliser le tableau `$_GET[...]` pour la valeur en PHP

```
<?php
    echo $_GET['nom'];
    echo $_GET['prenom'];
?>
```

Transmettre des données d'une page à l'autre via un lien hypertexte: exemple

☐ Afficher le détail d'une annonce

- lorsque l'on clique sur le titre d'un message ("*post*"), on est redirigé vers une page qui affiche le texte associé



Transmettre des données d'une page à l'autre via un lien hypertexte: exemple

```
...
<h1>List of Posts</h1>
<ul>
  <?php while ($row = mysqli_fetch_assoc($resultall)): ?>
    <li>
      <a href="post.php?id=<?php echo $row['id']; ?>">
        <?php echo $row['title']; ?>
      </a>
    </li>
  <?php endwhile ?>
</ul>
...
```

Modifier
annonces.php

post.php

```
<?php
$link = mysqli_connect('localhost', 'root', '', 'jobstage_db');
$result = mysqli_query($link, 'SELECT * FROM Post WHERE id='.$_GET['id']);
$post = mysqli_fetch_assoc($result);
?>

<!DOCTYPE html>
<html lang="fr">
<head>
  <title>Exemple Annonces Basic PHP</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>

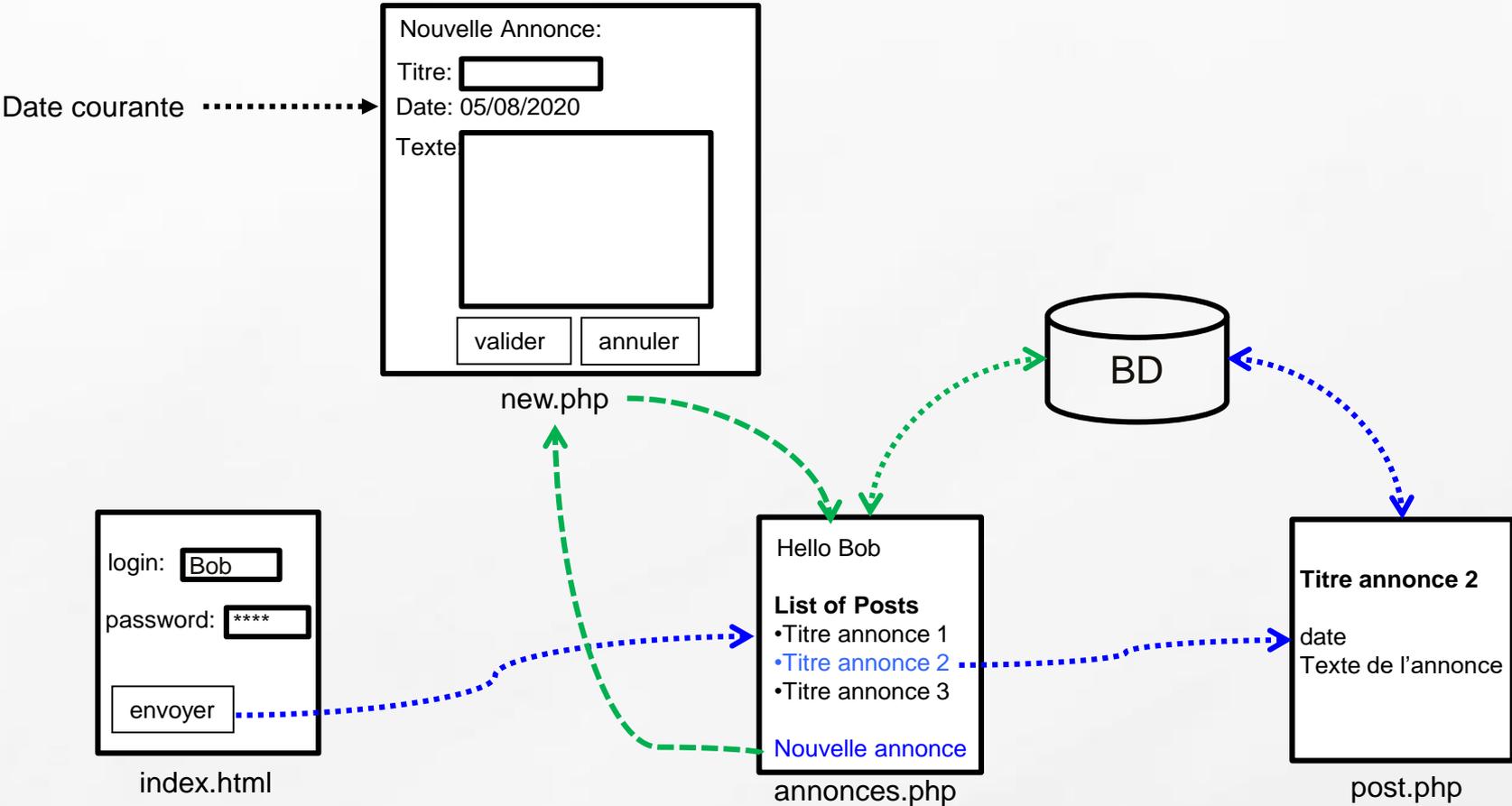
  <h1><?php echo $post['title']; ?></h1>
  <div class="date"> <?php echo $post['date']; ?> </div>
  <div class="body"> <?php echo $post['body']; ?> </div>

</body>
</html>

<?php  mysqli_close( $link );?>
```

