

420-KEF – Laboratoire 1 – 5 pts (individuel)

Enseignant: Nicolas Chourot

Département d'informatique

Cegep Lionel-Groulx

Objectifs

Manipulation d'une table d'une base de données Sql Xpress

Bâtir un « Wrapper » pour l'accès CRUD à une table

Bâtir les formulaires Web pour :

Créer un enregistrement

Lister les enregistrements

Modifier un enregistrement

Effacer un enregistrement

Énoncé

Monter un site Web client/serveur pour gérer une liste de personnes. Servez-vous du code fourni et par la suite ajouter un champ Avatar.

Livrable :

Répertoire compressé pour uniquement votre nom.zip contenant le fichier de solution et le répertoire du projet le 9 février avant minuit dans la boîte de remise prévue.

Prénom:

Nom:

Téléphone:

Code postal:

Naissance:

Sexe: ☐ masculin
☐ féminin

État civil: ☐ célibataire
☐ marié
☐ conjoint de fait
☐ séparé
☐ veuf



Prénom:

Nom:

Téléphone:

Code postal:

Naissance:

Sexe: ☐ masculin
☐ féminin

État civil: ☐ célibataire
☐ marié
☐ conjoint de fait
☐ séparé
☐ veuf

February 2015

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28




```

}
function ConstrainToAlpha(event) {
    var textBox = (window.event ? window.event.srcElement : event.srcElement || event.target);
    var car = "";
    var code = 0;
    var valide = "";
    var constrain = false;
    for (var c = 0; c < textBox.value.length; c++) {
        car = textBox.value.substr(c, 1);
        if (!CharIsAlpha(car)) {
            if (!constrain)
                alert("Lettres de 'a' a 'z' seulement!");
            constrain = true;
        }
        else {
            valide = valide + car;
        }
    }
    if (constrain)
        textBox.value = valide;
}

function CharIsDigit(c) {
    var digits = "0123456789";
    return (digits.indexOf(c, 0) != -1);
}

function ConstrainToDigit(event) {
    var textBox = (window.event ? window.event.srcElement : event.srcElement || event.target);
    var car = "";
    var code = 0;
    var valide = "";
    var constrain = false;
    for (var c = 0; c < textBox.value.length; c++) {
        car = textBox.value.substr(c, 1);
        if (!CharIsDigit(car)) {
            //if (!constrain)
            //    alert("Chiffre de '0' a '9' seulement!");
            constrain = true;
        }
        else {
            valide = valide + car;
        }
    }
    if (constrain)
        textBox.value = valide;
}

// cette fonction change la couleur de fond d'un input incomplet
function HighLiteIncompleteMaskedInput(event) {
    var textBox = (window.event ? window.event.srcElement : event.srcElement || event.target);
    var masque = textBox.getAttribute("alt");
    mp = textBox.value.length;
    if (textBox.value.length < masque.length)
        textBox.style.backgroundColor = "#FCC";
    else
        textBox.style.backgroundColor = "#FFF";
}

```

```

// cette fonction change la couleur de fond d'un input vide
function HighLiteEmptyInput(event) {
    var textBox = (window.event ? window.event.srcElement : event.srcElement || event.target);
    mp = textBox.value.length;
    if (textBox.value.length == 0)
        textBox.style.backgroundColor = "#FCC";
    else
        textBox.style.backgroundColor = "#FFF";
}

function KeyIsDigit(key) {
    return ((48 <= key) && (key <= 57));
}
function KeyIsLetter(key) {
    return ((65 <= key) && (key <= 90));
}
function KeyIsLegalControl(key) {
    return ((key == 8) || (key == 9) || (key == 13) || (key == 16) || (key == 27));
}

function Valide_Masque(event) {
    var key = (window.event ? window.event.keyCode : event.keyCode || event.which);
    var textBox = (window.event ? window.event.srcElement : event.srcElement || event.target);
    var masque = textBox.getAttribute("alt");
    mp = textBox.value.length;
    charTyped = String.fromCharCode(key);
    if (KeyIsLegalControl(key)) {
        if ((mp > 0) && (key == 8)) { // backspace
            mp--;
            if ((masque[mp] != "C") && (masque[mp] != "#"))
                mp--;
            textBox.value = textBox.value.substr(0, mp);
            return false;
        }
        else {
            textBox.value = textBox.value.toUpperCase();
            return true;
        }
    }
    while ((mp < masque.length) && (masque[mp] != "C") && (masque[mp] != "#")) {
        textBox.value += masque[mp];
        mp++;
    }
    current_pos = mp;
    if (current_pos < masque.length) {
        if ((masque[current_pos] == "C") && (!KeyIsLetter(key)))
            return false;
        if ((masque[current_pos] == "#") && (!KeyIsDigit(key)))
            return false;
    }
    else
        return false;
    textBox.value = textBox.value.toUpperCase();
    return true;
}

function Post_Check_Masque(event) {
    var textBox = (window.event ? window.event.srcElement : event.target);

