

PROJET 3

A REMETTRE, AU PLUS TARD,
LE 13 MAI 2016 À 13:00 VIA LÉA

1. Énoncé du problème

Le but ultime de ce projet est de programmer une application qui fait la gestion des succursales d'une entreprise quelconque. Dans la programmation de cette application, *Ajax* est utilisé.

Ce projet se divise en trois (3) parties :

- L'authentification de l'utilisateur
- L'affichage des succursales
- La gestion des opérations sur les succursales

2. Avant de commencer le projet

Créez un dossier nommé **Projet3** puis copiez les fichiers suivants (situés sur le lecteur P).

- **projet3.htm** (ce fichier contient le *XHTML* de votre projet)
- **projet3.js** (ce fichier contient le *Javascript* directement relié à votre projet)
- **libAjax.js**
- **libUtilitaires.js**
- **libValidationBaseER.js**
- **logoCGG.jpg**
- **styleProjet3.css**

Sur la page Web **projet3.htm** et dans le fichier **projet3.js**, remplacez toutes les occurrences de **VotreNom** par votre vrai nom.

3. Structure de la page Web

Dans le corps de la page Web, il y a trois (3) panneaux.

Le panneau *divIdentification* s'affiche lors du chargement de la page Web et les deux (2) autres panneaux sont masqués.

Si l'authentification de l'utilisateur réussit, le panneau *divIdentification* se masque et les panneaux *divOperationsSuccursales* et *divSuccursales* s'affichent.

Identifiez-vous !

Identifiant:

Mot de passe:

Opérations sur les succursales

AJOUT/MODIFICATION D'UNE SUCCURSALE

Ville:

Budget: \$

RETRAIT D'UNE SUCCURSALE

Ville:

AFFICHAGE DU BUDGET D'UNE SUCCURSALE

Ville:

Budget: \$

Succursale(s) enregistrée(s) par ALAIN LOYER

Nombre de succursale(s): 3

Dans le fichier **projet3.js**, la fonction *initialiseInterface* a déjà été programmée (par moi). Elle sert à indiquer lesquels des trois panneaux doivent être affichés. Par exemple, l'appel suivant : *initialiseInterface(true, false, false)* signifie que le panneau *divIdentification* doit être affiché et que les panneaux *divOperationsSuccursales* et *divSuccursales* doivent être masqués.

4. Projet 3 – L’authentification de l’utilisateur

Note importante : Lisez complètement cette section jusqu’à la fin de la page 6 avant de commencer la programmation.

Dans cette partie du projet, l’utilisateur doit être authentifié. Pour ce faire, il doit taper son identifiant et son mot de passe puis cliquer sur le bouton **CONNEXION**.

Son identifiant, c’est son matricule. Pour être valide, son matricule (*tbMatricule*) doit être composé d’exactly 7 chiffres (fabriquez une expression régulière pour vérifier cela).

Pour être valide, son mot de passe (*tbMotDePasse*) doit être composé de 1 à 6 lettres (majuscules ou minuscules) suivi d’exactly 5 chiffres (fabriquez une expression régulière pour vérifier cela).

Si l’une ou l’autre des données (ou les deux) n’est pas valide, un message doit s’afficher (dans *lblMessageConnexion*).

Identifiez-vous !

Identifiant:

Mot de passe:

Identifiant et mot de passe invalides !

Identifiez-vous !

Identifiant:

Mot de passe:

Identifiant invalide !

Identifiez-vous !

Identifiant:

Mot de passe:

Mot de passe invalide !

Par contre, si les deux données sont valides, vous devez effacer ce qu’il y a dans le message (au cas où il y aurait un vieux message).

Ensuite (si les deux données sont valides), l'application serveur **http://424w.cgodin.qc.ca/aloyer/Kit_Ajax_projet/gestion-bd-projet3.php** doit être appelée.

Les données transmises doivent être de la forme suivante :

Action=Connexion&Mat=<Matricule tapé>&MDP=<Mot de passe tapé>

Par exemple :

Action=Connexion&Mat=1234567&MDP=Secret12345

L'application serveur a deux réponses possibles¹ : **PASOK**; ou **OK;<Prénom de l'utilisateur>;<Nom de l'utilisateur>;**

La première réponse signifie que l'utilisateur est inconnu.