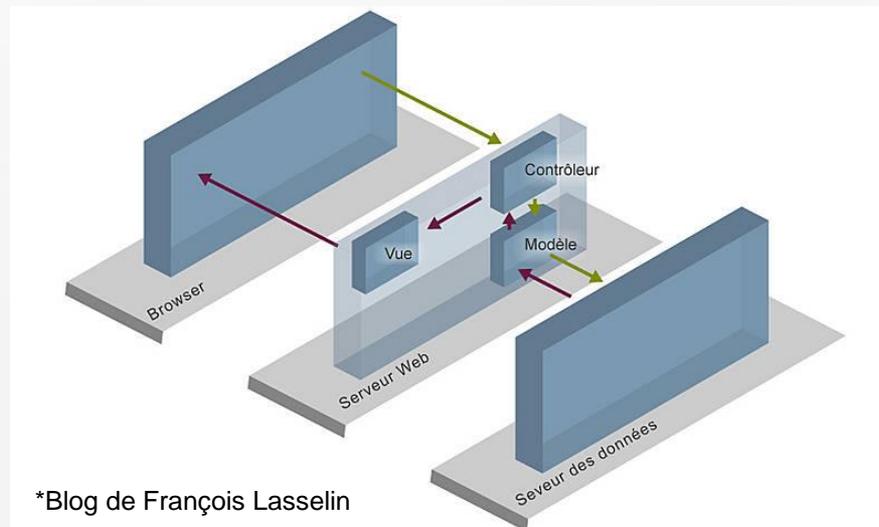
Exercice

- Modifier l'application afin d'intégrer un formulaire permettant de saisir de nouvelles annonces et de les enregistrer dans la base de données



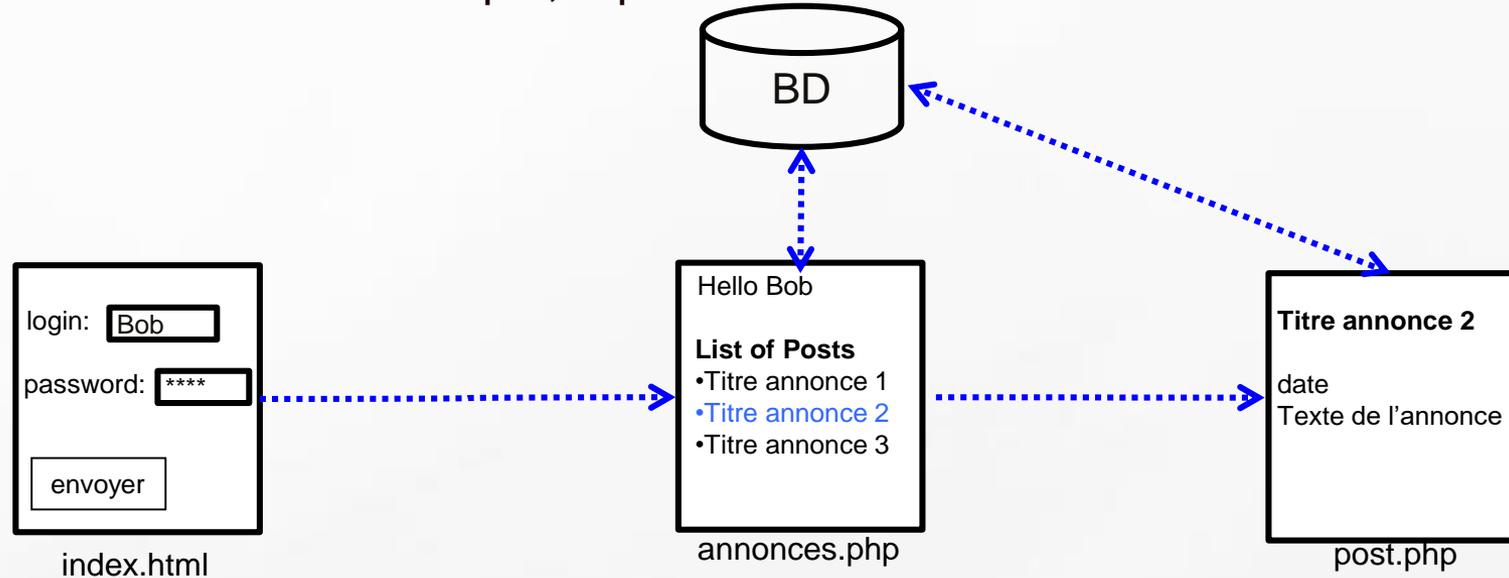
MVC et PHP: principe

- Le patron de conception (*design pattern*) Modèle-Vue-Contrôleur (MVC)
 - un modèle d'architecture logicielle = une solution générale permettant de résoudre un problème classique en développement logiciel
 - séparation de l'accès aux données (le modèle), de l'interface (la vue) et des traitements (le contrôleur)
 - simplifie la maintenance et l'évolution du logiciel
- Mise en œuvre un peu particulière dans le cas des pages web car c'est le navigateur qui gère l'affichage et non l'application



MVC et PHP: exemple du site d'annonces *

- Un site d'annonces simple, rapide à écrire



- tout est fortement couplé/lié
 - difficile à faire évoluer
- Comment le rendre plus modulaire et évolutif ?
 - Appliquer l'architecture MVC

