

td2PhpBDAjax

```
<select name="os" id="os" >
    <option value="-1">Sélectionnez un SE</option>
    <!-- <option value="win">WINDOWS</option>
    <option value="lin">LINUX</option>
    <option value="mac">MAC OS</option> -->
</select>
```

form.js

```
/****** Question 1 : Ajout fonction pour afficher liste des versions *****/
function genererListeOs() {
    // on initialise la valeur du champs ayant pour id typeDemande
    // avec la lettre 'o'
    $("#typeDemande").val('o');
    $("#os").empty();
    $("#os").append('<option value="-1">Sélectionnez un SE</option>');
    $.ajax({
        url: 'php/controleur.php',
        data: $("#typeDemande").serialize(), // on envoie typeDemande='o'
        type: 'GET',
        dataType: 'json',
        success: // si la requete fonctionne, mise à jour de la liste des versions
            function (objetJson) {
                $.each(objetJson, function (index, ligne) {
                    // ligne contient un objet json de la forme
                    // {"idOs" : "id de l'os"},
                    // {"nomOs" : "nom de l'os"}
                    $("#os").append('<option value="' + ligne.idOs + ">' + ligne.nomOs + '</option>');
                });
            },
        error:
            function (xhr, status, error) {
                console.log("param : " + JSON.stringify(xhr));
                console.log("status : " + status);
                console.log("error : " + error);
            }
    });
}
```

```

function genererListeVersions()
{
    var idOs = $(this).val(); // on récupère la valeur de la clef primaire correspondant à l'os
    // changement de la valeur du type de demande pour faire appel à la fonction majListeOs
    $("#typeDemande").val('v');

    $("#version").empty(); // on vide la liste des départements
    $("#version").append('<option value="-1">Choisissez une version</option>');

    if (idOs != '-1') { // si l'os sélectionné existe (pas le "choisissez un os")
        // appel à la page majListeOs via ajax
        $.ajax({
            url: 'php/controleur.php',
            data: $("#formulaire").serialize(),
            // typeDemande='r' &
            // idDepartement=-1 &
            // idVille=-1 &
            // idRegion=idDeLaRegionSelectionnée
            type: 'GET',
            dataType: 'json',
            success: // si la requete fonctionne, mise à jour de la liste des versions
                function (objetJson) {
                    $.each(objetJson, function (index, ligne) {
                        // ligne contient un objet json de la forme
                        // {"idVersion" : "id de la version"},
                        // {"nomVersion" : "nom de la version"}
                        $("#version").append('<option value="' + ligne.idVersion + "'>' + ligne.nomVersion +
'</option>');
                    });
                },
            error:
                function (xhr, status, error) {
                    console.log("param : " + JSON.stringify(xhr));
                    console.log("status : " + status);
                    console.log("error : " + error);
                }
        });
    }
}
/
****/

```

***** Question 2 : Ajout fonction pour gérer le login au click *****

```
function verifierLoginBoutton() {
```

```
    // appel du script controleur.php via ajax
```

```
    $.ajax({
```

```
        url: 'php/getLogin.php',
```

```
        data: $("#formulaire").serialize(), // serialisation des données du formulaire (toutes les balises ayant un  
        champs "name"
```

```
        type: 'POST',
```

```
        dataType: 'json',
```

```
        success: // si la requete fonctionne, mise à jour de la couleur de pastille
```

```
            function (objetJson) {
```

```
                if (objetJson == true) {
```

```
                    alert("Le login existe déjà, choisissez en un autre!");
```

```
                } else {
```

```
                    alert("Le login est nouveau");
```

```
                }
```

```
            },
```

```
            error:
```

```
                function (xhr, status, error) {
```

```
                    console.log("param : " + JSON.stringify(xhr));
```

```
                    console.log("status : " + status);
```

```
                    console.log("error : " + error);
```

```
                }
```

```
            });
```

```
    }
```

```
    /
```

```
*****
```

```
**/
```