Dans ce cas, vous devez afficher un message indiquant que l'utilisateur est inconnu (dans *lblMessageConnexion*).

Identifiez-vous !

Identifiant:	<input type="text" value="1234567"/>	<input type="button" value="NE PLUS SE SOUVENIR DE MOI"/>
Mot de passe:	<input type="password" value="s45678"/>	<input type="button" value="CONNEXION"/>
Utilisateur inconnu !		

La seconde réponse signifie que l'utilisateur est connu. Dans ce cas, vous devez masquer le panneau *divIdentification* et afficher les panneaux *divOperationsSuccursales* et *divSuccursales* (utilisez la fonction *initialiseInterface*). Vous devez également afficher le nom complet de l'utilisateur (dans *lblNomComplet*).

**Succursale(s) enregistrée(s) par
ALAIN LOYER**

Si vous utilisez votre propre matricule et le mot de passe que je vous ai donnés en classe, vous devriez voir votre prénom et votre nom de famille.

Attention : Vous devez utiliser le prénom et le nom de l'utilisateur que l'application serveur vous a transmis. Lorsque je vais corriger, l'utilisateur ne sera pas vous.

¹ Il est possible que l'application serveur émette la réponse **ERREUR**; Cette réponse ne devrait jamais se produire. Si l'application serveur répond par **ERREUR**; cela signifie que vous avez fait une erreur au niveau des données transmises. Cela s'applique également à toutes les requêtes que vous allez envoyer plus tard à cette application serveur.

Lorsque l'utilisateur clique sur le bouton **CONNEXION**, c'est le module *ajax_tenteConnexion* qui est appelé.

Observez que, dans le fichier **projet3.js**, ce module n'a pas été programmé.

Dans le module directeur, votre programme doit valider l'identifiant (le matricule) et le mot de passe. Si l'un ou l'autre n'est pas valide, votre programme doit afficher le message approprié.

```

/*
|-----|
| Module ajax_tenteConnexion
|-----|
*/
function ajax_tenteConnexion() {

    /*
    |-----|
    | recupereReponseServeur
    |-----|
    */

    function recupereReponseServeur(strVerdict) {
        // A programmer
    }

    /*
    |-----|
    | Module directeur (ajax_tenteConnexion)
    |-----|
    */
    // A programmer
}

```

Si les deux données sont valides, votre programme doit faire une requête au serveur Web puis attendre sa réponse (dans la fonction *recupereReponseServeur*) et, dépendant de sa réponse, faire le traitement approprié.

Ne pas oublier de faire la trace (dans *lblRequete* pour la requête et *lblReponse* pour la réponse). Cela va vous permettre de déboguer.

```

Requête: http://424w.cgodin.qc.ca/aloyer/Kit_Ajax_projet/gestion-bd-projet3.php?
Action=Connexion&Mat=1234567&MDP=S12345
Réponse: PASOK;

```

Vous n'avez pas à déclarer le nom complet de l'application sur le serveur Web. Il a déjà été déclaré (par moi). Vous n'avez qu'à l'utiliser :

```
var strNomApplication = 'http://424w.cgodin.qc.ca/aloyer/Kit_Ajax_projet/gestion-bd-projet3.php';
```

Dans cette partie du projet, il est également possible, pour l'utilisateur, de conserver son identifiant et son mot de passe (dans des *cookies*). Pour ce faire, il n'a qu'à cliquer sur le bouton **SE SOUVENIR DE MOI**. S'il le fait, la prochaine fois qu'il va ouvrir la page Web, son identifiant et son mot de passe doivent être automatiquement affichés.

Identifiez-vous !