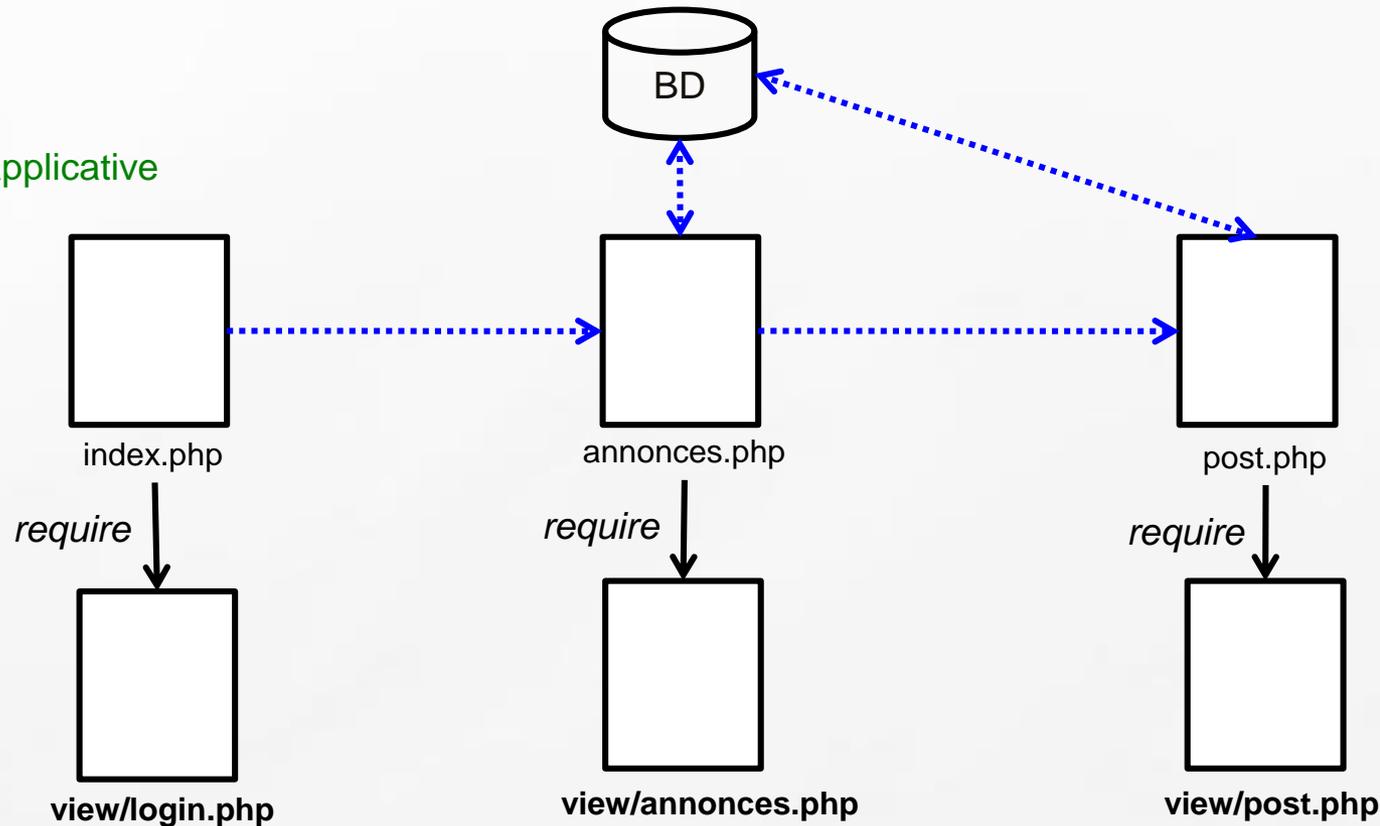
*cf Symfony versus PHP pur https://symfony.com/doc/current/introduction/from_flat_php_to_symfony.html

MVC et PHP: exemple du site d'annonces *

Etape 1: Isoler la présentation / l'interface

Fichiers chargés de la logique applicative
(appelée aussi **contrôleurs**)

- traite les entrées utilisateurs
- prépare une réponse



Fichiers chargés de la
présentation HTML

MVC et PHP: exemple du site d'annonces *

*inclut le fichier chargé de
l'affichage du login*

```
<?php  
require 'view/login.php';  
?>
```

index.php

```
<!DOCTYPE html>  
<html lang="fr">  
<head>  
  <title>Exemple Annonces Basic PHP</title>  
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
</head>  
<body>  
  <form method="post" action="annonces.php">  
    <label for="login"> Votre identifiant </label> :  
    <input type="text" name="login" id="login" placeholder="default" maxlength="12" required />  
    <br />  
    <label for="password"> Votre mot de passe </label> :  
    <input type="password" name="password" id="password" maxlength="12" required />  
  
    <input type="submit" value="Envoyer">  
  </form>  
</body>  
</html>
```

view/login.php

MVC et PHP: exemple du site d'annonces *

```
<?php
$link = mysqli_connect('localhost', 'root', '', 'jobstage_db');
$query= 'SELECT login FROM Users WHERE
login="$_POST[login]." and password="$_POST[password]."';

$resultlogin = mysqli_query($link, $query );

if( mysqli_num_rows( $resultlogin )){
    $login = $_POST['login'];
    mysqli_free_result( $resultlogin );
    $resultall = mysqli_query($link, 'SELECT id, title FROM Post');
    $posts = array();

    while ($row = mysqli_fetch_assoc($resultall)) {
        $posts[] = $row;
    }
}
mysqli_close( $link );

// inclut le code de la présentation HTML
require 'view/annonces.php';
?>
```

annonces.php

```
<?php
if( !isset( $login )){
    header( "refresh:5:url=index.php" );
    echo 'Erreur de login et/ou de mot de passe (redirection
automatique dans 5 sec.)';
    exit;
}
?>

<!DOCTYPE html>
<html lang="fr">
<head>
<title>Exemple Annonces Basic PHP</title>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
</head>
<body>
<p> Hello <?php echo $login; ?> </p>
<h1>List of Posts</h1>
<ul>
<?php foreach( $posts as $post ) : ?>
<li>
<a href="post.php?id=<?php echo $post['id']; ?>">
<?php echo $post['title']; ?>
</a>
</li>
<?php endforeach ?>
</ul>
</body>
</html>
```

view/annonces.php

récupère les données et les stocke dans des variables utilisées dans la présentation

MVC et PHP: exemple du site d'annonces *

```
<?php
$link = mysqli_connect('localhost', 'root', '', 'jobstage_db');
$result = mysqli_query($link, 'SELECT * FROM Post WHERE id='.$_GET['id']);

$post = mysqli_fetch_assoc($result);
mysqli_close($link);

// inclut le code de la présentation HTML
require 'view/post.php';
?>
```

post.php

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <title>Exemple Annonces Basic PHP</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>

