

Dans ce TD, nous allons transférer des données (login et mot de passe) en bouclant sur le même script PHP. Il s'agit de vérifier la validité des identifiants à leur saisie (une fois), de les stocker dans des variables de session, puis de les revérifier à chaque itération sans nouvelle saisie.

Page web exemple : <https://eric.univ-lyon2.fr/jdarmont/docs/web/td34controleur.php>

Uniquement au cas où vous ayez des problèmes avec les sessions sur le serveur phpetu :

- dans le répertoire Sites de phpetu, créer un répertoire nommé sessions
- dans votre code, après `session_start()` ; insérer le code `ini_set("session.save_path", "./sessions");`

I. Gabarit d'affichage de résultat et de boucle

- 1 Sur la base du squelette HTML déjà employé aux TD n° 1 et 2¹, créer un fichier `td34vue-msg.tpl.html` contenant dans le corps de la page :
 - un paragraphe contenant un champ `[onshow.message]` ;
 - un paragraphe contenant un lien hypertexte dont la cible est le champ `[onshow.cible]` et le libellé « Retour ».

II. Classe pour la gestion de login/mot de passe

ID
<code>\$login : String</code> <code>\$passwd : String</code>
<code>verif()</code>

- 2 Créer un fichier `td34modele.class.php`. Y placer les balises PHP, puis définir une classe ID contenant deux attributs privés `$login` et `$passwd` initialisés à une valeur au choix, ainsi qu'une méthode publique `verif($param_login, $param_passwd)` permettant de vérifier que les attributs `$this->login` et `$this->passwd` de la classe ID sont tous deux égaux aux paramètres `$param_login` et `$param_passwd`, respectivement. La méthode `verif()` renvoie 1 si le login et le mot de passe sont conformes, 0 sinon.
- 3 Créer un fichier `td34controleur.php`, y placer les balises PHP et inclure les fichiers `tbs_class.php` et `td34modele.class.php`. Créer un objet `$tbs` de classe `clsTinyButStrong` et un objet `$id` de classe ID.
- 4 Toujours dans `td34controleur.php`, affecter à la cible du gabarit `td34vue-msg.tpl.html` la valeur `$_SERVER["PHP_SELF"]`². Ensuite, appeler la méthode `verif()` pour `$id`. Si le résultat est 1, affecter au message du gabarit « Identifiants corrects », sinon « Login ou mot de passe incorrect ». Afficher le gabarit `td34vue-msg.tpl.html`.
- 5 Tester en passant les bons paramètres à `verif()`, puis en indiquant successivement un login et/ou un mot de passe erroné.

III. Gabarit/formulaire de saisie de login/mot de passe

Identification

Login :
 Mot de passe :

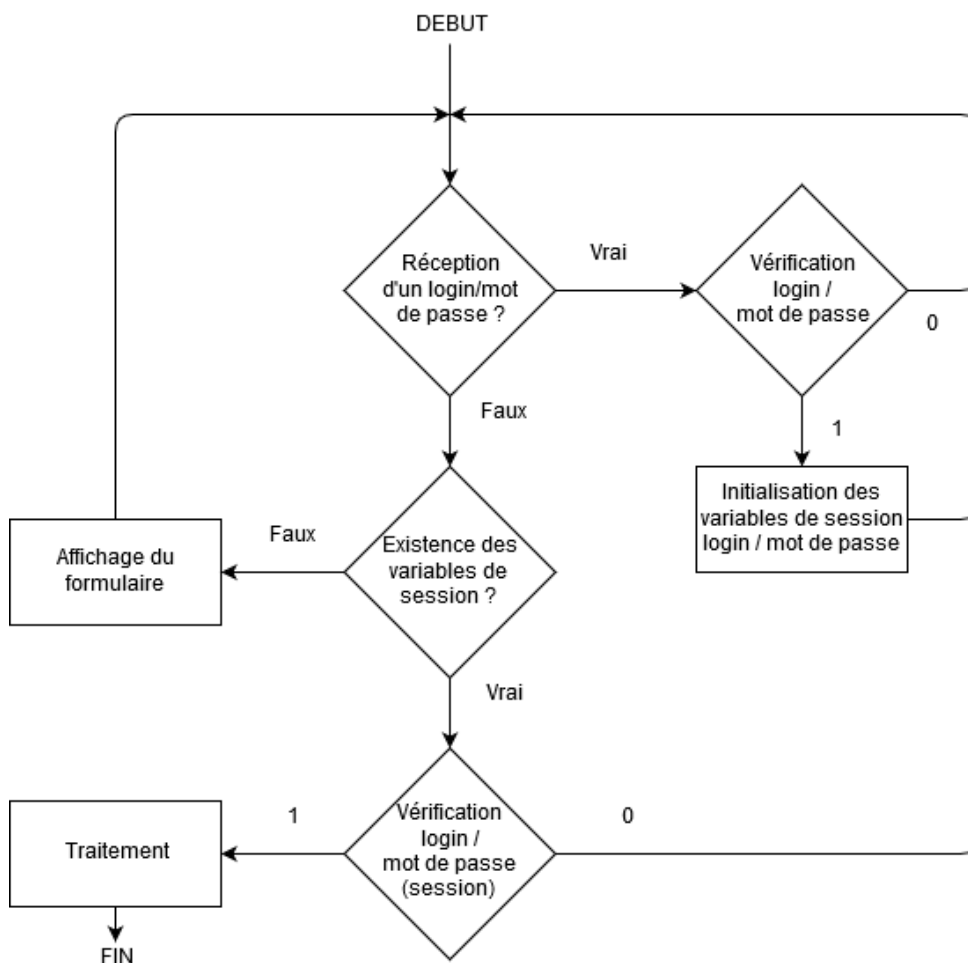
¹ <https://eric.univ-lyon2.fr/jdarmont/docs/web/squelette.html>