Identifiant:	<input type="text" value="1234567"/>	<input type="button" value="NE PLUS SE SOUVENIR DE MOI"/>
Mot de passe:	<input type="text" value="s54321"/>	<input type="button" value="CONNEXION"/>
<div style="border: 1px solid red; height: 20px; width: 100%;"></div>		

Il est possible pour l'utilisateur de supprimer l'identifiant et le mot de passe qui ont été conservés. Pour ce faire, il n'a qu'à cliquer sur le bouton **NE PLUS SE SOUVENIR DE MOI**. S'il le fait, la prochaine fois qu'il va ouvrir la page Web, son identifiant et son mot de passe ne seront pas automatiquement affichés.

Dans le fichier **projet3.js**, la fonction *initialiseIdentifiant* est appelée au chargement de la page Web. Dans cette fonction, vous devez vérifier si l'identifiant et le mot de passe ont été conservés (dans des *cookies*). Si non, vous devez masquer le bouton **NE PLUS SE SOUVENIR DE MOI** (*btnNonSouvenir*) et afficher le bouton **SE SOUVENIR DE MOI** (*btnSouvenir*). Si oui, vous devez afficher le bouton **NE PLUS SE SOUVENIR DE MOI** (*btnNonSouvenir*), masquer le bouton **SE SOUVENIR DE MOI** (*btnSouvenir*), récupérer l'identifiant et le mot de passe puis l'afficher dans les zones de texte correspondantes. (A cause des *cookies*, vous devez ouvrir la page Web avec *Internet Explorer*).

La fonction *enregistreIdentifiant* est appelée lorsque l'utilisateur clique sur le bouton **SE SOUVENIR DE MOI** (*btnSouvenir*). Dans ce cas, vous devez enregistrer l'identifiant et le mot de passe dans des *cookies* pendant 365 jours puis masquer le bouton **NE PLUS SE SOUVENIR DE MOI** et afficher le bouton **SE SOUVENIR DE MOI**.

La fonction *detruitIdentifiant* est appelée lorsque l'utilisateur clique sur le bouton **NE PLUS SE SOUVENIR DE MOI** (*btnNonSouvenir*). Dans ce cas, vous devez supprimer l'identifiant et le mot de passe qui ont été conservés dans des *cookies* puis masquer le bouton **SE SOUVENIR DE MOI** et afficher le bouton **NE PLUS SE SOUVENIR DE MOI**.

Note importante : Dès que l'authentification fonctionne, directement sur la page Web, dans l'attribut *type* de *tbMotDePasse* tapez **password** au lieu de **text**. Cela va faire en sorte que le mot de passe ne sera pas visible.

Mot de passe:

5. Projet 3 – L’affichage des succursales

C’est dans le panneau *divSuccursales* que les succursales enregistrées s’affichent. Deux modules ont été prévus à cet effet : *ajax_compteSuccursales* et *ajax_afficheListeSuccursales*.

Le module *ajax_compteSuccursales* s’occupe de compter et d’afficher le nombre de succursales enregistrées.

Vous devez appeler ce module dans la fonction *recupereReponseServeur* du module *ajax_tenteConnexion*,

Les données transmises doivent être de la forme suivante :

Action=Succursale-Compte&Aut=<Matricule tapé><Les 5 chiffres du mot de passe tapé>

Par exemple : (Ici, le matricule est **1234567** et le mot de passe est **Secret12345**)

Action=Succursale-Compte&Aut=123456712345

La réponse² est : *<Le nombre de succursales enregistrées>*;

Dans ce cas, vous devez afficher un message indiquant le nombre de succursales enregistrées (dans *lblSuccursales*; cette étiquette est située dans le panneau *divSuccursales*). **Normalement, pour l’instant, vous ne devriez pas avoir de succursale.**

**Succursale(s) enregistrée(s) par
ALAIN LOYER**

Nombre de succursale(s): 3

S’il n’y a aucune succursale enregistrée, votre programme doit masquer le bouton **RÉINITIALISER** ainsi que les tableaux XHTML *tabRetrait* et *tabVisualisation* (ces tableaux contiennent respectivement les opérations « RETRAIT D’UNE SUCCURSALE » et « AFFICHAGE DU BUDGET D’UNE SUCCURSALE »).

Dans le cas contraire, vous devez les afficher.