  <h1><?php echo $post['title']; ?></h1>
  <div class="date"> <?php echo $post['date']; ?> </div>
  <div class="body"> <?php echo $post['body']; ?> </div>

</body>
</html>
```

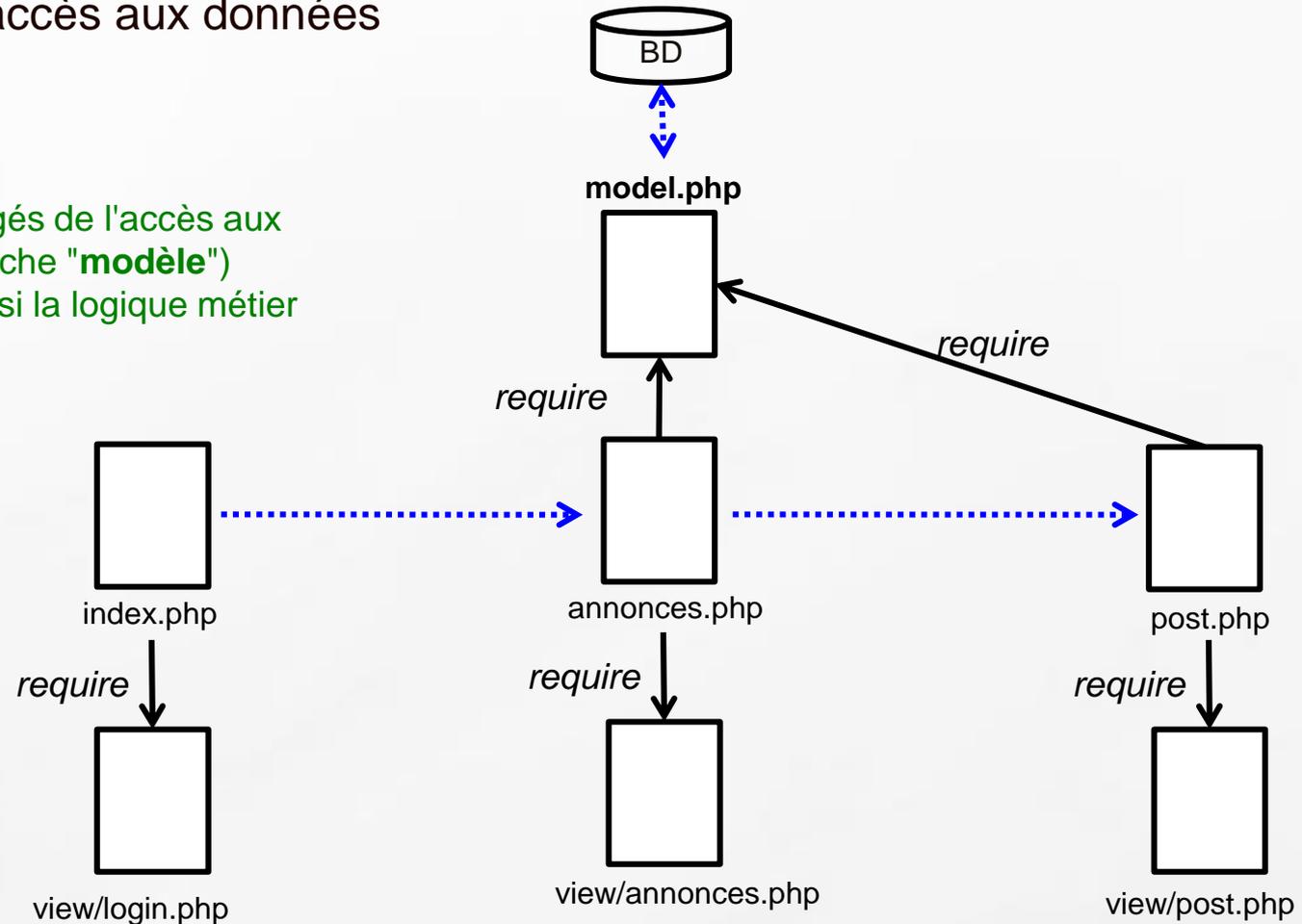
view/post.php

MVC et PHP: exemple du site d'annonces *

Etape 2: Isoler l'accès aux données

Fichiers chargés de l'accès aux données (couche "**modèle**")

- contient aussi la logique métier



- rôle des contrôleurs:
- récupérer les données de la couche modèle
 - appeler la vue permettant d'afficher les données

MVC et PHP: exemple du site d'annonces *

```
<?php

function open_database_connection()
{
    $link = mysqli_connect('localhost', 'root', '', 'jobstage_db');
    return $link;
}

function close_database_connection($link)
{
    mysqli_close($link);
}

function is_user( $login, $password )
{
    $isuser = False ;
    $link = open_database_connection();

    $query= 'SELECT login FROM Users WHERE
login="'.$login.'" and password="'.$password.'";'
    $result = mysqli_query($link, $query );

    if( mysqli_num_rows( $result )
        $isuser = True;

    mysqli_free_result( $result );
    close_database_connection($link);

    return $isuser;
}
```

```
function get_all_posts()
{
    $link = open_database_connection();