```

```

var masque = textBox.getAttribute("alt");
var value = textBox.value;
var constrain = false;
var valide_position = 0;
for (position = 0; position < textBox.value.length; position++) {
    if (masque.charAt(position) == "C") {
        if (!CharIsAlpha(value.substr(position, 1)))
            constrain = true;
    }
    else {
        if (masque.charAt(position) == "#") {
            if (!CharIsDigit(value.substr(position, 1)))
                constrain = true;
        }
        else
            if (masque.charAt(position) != value.charAt(position))
                constrain = true;
    }
    if (constrain)
        break;
    else
        valide_position++;
}
if (constrain) {
    textBox.value = value.substr(0, valide_position);
}
else {
    if (value.length > masque.length) {
        textBox.value = value.substr(0, masque.length);
    }
}
textBox.value = textBox.value.toUpperCase();
var valide = value.length == masque.length;
return (valide);
}

```

```

var cellsIdPrefix = "cell_";

```

```

function CellAppendChild(row, col, object) {
    // extraire le pointeur sur la cellule (row, col)
    target = document.getElementById(cellsIdPrefix + row + col);
    if (target) {
        target.appendChild(object);
    }
}

```

```

function CellSetText(row, col, text) {
    // extraire le pointeur sur la cellule (row, col)
    target = document.getElementById(cellsIdPrefix + row + col);
    if (target) {
        target.innerHTML = text;
    }
}

```

```

// création de tableau qui recevra les controles du formulaire
// pour inserer du texte dans une cellule il suffira
// d'utiliser la fonction CellSetText(row, col, value)
// pour inserer un controle dans une cellule il suffira
// d'utiliser la fonction CellAppendChild(row, col, object)
function BuildTable(rows, cols) {

```

```

var tableObject = document.createElement("table");
for (row = 0; row < rows; row++) {
    var rowObject = document.createElement("tr");
    for (col = 0; col < cols; col++) {
        var cellObject = document.createElement("td");
        cellObject.id = cellsIdPrefix + row + col;
        rowObject.appendChild(cellObject);
    }
    tableObject.appendChild(rowObject);
}
return tableObject;
}

function AddInputText(row, caption, fieldName, className) {
    // ajout de caption dans la cellule (row, 0)
    CellSetText(row, 0, caption);
    //Création d'un élément input :
    //<input type="text" class="className" id="fieldName" />
    inputObject = document.createElement("input");
    inputObject.setAttribute("id", fieldName);
    inputObject.setAttribute("name", fieldName);
    inputObject.setAttribute("type", "text");
    inputObject.setAttribute("class", className);
    inputObject.id = fieldName;
    // ajout du de l'objet input dans la cellule (row, 1)
    CellAppendChild(row, 1, inputObject);
}

function AddMaskedInputText(row, caption, fieldName, mask) {
    // ajout de caption dans la cellule (row, 0)
    CellSetText(row, 0, caption);
    //Création d'un élément input :
    //<input type="text" alt="mask" class="masked" id="fieldName" />
    inputObject = document.createElement("input");
    inputObject.setAttribute("id", fieldName);
    inputObject.setAttribute("name", fieldName);
    inputObject.setAttribute("type", "text");
    inputObject.setAttribute("alt", mask);
    inputObject.setAttribute("class", "masked");
    inputObject.id = fieldName;
    CellAppendChild(row, 1, inputObject);
}

// Cette fonction construit un groupe de boutons radio HTML 5
// Chaque boutons auront le format suivant
// <div class="radio">
//     <input type="radio" id="name_X" name="name" />
//     <label for="name_X"> arguments[X] </label>
// </div>
function AddRadioButtonGroup(row, caption, name) {
    // ajout de caption dans la cellule (row, 0)
    CellSetText(row, 0, caption);

    // parcourt des arguments qui suivent row, caption et name
    // exemple: si on fait l'appel suivant
    // AddRadioButtonGroup(row, caption, name, "masculin", "féminin")
    // arguments[0] = row, arguments[1] = caption, arguments[2] = name
    // arguments[3]="masculin" et arguments[4]="féminin"
    for (i = 3; i < arguments.length; i++) {
        // Création d'un div conteneur du bouton radio
        divObject = document.createElement("div");
    }
}