Ne pas oublier de faire la trace (dans *lblRequete* pour la requête et *lblReponse* pour la réponse). Cela va vous permettre de déboguer.

RÉINITIALISER

² **Attention** : Il est possible que la réponse soit **ERREUR**;

Lorsque l'utilisateur clique sur le bouton **LISTER**, c'est le module *ajax_afficheListeSuccursales* qui est appelé.

LISTER

Les données transmises doivent être de la forme suivante :

Action=Succursale-Liste&Aut=<Matricule tapé><Les 5 chiffres du mot de passe tapé>

Par exemple : (Ici, le matricule est **1234567** et le mot de passe est **Secret12345**)

Action=Succursale-Liste&Aut=123456712345

Les réponses possibles³ sont : *AUCUNE*; ou *<Ville1>,<Budget1>;<Ville2>,<Budget2>;<Ville3>,<Budget3>;* (et ainsi de suite)

La première réponse signifie qu'il n'y a aucune succursale. Dans ce cas, vous devez afficher ce message (dans *lblSuccursales*).

**Succursale(s) enregistrée(s) par
ALAIN LOYER**

Aucune succursale enregistrée...

La seconde réponse signifie qu'il y a des succursales. Dans ce cas, vous devez construire un tableau XHTML dans un flux de données (dans *lblSuccursales*).

**Succursale(s) enregistrée(s) par
ALAIN LOYER**

Dans la construction de votre tableau XHTML, voici les sélecteurs de mises en forme que vous devez utiliser :

Pour le tableau : **sTableauSuccursales**

Pour la 1^{ère} ligne du tableau : **sEnteteTableauSuccursales**

Pour les autres lignes du tableau: **sCorpsTableauSuccursales**

Pour la cellule qui contient le numéro : **sCelNoSuccursale**

Pour la cellule qui contient la ville : **sCelVille**

Pour la cellule qui contient le budget : **sCelBudget** (Attention : il doit y avoir un espace suivi d'un \$ à la fin du montant).

No	Ville	Budget
1	Ile-Bizard	5392 \$
2	Montreal	5000000 \$
3	Quebec	1500000 \$

Ne pas oublier de faire la trace (dans *lblRequete* pour la requête et *lblReponse* pour la réponse). Cela va vous permettre de déboguer.

³ **Attention** : Il est possible que la réponse soit **ERREUR**;

6. Projet 3 – La gestion des opérations sur les succursales

Il existe trois (3) opérations possibles :

- 1- AJOUT/MODIFICATION D'UNE SUCCURSALE
- 2- RETRAIT D'UNE SUCCURSALE
- 3- AFFICHAGE DU BUDGET D'UNE SUCCURSALE

Les deux dernières opérations sont visibles lorsqu'il existe au moins une succursale.

Il y a également une opération spéciale qui consiste à tout réinitialiser (visible lorsqu'il existe au moins une succursale).

Toutes ces opérations sont situées dans le panneau *divOperationsSuccursales*.

6.1 L'opération AJOUT/MODIFICATION D'UNE SUCCURSALE

Dans cette partie du projet, l'utilisateur a la possibilité d'ajouter ou de modifier une succursale.

Lorsque l'utilisateur clique sur le bouton **SOUMETTRE**, c'est le module *ajax_tenteAjoutModificationSuccursale* qui est appelé.

La ville doit être validée. Pour être valide, la ville (*tbVilleAjout*) doit obligatoirement débiter par au moins 1 lettre et être suivi (possiblement) d'un tiret et d'au moins 1 lettre; ce tiret et ces lettres peuvent se répéter 0, 1 ou plusieurs fois (fabriquez une expression régulière pour vérifier cela). Par exemple, **Montreal**, **St-Joachim**, **St-Joachim-Des-Creux** sont des villes valides. Les chiffres, les caractères accentués, les espaces et les caractères spéciaux ne doivent pas être acceptés (car les caractères accentués, les espaces et certains caractères spéciaux demandent des traitements spéciaux quand ils passent à travers le Web).