² Cette variable système retourne l'URL relative de la page courante, ici `td34controleur.php`. Après chaque action, nous allons donc « boucler » sur `td34controleur.php`. C'est une pratique courante qui minimise le nombre de fichiers PHP.

- 6 Nous allons maintenant remplacer l'initialisation du login et du mot de passe « en dur » par un formulaire de saisie. Créer un fichier gabarit nommé td34vue-form.tpl.html sur la base du squelette HTML¹. Créer dans le corps de la page un formulaire selon les spécifications suivantes.
- La cible du formulaire (attribut action) est [onshow.cible].
 - La méthode de transmission des données est « post ».
 - Tous les champs du formulaire sont regroupés dans un *fieldset* de légende « Identification ».
 - Chaque champ doit faire l'objet d'une description *label* (accessibilité).
 - La liste des champs est indiquée ci-dessous.

ID champ	Nom champ	Intitulé	Type	Particularités
id_log	login	Login	text	
id_pwd	passwd	Mot de passe	password	
id_val			submit	Valeur : Valider

IV. Mise en œuvre d'une session



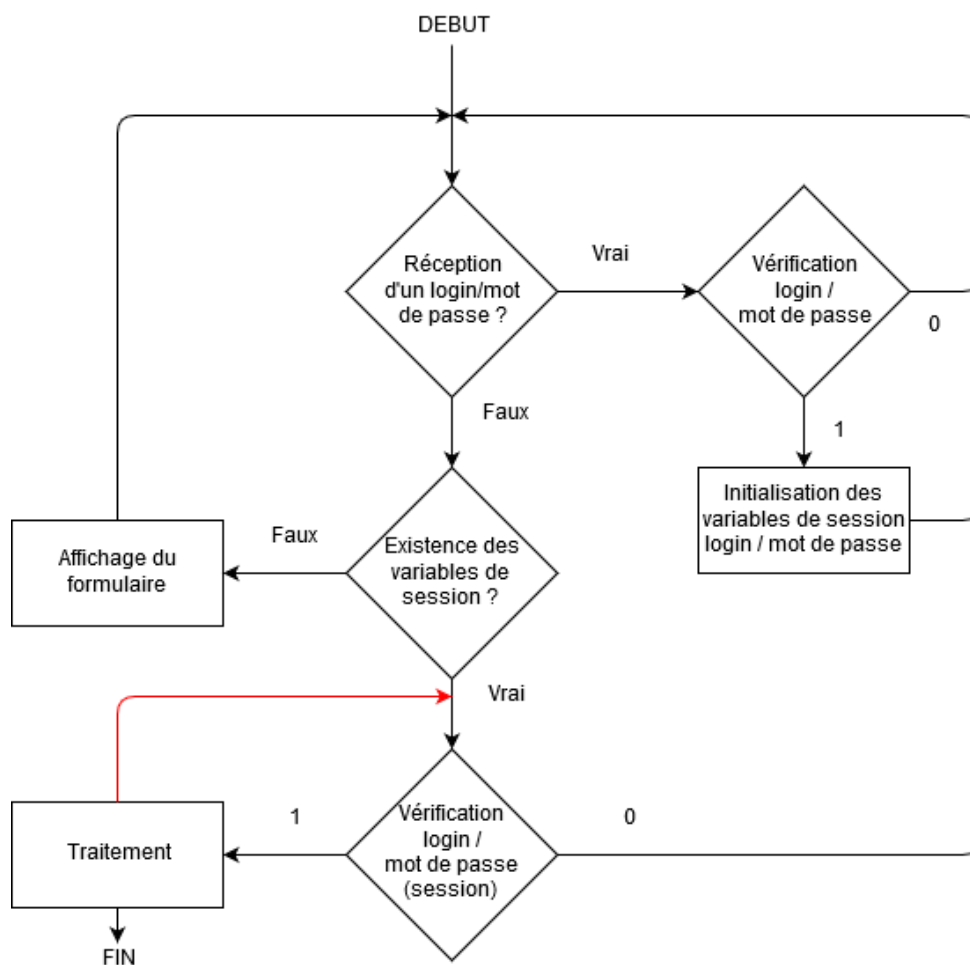
- 7 Dans td34controleur.php, démarrer une session (T81) en tout début de fichier. Après la ligne \$cible = \$_SERVER["PHP_SELF"], commenter le code correspondant à la partie II.
- 8 Après la création de l'objet \$id et l'initialisation de la variable \$cible, si un login a été transmis depuis le formulaire (tester l'existence de \$_POST["login"] à l'aide de la fonction isset() ; il est inutile de faire de même pour le mot de passe car il est transmis en même temps que le login), copier/coller la vérification des identifiants écrite dans la partie II en remplaçant le login et le mot de passe « en dur » par \$_POST["login"] et \$_POST["passwd"], respectivement.

Dans le cas où la vérification est positive (égale à 1), initialiser les variables de session \$_SESSION["login"] et \$_SESSION["passwd"] (T83) à \$_POST["login"] et \$_POST["passwd"], respectivement, puis affecter à \$message la valeur « Identifiants corrects ». Sinon, affecter à \$message la valeur « Login ou mot de passe incorrect ». Que la vérification soit positive ou négative, afficher le gabarit td34vue-msg.tpl.html (affichage des messages).