***** Question 3 : Ajout fonction pour gérer le login au changement *****/

```
function verifierLoginChangement() {
```

```
    // appel du script controleur.php via ajax
```

```
    $.ajax({
```

```
        url: 'php/getLogin.php',
```

```
        data: $("#formulaire").serialize(), // serialisation des données du formulaire (toutes les balises ayant un  
        champs "name"
```

```
        type: 'POST',
```

```
        dataType: 'json',
```

```
        success: // si la requete fonctionne, mise à jour de la couleur de pastille
```

```
            function (objetJson) {
```

```
                $("#login").next("span").remove();
```

```
                if (objetJson == true) {
```

```
                    $("<span style = 'color:red; margin-left:440px'><b>Ce login existe  
déjà</b></span>").insertAfter("#login");
```

```
                } else {
```

```
                    $("<span style = 'color:green; margin-left:440px'><b>Ce nouveau login est  
correct</b></span>").insertAfter("#login");
```

```
                }
```

```
            },
```

```
            error:
```

```
                function (xhr, status, error) {
```

```
                    console.log("param : " + JSON.stringify(xhr));
```

```
                    console.log("status : " + status);
```

```
                    console.log("error : " + error);
```

```
                }
```

```
        });
```

```
    }
```

```
    /
```

```
*****  
**/
```

```
$(document).ready(function ()
{
    genererListeOs();

    // gestion liste deroulante
    $("#os").change(genererListeVersions);

    /******* Question 2 : Ajout fonction pour gérer le login avec bouton inscription
    *****/
    //gestion login
    $("#inscription").click(verifierLoginBoutton);

    /******* Question 3 : Ajout fonction pour gérer le login avec changement
    *****/
    //gestion login
    $("#login").change(verifierLoginChangement);
    $(document).on("change", "#login", verifierLoginChangement);
});
```

config.inc

```
<?php
/***** Serveur lycée *****/
/*define("SERVEURBD","172.18.58.14");
define("LOGIN","snir");
define("MOTDEPASSE","snir");
define("NOMDELABASE","exempleAjax");*/

/***** Serveur maison *****/
define("SERVEURBD","localhost");
define("LOGIN","root");
define("MOTDEPASSE","raspberry");
define("NOMDELABASE","exempleAjax");

?>
```

getLogin.php

```
<?php
$login = $_POST['login'];
require_once '../inc/fonctions.inc';
verifierLogin($login);
?>
```

```
<?php
```

controleur.php

```
$source=$_GET['typeDemande'];
require_once '../inc/fonctions.inc';
switch ($source){
    case 'o':getListeOs(); break;
    case 'v':getListeVersionsFromIdOs($_GET['os']);break;
}
?>
```

fonctions.inc

```
<?php
```

```
require_once 'config.inc';
```

```
function connexionBD() {  
    try {  
  
        $bdd = new PDO('mysql:host=' . SERVEURBD . ';dbname=' . NOMDELABASE, LOGIN,  
MOTDEPASSE);  
        } catch (Exception $ex) {  
            die('<br />Pb connexion serveur BD : ' . $ex->getMessage());  
        }  
        return $bdd;  
    }  
}
```

```
function getListeOs() {  
    // connexion BD  
    $bdd = connexionBD();  
  
    $requete = $bdd->query("SELECT * FROM systeme ORDER BY nomSysteme;");  
  
    $tabOs = array();  
  
    while ($tab = $requete->fetch()) {  
    // ajout d'une case dans le tableau  
    // la case est elle-même un tableau contenant 2 champs : idOs, nomOs  
        array_push($tabOs, array('idOs' => $tab['idSysteme'], 'nomOs' => $tab['nomSysteme']));  
    }  
  
    $requete->closeCursor();  
    //on previent qu'on repond en json  
    header('Content-Type: application/json;charset=utf-8');  
    // envoyer les données au format json  
    echo json_encode($tabOs);  
}
```

```
function getListeVersionsFromIdOs($os) {  
    // connexion BD  
    $bdd = connexionBD();  
    $requete = $bdd->prepare("SELECT idVersion, nomVersion FROM version WHERE idSysteme = :idos  
ORDER BY nomVersion;");  
    $requete->bindParam(":idos", $os);  
    $requete->execute() or die(print_r($requete->errorInfo()));  
  
    $tabVersion = array();  
  
    while ($tab = $requete->fetch()) {  
    // ajout d'une case dans le tableau  
    // la case est elle-même un tableau contenant 2 champs : idVersion, nomVersion  
        array_push($tabVersion, array('idVersion' => $tab['idVersion'], 'nomVersion' => $tab['nomVersion']));  
    }  
  
    $requete->closeCursor();  
    //on previent qu'on repond en json  
    header('Content-Type: application/json;charset=utf-8');  
    // envoyer les données au format json
```

```

    echo json_encode($tabVersion);
}

/* * ***** Question 2 : Vérification du login
***** */

function verifierLogin($login) {

//Connexion à la BDD
    $bdd = connexionBD();

// Recherche du login dans la table user
    $requete = $bdd->prepare("SELECT * FROM user WHERE login = :log;");
    $requete->bindParam(":log", $login);
    $requete->execute() or die(print_r($requete->errorInfo()));

// Comptage du nombre de résultats
    $nbLigne = $requete->rowCount();

    if ($nbLigne == 0) {          // Le login n'est pas présent dans la table user
        $retour = false;        // Il faut retourner false
    } else {                     // Le login est présent dans la table user
        $retour = true;         // Il faut retourner true
    }

    $requete->closeCursor();

// On prévient qu'on répond en json
    header('Content-Type: application/json;charset=utf-8');
// Envoyer les données au format json
    echo json_encode($retour);
}

/* * ***** */
?>

```