Le budget doit être validé. Pour être valide, le budget (*tbBudgetAjout*) doit être composé de 3 à 7 chiffres (fabriquez une expression régulière pour vérifier cela). De plus, ce budget doit être plus grand ou égal à 500.

Si l'une ou l'autre des données (ou les deux) n'est pas valide un message doit s'afficher (dans *lblMessageAjout*).

Ville: <input type="text" value="L'Islet"/> Budget: <input type="text" value="670"/> \$ <input type="button" value="SOUMETTRE"/> <div>Ville et budget invalides !</div> <input type="button" value="EFFACER"/>	Ville: <input type="text"/> Budget: <input type="text"/> \$ <input type="button" value="SOUMETTRE"/> <div>Ville et budget invalides !</div> <input type="button" value="EFFACER"/>	Ville: <input type="text" value="St-Joachim-"/> Budget: <input type="text" value="500"/> \$ <input type="button" value="SOUMETTRE"/> <div>Ville invalide !</div> <input type="button" value="EFFACER"/>
Ville: <input type="text" value="St-Joachim-Des-Creux"/> Budget: <input type="text" value="500abc"/> \$ <input type="button" value="SOUMETTRE"/> <div>Budget invalide !</div> <input type="button" value="EFFACER"/>	Ville: <input type="text" value="Montréal"/> Budget: <input type="text" value="500"/> \$ <input type="button" value="SOUMETTRE"/> <div>Ville invalide !</div> <input type="button" value="EFFACER"/>	Ville: <input type="text" value="Montreal"/> Budget: <input type="text" value="abc"/> \$ <input type="button" value="SOUMETTRE"/> <div>Budget invalide !</div> <input type="button" value="EFFACER"/>

Par contre, si les deux données sont valides, vous devez appeler l'application serveur **http://424w.cgodin.qc.ca/aloyer/Kit_Ajax_projet/gestion-bd-projet3.php**.

Les données transmises doivent être de la forme suivante :

Action=Succursale-Ajout&Aut=<Matricule tapé><Les 5 chiffres du mot de passe tapé>&Ville=<Ville tapée>&<Budget tapé>

Par exemple : (Ici, le matricule est **1234567** et le mot de passe est **Secret12345**)

Action=Succursale-Ajout&Aut=123456712345&Ville=Montreal&Budget=10000

Les réponses possibles⁴ sont : **PASOK**; ou **OKM**; ou **OKI**;

La première réponse signifie que cette succursale existe (elle a déjà été ajoutée dans le passé). Dans ce cas, vous devez afficher un message (dans *lblMessageAjout*) pour indiquer que la succursale existe déjà.

La seconde réponse signifie que le budget de la succursale vient tout juste d'être modifié. Dans ce cas, vous devez afficher un message (dans *lblMessageAjout*) pour indiquer la modification.

La troisième réponse signifie que la succursale vient tout juste d'être ajoutée. Dans ce cas, vous devez afficher un message (dans *lblMessageAjout*) pour indiquer l'ajout. De plus, vous devez réafficher le nombre de succursales (appelez le module *ajax_compteSuccursales*).

Ne pas oublier de faire la trace (dans *lblRequete* pour la requête et *lblReponse* pour la réponse). Cela va vous permettre de déboguer.

Lorsque l'utilisateur clique sur le bouton **EFFACER**, c'est la fonction *effacerAjoutModification* qui est appelée. Vous devez programmer cette fonction en effaçant ce qu'il y a dans *tbVilleAjout*, *tbBudgetAjout* et *lblMessageAjout*.

Ville:

Budget: \$

Succursale existante !

Ville:

Budget: \$

Succursale modifiée

Ville:

Budget: \$

Succursale ajoutée

**Succursale(s) enregistrée(s) par
ALAIN LOYER**

Nombre de succursale(s): 3

⁴ **Attention** : Il est possible que la réponse soit **ERREUR**;

6.2 L'opération RETRAIT D'UNE SUCCURSALE

Dans cette partie du projet, l'utilisateur a la possibilité de retirer une succursale.

Lorsque l'utilisateur clique sur le bouton **SOUMETTRE**, c'est le module *ajax_tenteRetraitSuccursale* qui est appelé.