- 9 Sinon, si les variables de session n'existent pas (tester la non-existence de `$_SESSION["login"]` est suffisant), afficher le gabarit `td34vue-form.tpl.html` (formulaire défini en partie III).
- 10 Sinon, effectuer la vérification des identifiants, cette fois-ci avec les paramètres `$_SESSION["login"]` et `$_SESSION["passwd"]` qui ont été sauvegardés dans la session. Dans le cas où la vérification est positive (`== 1`), affecter à `$message` la valeur « Session OK ». Que la vérification soit positive ou négative, afficher le gabarit `td34vue-msg.tpl.html` (affichage des messages).
- 11 Tester enfin ! Il est normal que le traitement boucle sur le message « Session OK ». Pourquoi ?
- 12 Pour éviter cette boucle infinie, ajouter la commande `session_destroy()`; (T82) quand la vérification est positive.

V. Poussons le bouchon la session !

Jusqu'à maintenant, la session était détruite automatiquement. Nous allons prolonger le traitement dans la session avant de l'interrompre. Pour cela, nous allons paramétrer l'URL courante et utiliser ces paramètres grâce au pseudo-tableau associatif `$_GET[]` (T75).



- 13 Dans le bloc else du fichier `td34controleur.php`, commenter les lignes `$message = "Session OK";` et `session_destroy()`;
- 14 À la suite, si la variable `$_GET["action"]` (action est un paramètre d'URL) n'est pas définie, réinitialiser la variable `$cible` à la concaténation de `$_SERVER["PHP_SELF"]` et de « ?action=suite », et la variable `$message` à « On continue ! ». Afficher le gabarit `td34vue-msg.tpl.html`.

- 15 Sinon, en fonction de la valeur de `$_GET["action"]` (utiliser l'instruction `switch`) :
 - 15.a cas « suite » : réinitialiser la variable `$cible` à la concaténation de `$_SERVER["PHP_SELF"]` et de « `?action=fin` », et la variable `$message` à « Il faut s'arrêter maintenant ». Afficher le gabarit `td34vue-msg.tpl.html` ;
 - 15.b cas « fin » : détruire la session. Réinitialiser la variable `$cible` à `$_SERVER["PHP_SELF"]` et la variable `$message` à « Bye bye ».
- 16 Tester en regardant l'URL à chaque étape. Vérifier la validité du code HTML généré par PHP.