```

```

divObject.setAttribute("class", "radio");
// alternance de couleur de fond
if (i % 2)
    divObject.style.backgroundColor = "#eee";
else
    divObject.style.backgroundColor = "#ddd";
// Création du bouton radio
radioObject = document.createElement("input");
radioObject.type = "radio";
radioObject.name = name;
radioObject.value = i - 3;
radioObject.id = name + "_" + (i - 3);
if (i == 3) {
    // premier bouton du groupe
    // servira éventuellement à repérer un groupe
    // afin de déterminer si un choix a été fait
    radioObject.setAttribute("class", "RadioButtonGroup");
}

// Création du label associé au bouton radio
labelObject = document.createElement("label");
labelObject.setAttribute("for", radioObject.id);
labelObject.appendChild(document.createTextNode(arguments[i]));

// insertion du bouton radio et de son label dans le div
divObject.appendChild(radioObject);
divObject.appendChild(labelObject);

// insertion du div dans la cellule (row,2)
CellAppendChild(row, 1, divObject);
}
}
function AddSubmitButton(row, caption) {
    buttonObject = document.createElement("button");
    buttonObject.innerHTML = caption;
    CellAppendChild(row, 1, buttonObject);
    buttonObject.onclick = function () {
        return CheckForEmptyInput();
    };
}
function CheckForEmptyInput() {
    var canSubmit = true;
    var inputObjects = document.getElementsByTagName("input");
    for (i = 0; i < inputObjects.length; i++) {
        mask = inputObjects[i].getAttribute("alt");
        if (mask) {
            if (inputObjects[i].value.length < mask.length) {
                canSubmit = false;
            }
        }
        else
            if (inputObjects[i].value.length == 0) {
                inputObjects[i].style.backgroundColor = "#FCC";
                canSubmit = false;
            }
    }
    // chercher tous les group de boutons radio
    var radioButtonGroups = document.getElementsByClassName("RadioButtonGroup");

```



```

for (i = 0; i < radioButtonGroups.length; i++) {
    // chercher tous les boutons du même name
    var buttons = document.getElementsByName(radioButtonGroups[i].name);
    oneButtonIsChecked = false;
    // déterminer si un choix a été fait dans le groupe
    for (b = 0; b < buttons.length; b++) {
        if (buttons[b].checked)
            oneButtonIsChecked = true;
    }
    // changer la couleurs des labels du groupe de boutons
    for (b = 0; b < buttons.length; b++) {
        // le label correspondant au bouton on le même parent
        labelObjects = buttons[b].parentNode.getElementsByTagName("label");
        // en principe il n'y a qu'un seul label qui suit le bouton radio
        if (!oneButtonIsChecked) {
            labelObjects[0].style.color = "#F00";
            canSubmit = false;
        }
        else
            labelObjects[0].style.color = "#000";
    }
}
return canSubmit;
}

function BuildForm(targetFormID) {
    // création du div qui englobe le formulaire
    divObject = document.createElement("div");
    divObject.setAttribute("class", "main");
    divObject.appendChild(BuildTable(9, 2));
    document.getElementById(targetFormID).appendChild(divObject);
    // création des controles
    AddInputText(0, "Prénom:", "Prenom", "ident");
    AddInputText(1, "Nom:", "Nom", "ident");

    AddMaskedInputText(2, "Téléphone:", "Telephone", "(###) ###-####");
    AddMaskedInputText(3, "Code postal:", "CodePostal", "C#C #C#");
    AddMaskedInputText(4, "Naissance:", "Naissance", "####-##-##");
    document.getElementById("Naissance").className = " masked datepicker";

    AddRadioButtonGroup(5, "Sexe:", "Sexe", "masculin", "féminin");
    AddRadioButtonGroup(6, "État civil:", "Etatcivil", "célibataire", "marié", "conjoint de fait",
"séparé", "veuf");
    // la rangée 7 est volontairement sautée
    AddSubmitButton(8, "Ajouter...");
    // installation des "delegates" pour les inputs de classe "ident"
    var inputObjects = document.getElementsByClassName("ident");
    for (i = 0; i < inputObjects.length; i++) {
        inputObjects[i].onkeyup = function () { ConstrainToAlpha(event); };
        inputObjects[i].onblur = function () { HighLiteEmptyInput(event); };
    }
    // installation des "delegates" pour les inputs de classe "masked"
    var maskedInputs = document.getElementsByClassName("masked");
    for (i = 0; i < maskedInputs.length; i++) {
        maskedInputs[i].onkeydown = function () { return Valide_Masque(event); };
        maskedInputs[i].onkeyup = function () { Post_Check_Masque(event); };
        maskedInputs[i].onblur = function () { Post_Check_Masque(event); };
        HighLiteIncompleteMaskedInput(event);
    }
}