La ville (*tbVilleRetrait*) doit être validée. Les règles de validation de la ville sont les mêmes que pour l'ajout. Si la ville n'est pas valide, un message doit s'afficher (dans *lblMessageRetrait*).

Ville:	<input type="text" value="3-Rivieres"/>	<input type="button" value="SOUMETTRE"/>
	<div>Ville invalide !</div>	<input type="button" value="EFFACER"/>

Par contre, si la ville est valide, l'application serveur http://424w.cgodin.qc.ca/aloyer/Kit_Ajax_projet/gestion-bd-projet3.php doit être appelée.

Les données transmises doivent être de la forme suivante :

Action=Succursale-Retrait&Aut=<Matricule tapé><Les 5 chiffres du mot de passe tapé>&Ville=<Ville tapée>

Par exemple : (Ici, le matricule est **1234567** et le mot de passe est **Secret12345**)

Action=Succursale-Retrait&Aut=123456712345&Ville=Montreal

Les réponses possibles⁵ sont : *PASOK*; ou *OK*;

La première réponse signifie que la succursale est inconnue. Dans ce cas, vous devez afficher le message (dans *lblMessageRetrait*) indiquant que la succursale n'est pas connue.

Ville:	<input type="text" value="St-Tite-Des-Caps"/>	<input type="button" value="SOUMETTRE"/>
	<div>Succursale inconnue !</div>	<input type="button" value="EFFACER"/>

La seconde réponse signifie que la succursale vient tout juste d'être retirée. Dans ce cas, vous devez afficher un message (dans *lblMessageRetrait*) pour indiquer le retrait et réafficher le nombre de succursales (appelez le module *ajax_compteSuccursales*).

Ville:	<input type="text" value="Montreal"/>	<input type="button" value="SOUMETTRE"/>
	<div>Succursale retirée</div>	<input type="button" value="EFFACER"/>

Lorsque l'utilisateur clique sur le bouton **EFFACER**, c'est la fonction *effacerRetrait* qui est appelée. Vous devez programmer cette fonction en effaçant ce qu'il y a dans *tbVilleRetrait* et *lblMessageRetrait*.

⁵ **Attention** : Il est possible que la réponse soit **ERREUR**;

6.3 L'opération AFFICHAGE DU BUDGET D'UNE SUCCURSALE

Dans cette partie du projet, l'utilisateur a la possibilité de visualiser le budget d'une succursale.

Lorsque l'utilisateur clique sur le bouton **SOUMETTRE**, c'est le module *ajax_tenteVisualisationBudgetSuccursale* qui est appelé.

La ville (*tbVilleBudgetVisualisation*) doit être validée. Si la ville n'est pas valide, un message doit s'afficher (dans *lblMessageBudgetVisualisation*).

Ville:

Budget: \$

Ville invalide !

Par contre, si la ville est valide, l'application serveur http://424w.cgodin.qc.ca/aloyer/Kit_Ajax_projet/gestion-bd-projet3.php doit être appelée.

Les données transmises doivent être de la forme suivante :

Action=Succursale-Budget&Aut=<Matricule tapé><Les 5 chiffres du mot de passe tapé>&Ville=<Ville tapée>

Par exemple : (Ici, le matricule est **1234567** et le mot de passe est **Secret12345**)

Action=Succursale-Budget&Aut=123456712345&Ville=Montreal

Les réponses possibles⁶ sont : *PASOK*; ou *<le budget de la ville>*;

La première réponse signifie que la succursale est inconnue. Dans ce cas, vous devez afficher le message (dans *lblMessageBudgetVisualisation*) indiquant que la succursale n'est pas connue puis effacer ce qu'il y a dans le budget (dans *lblBudgetVisualisation*)

Ville:

Budget: \$

Succursale inconnue !

La seconde réponse signifie qu'il y a un budget pour cette ville. Dans ces cas, vous devez afficher un message (dans *lblMessageBudgetVisualisation*) puis afficher le budget (dans *lblBudgetVisualisation*).