Correction

```
<!DOCTYPE html> <!-- td34vue-msg.tpl.html - Partie I -->
<html lang="fr">
  <head>
    <title>Affichage</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <p>[onshow.message]</p>
    <p>
      <a href="[onshow.cible]">Retour</a>
    </p>
  </body>
</html>
```

<?php // td34modele.class.php - Partie II

```
class ID {
  private $login = "darmont";
  private $passwd = "3sMp7XZ6y7Zy64Nmcn7S";
  // Essayer https://www.motdepasse.xyz c'est l'adopter !

  public function verif($param_login, $param_passwd) {
    if (($this->login == $param_login)
        && ($this->passwd == $param_passwd))
      return 1;
    else
      return 0;
  }
}
?>
```

<!DOCTYPE html> <!-- td34vue-form.tpl.html - Partie III -->

```
<html lang="fr">
  <head>
    <title>Formulaire</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <form action="[onshow.cible]" method="post">
      <fieldset>
        <legend>Identification</legend>
        <label for="id_log">Login</label> :
          <input type="text" id="id_log" name="login" />
        <label for="id_pwd">Mot de passe</label> :
          <input type="password" id="id_pwd" name="passwd" />
        <label for="id_val"></label>
          <input type="submit" id="id_val" value="Valider" />
      </fieldset>
    </form>
  </body>
</html>
```

```
<?php // td34controleur.php - Parties II, IV et V
```

```
// Session
session_start();

// Appel aux fichiers des classes
require("tbs_class.php");
require("td34.class.php");

// Création des objets
$tbs = new clsTinyButStrong;
$id = new ID;

// Traitement
// On boucle toujours sur td34controleur.php
$cible = $_SERVER["PHP_SELF"];

/* Partie II
if ($id->verif("darmont", "3sMp7XZ6y7Zy64Nmcn7S"))
    $message = "Identifiants corrects";
else
    $message = "Login ou mot de passe incorrect";
    $tbs->LoadTemplate("td34msg.tpl.html");
    $tbs->Show(); */

// Partie IV

// Si un login a été transmis via le formulaire
if (isset($_POST["login"])) { // Transmission d'un login
    // Vérification des identifiants
    if ($id->verif($_POST["login"], $_POST["passwd"])) { // Vérification +
        $_SESSION["login"] = $_POST["login"];
        $_SESSION["passwd"] = $_POST["passwd"];
        $message = "Identifiants corrects";
    } else // Vérification -
        $message = "Login ou mot de passe incorrect";
    // Affichage de la page de message
    $tbs->LoadTemplate("td34msg.tpl.html");
    $tbs->Show();
} elseif (!isset($_SESSION["login"])) { // Variables de session initialisées
    // Affichage du formulaire
    $tbs->LoadTemplate("td34form.tpl.html");
    $tbs->Show();
} else {
    // Vérification des identifiants
    if ($id->verif($_SESSION["login"], $_SESSION["passwd"])) {
        /* Partie IV
        $message = "Session OK";
        session_destroy(); */
        /* Questions 11-12 : Sans session_destroy(), une fois les variables
        de session correctes, comme on ne les modifie pas, on retombe à
        chaque fois dans le dernier cas (else) et la vérification est
        toujours positive. */
```

```

// Partie V
if (!isset($_GET["action"])) {
    $cible = $_SERVER["PHP_SELF"] . "?action=suite";
    $message = "On continue !";
    $tbs->LoadTemplate("td34vue-msg.tpl.html");
    $tbs->Show();
} else {
    switch($_GET["action"]) {
        case "suite":
            $cible = $_SERVER["PHP_SELF"] . "?action=fin";
            $message = "Il faut s'arrêter maintenant";
            $tbs->LoadTemplate("td34vue-msg.tpl.html");
            $tbs->Show();
            break;
        case "fin":
            session_destroy();
            $cible = $_SERVER["PHP_SELF"];
            $message = "Bye bye";
            break;
    }
}
// Affichage de la page de message
$tbs->LoadTemplate("td34msg.tpl.html");
$tbs->Show();
}
?>

```