    $resultall = mysqli_query($link,'SELECT id, title FROM Post');
    $posts = array();

    while ($row = mysqli_fetch_assoc($resultall)) {
        $posts[] = $row;
    }

    mysqli_free_result( $resultall);
    close_database_connection($link);

    return $posts;
}

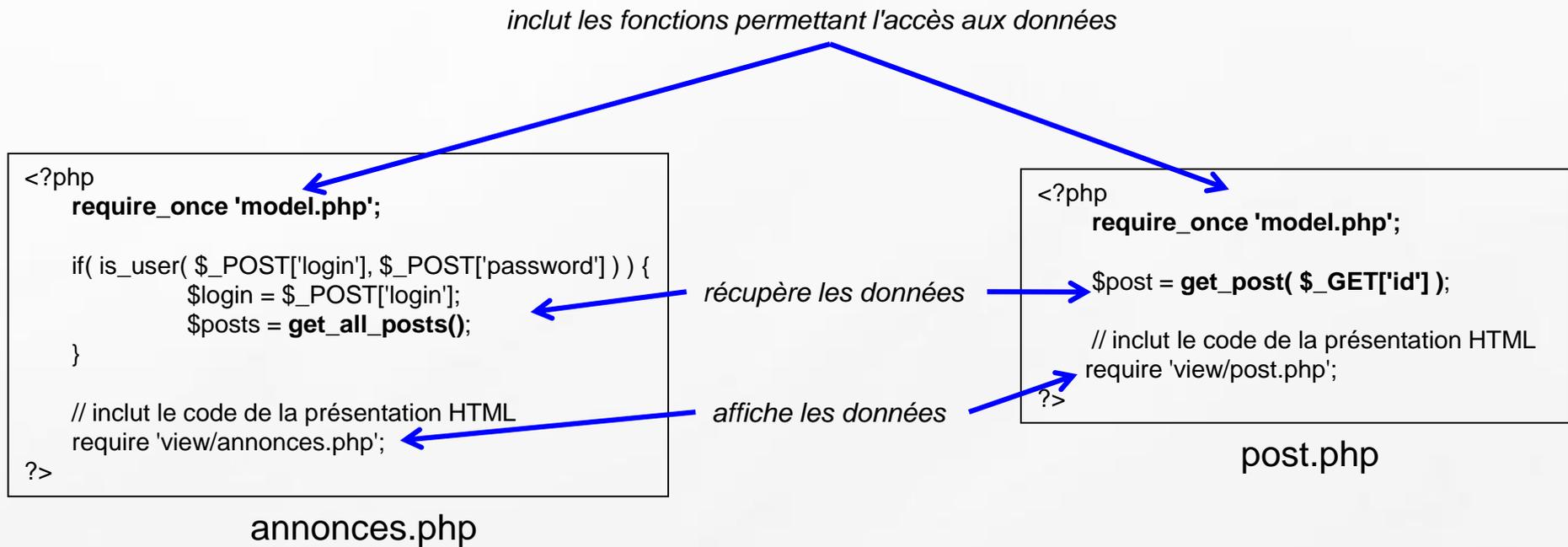
function get_post( $id )
{
    $link = open_database_connection();

    $id = intval($id);
    $result = mysqli_query($link, 'SELECT * FROM Post WHERE
id='.$id );
    $post = mysqli_fetch_assoc($result);

    mysqli_free_result( $result);
    close_database_connection($link);
    return $post;
}
?>
```

model.php

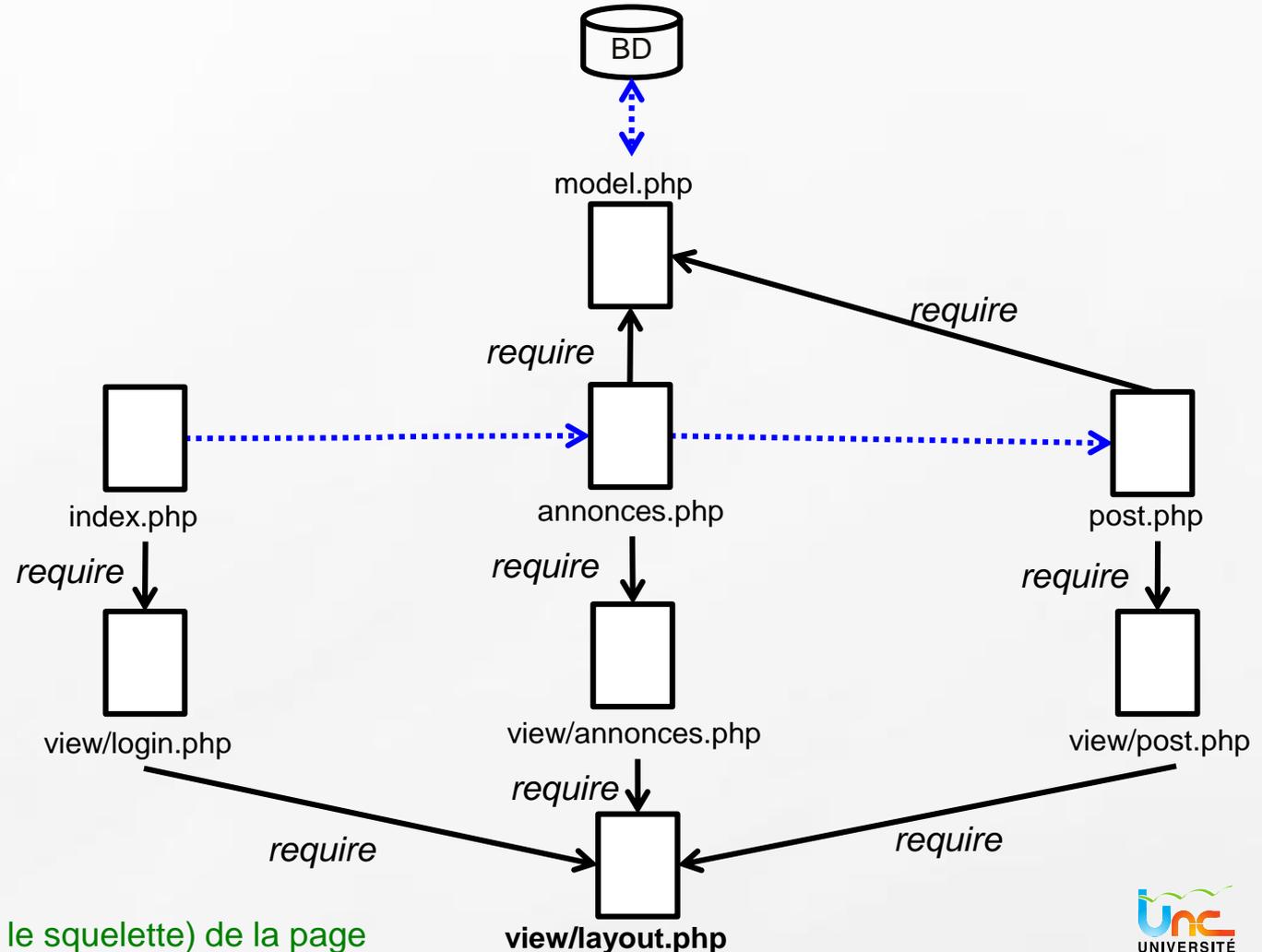
MVC et PHP: exemple du site d'annonces *



*cf Symfony versus PHP pur https://symfony.com/doc/current/introduction/from_flat_php_to_symfony.html

MVC et PHP: exemple du site d'annonces *

- Etape 3: Isoler la mise en page de base de la page (le *layout*)



Fichier factorisant le layout (ie le squelette) de la page

view/layout.php

MVC et PHP: exemple du site d'annonces *

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title><?php echo $title; ?></title>
    <meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
  </head>
  <body>
    <?php echo $content; ?>
  </body>
</html>
```