Ville:

Budget: **1500000 \$**

Budget affiché

⁶ **Attention** : Il est possible que la réponse soit **ERREUR**;

Ne pas oublier de faire la trace (dans *lblRequete* pour la requête et *lblReponse* pour la réponse).

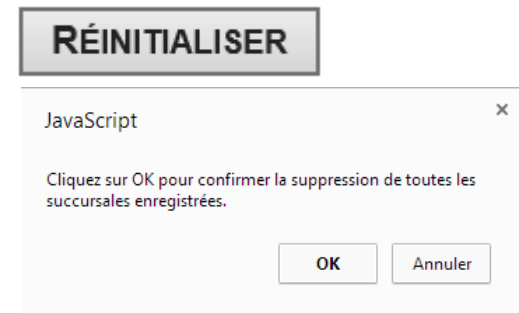
Lorsque l'utilisateur clique sur le bouton **EFFACER**, c'est la fonction *effacerBudgetVisualisation* qui est appelée. Vous devez programmer cette fonction en effaçant ce qu'il y a dans *tbVilleBudgetVisualisation*, *lblBudgetVisualisation* et *lblMessageBudgetVisualisation*.

6.4 L'opération RÉINITIALISER

Dans cette partie du projet, l'utilisateur a la possibilité de tout réinitialiser.

Lorsque l'utilisateur clique sur le bouton **RÉINITIALISER**, c'est le module *ajax_reinitialiseSuccursales* qui est appelé.

Tout d'abord, vous devez afficher une alerte qui demande à l'utilisateur de confirmer (utilisez la fonction *confirm* qui a déjà été programmée en *Javascript*).



Si l'utilisateur confirme, l'application serveur http://424w.cgodin.qc.ca/aloyer/Kit_Ajax_projet/gestion-bd-projet3.php doit être appelée.

Les données transmises doivent être de la forme suivante :

Action=Succursale-Suppression&Aut=<Matricule tapé><Les 5 chiffres du mot de passe tapé>

Par exemple : (Ici, le matricule est **1234567** et le mot de passe est **Secret12345**)

Action=Succursale-Suppression&Aut=123456712345

La réponse devrait être **OK**;

Dans ce cas, vous devez effacer toutes les données affichées sur la page Web et réafficher le nombre de succursales (appelez le module *ajax_compteSuccursales*). Dans cette liste, vous ne devriez plus voir aucune succursale.

Ne pas oublier de faire la trace (dans *lblRequete* pour la requête et *lblReponse* pour la réponse).

7. La déconnexion

Dans cette partie du projet, l'utilisateur a la possibilité de se déconnecter.



DÉCONNEXION

Lorsque l'utilisateur clique sur le bouton **DÉCONNEXION**, c'est la fonction *deconnexion* qui est appelée.

Pour déconnecter l'utilisateur, vous n'avez qu'à recharger la page Web.

8. Barème de correction

Note 1 : Dans **projet3.js**, vous pouvez ajouter toutes les fonctions que vous voulez.

Note 2 : Dans **styleProjet3.js**, vous pouvez modifier les couleurs si vous voulez (mais allez-y mollo!).

Note 3 : Lorsque je vais corriger, je vais utiliser un autre utilisateur que vous.

Ce projet sera corrigé sur 100 et comptera pour 10% de la session.

- ☐ **Bon fonctionnement du programme et validité des résultats** 100 points
- ☐ **Modification du XHTML déjà tapé (sauf aux endroits indiqués) et/ou modification des librairies** -20 points
- ☐ **Absence de votre nom aux endroits appropriés** jusqu'à -05 points
- ☐ **Projet non remis sur LÉA dans un dossier compressé** -05 points
- ☐ **Présentation du code source** (identificateurs significatifs, convention des noms respectée, etc.) ... jusqu'à -20 points
- ☐ **Non utilisation d'expressions régulières dans les validations** jusqu'à -10 points
- ☐ **Retard** -10% par jour (maximum 5 jours)
- ☐ **Utilisation d'une librairie extérieure (que vous n'avez pas programmée)** -100 points
- ☐ **Plagiat partiel ou total** -100 points pour tous les étudiants impliqués