```

```

    }
</script>
<style>
    table, td {
        border:0px solid black;
        border-collapse:collapse;
        width:inherit;
        padding:5px;

    }
    tr > td:first-child {
        text-align:right;
        vertical-align:text-top;
    }
    .main {
        width:300px;
        margin:auto;
        border:1px solid black;
    }
    .radio {
        width: 130px;
    }
    input.ident, input.masked {
        width:125px;
    }
    div.ui-datepicker{
        font-size:10px;
    }
</style></head>
<body id="body">
    <form id="form1" method="get" action="CreateUser.aspx">
    <div>
        <script type="text/javascript">
            BuildForm("form1");
        </script>
        <asp:Panel ID="PN_GridView" runat="server"></asp:Panel>
    </div>
    </form>
</body>
</html>

```

```

public class PersonnesTable:SqlExpressUtilities
{
    public long ID { get; set; }
    public String Prenom { get; set; }
    public String Nom { get; set; }
    public String Telephone { get; set; }
    public DateTime Naissance { get; set; }
    public String CodePostal { get; set; }
    public int Sexe { get; set; }
    public int EtatCivil { get; set; }

    public PersonnesTable(String connexionString, System.Web.UI.Page Page) :
        base(connexionString, Page)
    {
        :base(connexionString, Application, Session)
    }
}

```

```

        SQLTableName = "PERSONNES";
    }
    public override void GetValues()
    {
        ID = long.Parse(FieldsValues[0]);

        Prenom = FieldsValues[1];

        Nom = FieldsValues[2];

        Telephone = FieldsValues[3];

        CodePostal = FieldsValues[4];

        Naissance = DateTime.Parse(FieldsValues[5]);

        Sexe = int.Parse(FieldsValues[6]);
        ContentDelegates[6] = ContentDelegateSexe;

        EtatCivil = int.Parse(FieldsValues[7]);
        ContentDelegates[7] = ContentDelegateEtatCivil;
    }

    System.Web.UI.WebControls.WebControl ContentDelegateSexe()
    {
        Label lbl = new Label();
        if (Sexe == 0)
            lbl.Text = "Masculin";
        else
            lbl.Text = "Féminin";
        return lbl;
    }

    System.Web.UI.WebControls.WebControl ContentDelegateEtatCivil()
    {
        Label lbl = new Label();
        switch (EtatCivil)
        {
            case 0: lbl.Text = "Célibataire"; break;
            case 1: lbl.Text = "Marié(e)"; break;
            case 2: lbl.Text = "conjoint(e) de fait"; break;
            case 3: lbl.Text = "Séparé(e)"; break;
            case 4: lbl.Text = "Veuf/Veuve"; break;
        }
        return lbl;
    }

    public void Insert()
    {
        InsertRecord(Prenom, Nom, Telephone, CodePostal, Naissance, Sexe, EtatCivil);
    }

    public void Update()
    {
        UpdateRecord(ID, Prenom, Nom, Telephone, CodePostal, Naissance, Sexe, EtatCivil);
    }
}

```