view/layout.php

squelette de la page :

- un titre
- un contenu

```
<?php $title= 'Exemple Annonces Basic PHP: Connexion'; ?>
```

```
<?php ob_start(); ?>
```

```
<form method="post" action="annonces.php">
```

```
<label for="login"> Votre identifiant </label> :
```

```
<input type="text" name="login" id="login" placeholder="default" maxlength="12" required />
```

```
<br />
```

```
<label for="password"> Votre mot de passe </label> :
```

```
<input type="password" name="password" id="password" maxlength="12" required />
```

```
<input type="submit" value="Envoyer">
```

```
</form>
```

```
<?php $content = ob_get_clean(); ?>
```

```
<?php include 'layout.php'; ?>
```

view/login.php

← initialisation du titre utilisé dans le squelette

initialisation du contenu (le formulaire de login) et stockage en mémoire

← affichage de la page

MVC et PHP: exemple du site d'annonces *

```
<?php
if( !isset( $login ) ){
    header( "refresh:5:url=index.php" );
    echo 'Erreur de login et/ou de mot de passe (redirection
automatique dans 5 sec.)';
    exit;
}
?>

<?php $title= 'Exemple Annonces Basic PHP: Annonces'; ?>

<?php ob_start(); ?>
<p> Hello <?php echo $login; ?> </p>
<h1>List of Posts</h1>
<ul>
    <?php foreach( $posts as $post ) : ?>
        <li>
            <a href="post.php?id=<?php echo $post['id']; ?>">
                <?php echo $post['title']; ?>
            </a>
        </li>
    <?php endforeach ?>
</ul>
<?php $content = ob_get_clean(); ?>

<?php include 'layout.php'; ?>
```

view/annonces.php

1. *initialisation du titre*
2. *initialisation du contenu*
3. *affichage de la page*

```
<?php $title= 'Exemple Annonces Basic PHP: Post'; ?>

<?php ob_start(); ?>
<h1><?php echo $post['title']; ?></h1>
<div class="date"> <?php echo $post['date']; ?> </div>
<div class="body"> <?php echo $post['body']; ?> </div>
<?php $content = ob_get_clean(); ?>

<?php include 'layout.php'; ?>
```

view/post.php

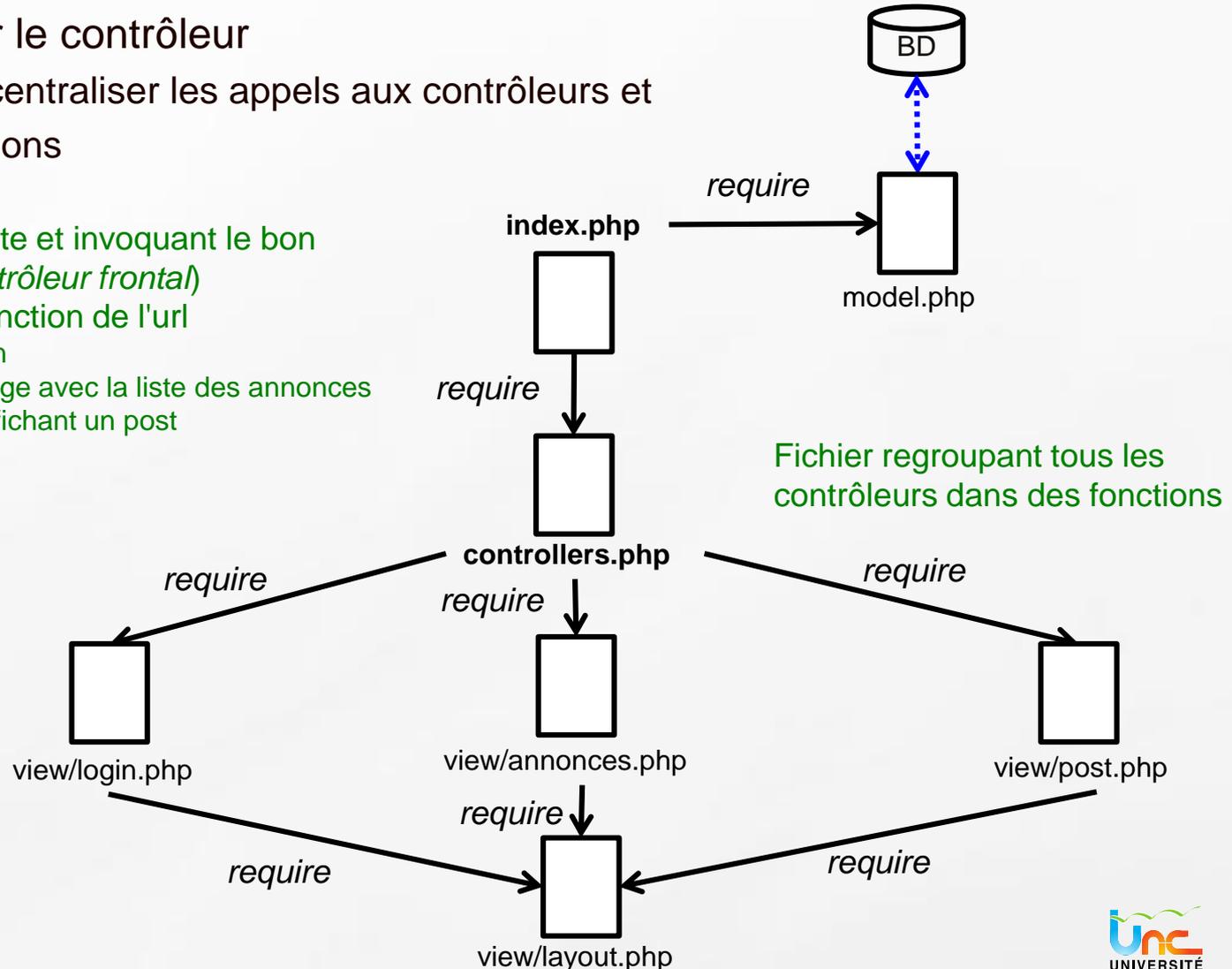
MVC et PHP: exemple du site d'annonces *

Etape 4: Isoler le contrôleur

- permet de centraliser les appels aux contrôleurs et les vérifications

Fichier traitant chaque requête et invoquant le bon contrôleur (appelé aussi *contrôleur frontal*)

- routage des requêtes en fonction de l'url
 - /index.php -> page de login
 - /index.php/annonces -> page avec la liste des annonces
 - /index.php/post -> page affichant un post



*cf Symfony versus PHP pur https://symfony.com/doc/current/introduction/from_flat_php_to_symfony.html

MVC et PHP: exemple du site d'annonces *

```
<?php

function login_action()
{
    require 'view/login.php';
}

function annonces_action( $login,
$password)
{
    if( is_user( $login, $password ) )
        $posts = get_all_posts();
    else
        $login="";

    require 'view/annonces.php';
}

function post_action($id)
{
    $post = get_post($id);
    require 'view/post.php';
}

?>
```

controllers.php

```
<?php

// charge et initialise les bibliothèques globales
require_once 'model.php';
require_once 'controllers.php';

// route la requête en interne
$uri = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);

if ('/JobStagesUNC/index.php' == $uri) {
    login_action();
}
elseif ( '/JobStagesUNC/index.php/annonces' == $uri
    && isset($_POST['login']) && isset($_POST['password']) ){

    annonces_action($_POST['login'], $_POST['password']);
}
elseif ('/JobStagesUNC/index.php/post' == $uri
    && isset($_GET['id'])) {

    post_action($_GET['id']);
}
else {
    header('Status: 404 Not Found');
    echo '<html><body><h1>My Page NotFound</h1></body></html>';
}

?>
```

index.php

extraction du chemin de l'url

*Si index.php
-> contrôleur login*

*Si
index.php/annonces
et login+password
transmis
-> contrôleur blog*

*Si index.php/post
et id post transmis
-> contrôleur post*

*Sinon affichage
erreur 404*

Exercice

- ☐ Intégrer les autres parties du site d'annonces dans cette architecture MVC
 - Modifier le contrôleur frontal
 - Ajouter une vue correspondant à l'enregistrement d'un nouvel utilisateur
 - Ajouter une vue permettant d'ajouter une nouvelle annonce

Sessions PHP et cookies

- Il peut être utile de conserver des informations d'une page sur l'autre lors de la visite d'un utilisateur
 - p.ex. pour se souvenir du login de l'utilisateur connecté, de panier d'achat en cours d'un utilisateur, des dernières pages visitées, etc

- Jusqu'ici, seuls moyens:
 - utiliser des paramètres et les transmettre pages après pages via des liens de navigation ou des formulaires (voire des formulaires cachés)
 - passer par une base de données
 - mécanismes relativement "lourds" d'un point de vue programmation

- Deux autres alternatives
 - les sessions
 - les cookies

Les sessions

- Permet de conserver des informations entre les différentes pages
 - durée de conservation par défaut: 24 min. (modifiable dans le fichier php.ini)
- En PHP, la session est une variable spéciale de type tableau associatif appelée **\$_SESSION**
- Utilisation:
 - commencer chaque page par l'instruction **session_start()** (*avant génération du code HTML*)
 - utiliser la variable **\$_SESSION** comme un tableau associatif classique
 - lorsque l'utilisateur "se déconnecte", il est important de détruire la session
 - p.ex. pour éviter qu'une seconde personne utilisant le même ordinateur ne se fasse passer pour la première personne

```
<?php
    session_start();
    ...
    $_SESSION['login']='root';
    ...
?>
```

```
<?php
    ...
    session_destroy();
    ...
?>
```

Les sessions: un exemple simple

Sans session

```
...  
<a href="msg.php?id=1&login=root">  
    titre message 1  
</a>  
...
```

listeMail.php

```
...  
<?php  
    echo $_GET['login'];  
?>  
...  
<a href="msg.php?id=1&login=$_GET['login']">  
    next page  
</a>  
...
```

msg.php

Avec session

```
<?php  
    session_start();  
    ...  
    $_SESSION['login']='root';  
    ...  
?>  
...  
<a href="msg.php?id=1"> titre message 1 </a>  
...
```

listeMail.php

```
<?php  
    session_start();  
    ...  
    echo $_SESSION['login'];  
    ...  
?>  
...  
<a href="msg.php?id=1"> next page </a>  
...
```

msg.php

Les sessions: exemple du site d'annonces

- Vérifier l'authentification sur toutes les pages du site pour éviter les accès non connectés
 - supprimer les tests dans le contrôleur de la partie "*annonces*" et mettre en place les tests dans le contrôleur frontal
 - utilisation de la fonction `is_user`
 - **vérification et enregistrement de session**
 - création d'une variable pour stocker le type d'erreur et passage en paramètre des contrôleurs
 - supprimer l'affichage du message d'erreur (et de la redirection) dans la vue de la partie "*annonces*" et gérer l'affichage des erreurs dans la vue *layout*

- Généraliser l'affichage d'un message d'accueil "*Connecté en tant que ...*" à toutes les pages du site
 - modifier les contrôleurs en leur passant en paramètre le login de l'utilisateur
 - supprimer l'affichage du message d'accueil dans la vue de la partie "*annonces*" et gérer son affichage dans la vue "*layout*"

- Ajouter la fonctionnalité "Déconnexion" pour fermer une session utilisateur
 - ajouter dans le contrôleur frontal le cas "Déconnexion" via l'url "`/index.php/logout`"
 - ajouter dans la vue "layout" l'affichage d'un lien "Déconnexion" si la personne est connectée

Les sessions: exemple du site d'annonces

```
<?php
// charge et initialise les bibliothèques globales
require_once 'model.php';
require_once 'controllers.php';

// démarrage de la session
session_start();

// récupération du nom de la page demandée
$uri = parse_url($_SERVER['REQUEST_URI'],
PHP_URL_PATH);

// vérification utilisateur authentifié
if( !isset($_SESSION['login']) ) {

if( !isset($_POST['login']) || !isset($_POST['password']) ) {
    $error='not connected';
    $uri='/jobStageUNC/index.php' ;
}
elseif( !is_user($_POST['login'],$_POST['password']) ){
    $error='bad login/pwd';
    $uri='/jobStageUNC/index.php' ;
}
else {
    $_SESSION['login'] = $_POST['login'] ;
    $login = $_SESSION['login'];
}
}
else
    $login = $_SESSION['login'] ;
```

```
// route la requête en interne
if ('/jobStageUNC/index.php' == $uri ) {
    login_action($login,$error);
}
elseif ( '/jobStageUNC/index.php/annonces' == $uri ){
    annonces_action($login,$error);
}
elseif ('/jobStageUNC/index.php/post' == $uri &&
isset($_GET['id'])) {
    post_action($_GET['id'],$login,$error);
}
elseif('/jobStageUNC/index.php/logout' == $uri ) {
    // fermeture de la session
    session_destroy();
    // affichage de la page de connexion
    login_action("", "");
}
else {
    header('Status: 404 Not Found');
    echo '<html><body><h1>My Page Not Found</h1></body>
</html>';
}
?>
```

index.php

Les sessions: exemple du site d'annonces

```
<?php
function login_action($login,$error)
{
    require 'view/login.php';
}

function annonces_action($login,$error)
{
    $posts = get_all_posts();
    require 'view/annonces.php';
}

function post_action($id,$login,$error)
{
    $post = get_post($id);
    require 'view/post.php';
}
?>
```

controllers.php

```
<?php $title= 'Exemple Annonces Basic PHP: Annonces'; ?>

<?php ob_start(); ?>
<p> Hello <?php echo $login; ?> </p>
<h1>List of Posts</h1>
<ul>
    <?php foreach( $posts as $post ) : ?>
        <li>
            <a href="post.php?id=<?php echo $post['id']; ?>">
                <?php echo $post['title']; ?>
            </a>
        </li>
    <?php endforeach ?>
</ul>
<?php $content = ob_get_clean(); ?>

<?php include 'layout.php'; ?>
```

view/annonces.php

Les sessions: exemple du site d'annonces

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <title><?php echo $title; ?></title>
  <meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
</head>
<body>

  <p>
    <?php
      if( isset($error) ){
        switch( $error ) {
          case 'not connected':
            echo 'Veuillez svp vous authentifier';
            break;
          case 'bad login/pwd':
            echo 'Erreur de login et/ou de mot de passe';
            break;
        }
      }
      elseif( isset($login) ) {
        echo 'Connecté en tant que '.$login ;
        echo ' <a href="/jobStageUNC/index.php/logout">Déconnexion</a>';
      }
    ?>
  </p>

  <?php echo $content; ?>
</body>
</html>
```

view/layout.php

Les cookies

- Permet de conserver des informations entre les différentes pages sur de longues périodes et entre plusieurs visites
 - stocké dans un fichier sur la machine du client
- En PHP, les cookies sont une variable spéciale de type tableau associatif appelée **\$_COOKIE**

Utilisation:

- utiliser la fonction **setcookie(\$name,\$value,\$duration)** (*avant toute génération de code HTML*)
- utiliser la variable **\$_COOKIE** comme un tableau associatif classique
- pour détruire un cookie faire **setcookie(\$name,' ', time() - 3600)**

```
<?php
setcookie('login_cookie', 'flouvat', time()+(86400*30))
...
echo $_COOKIE['login_cookie'];
...
?>
```

30 jours

```
<?php
...
setcookie('login_cookie', "", time()-3600)
...
?>
```

Exercice

- Intégrer dans l'application une visualisation des annonces d'emploi mise à disposition par l'API du gouvernement
 - <https://data.gouv.nc/explore/dataset/offres-demploi/api/>
- Donner la possibilité à un utilisateur de « signaler » une annonce
- Intégrer une section « administration » dans l'application permettant de supprimer une annonce ou un utilisateur