

# EFREI – M1 – Java EE (déc. 2014)

## Projet

---

Objectif :

**Réaliser une petite application permettant de gérer un annuaire de membres.**

### Exigences fonctionnelles

#### Connexion (Ecran #1)

Sur la page d'accueil affichez une page de connexion comme suit :

**EFREI / M1A / JEE / Mini projet**

Identifiez-vous

Login	<input type="text"/>
Mot de passe	<input type="password"/>

Il faut que l'utilisateur renseigne les informations suivantes pour accéder à l'annuaire :

Login : **admin**

Mot de passe : **efreijee**

(Dans la partie "Exigences techniques" vous trouverez des indications sur la gestion des paramètres de connexion).

- a) Si la connexion échoue, affichez le même écran mais avec un message d'erreur en rouge :

**EFREI / M1A / JEE / Mini projet**

Echec de la connexion! Verifiez votre login et/ou mot de passe et essayez à nouveau.

Identifiez-vous

Login	<input type="text"/>
Mot de passe	<input type="password"/>

## Menu principal (Ecran #2)

En cas de succès, l'écran suivant est proposé :

- ✓ Si des membres existent en base, alors afficher la liste comme suit :

### Liste des membres du club EFREI/M1A/JEE

Client:

ID Client	NOM	PRENOM	TEL DOMICILE	TEL PORTABLE	TEL PRO	ADRESSE	CODE POSTAL	VILLE	EMAIL
<input type="checkbox"/>	Maradona	Diego	+54 234 458 748	+54 222 458 711	+54 111 000 707	10 rue de la Main de Dieu	57481	Buenos Aires	diego.maradona@efrejee.fr
<input type="checkbox"/>	Phelps	Michael	+1 148 721 988	+1 222 333 555	+1 444 777 222	11 rue des 8 Medailles d'Or	41781	Baltimore	michael.phelps@efrejee.fr
<input type="checkbox"/>	Riner	Teddy	+33 1 44 55 88 77	+33 6 88 99 22 44	+33 1 99 66 77 44	12 rue du Judoka Qui Gagne Tout	87451	Levallois	tedy.riner@efrejee.fr
<input type="checkbox"/>	Bryant	Kobe	+33 1 22 33 44 55	+33 6 77 66 44 11	+33 1 88 44 77 11	13 rue des Lakers	11448	Los Angeles	kobe.bryant@efrejee.fr

- ✓ La sélection d'une ligne du tableau et un clic sur **Supprimer** doit provoquer la suppression en base du membre sélectionné.
  - Si l'opération a réussi, affichez à nouveau l'écran avec la liste actualisée et le message de réussite suivant, en vert : **La suppression a réussi !**
  - Sinon, affichez l'écran avec la même liste et avec le message d'échec suivant, en rouge : **La suppression a échoué !**
- ✓ Si aucun membre n'est enregistré en base, affichez tout simplement le message suivant (en bleu) :

**Le club a besoin de membre(s) !**

- ✓ Un champ de recherche libre permet, lorsqu'on tape le nom d'un membre puis qu'on clique sur "Chercher", d'afficher la ligne correspondant à ce membre. En page de page, il faudra toujours afficher les boutons "Détails" et "Supprimer".  
Si aucun membre ne correspondant au critère de recherche, afficher une page vide avec le message (en rouge) : **Aucun résultat trouvé.**

## Ecran "Détails d'un membre" (Ecran #3)

La sélection d'une ligne et un clic sur "**Détails**" provoque l'affichage d'un écran affichant les informations du membre sélectionné. Remarquez que le titre doit être adapté à l'identité du membre.

### Détails d'un membre du club EFREI/M1A/JEE

Nom :  Prenom :

#### Numero de telephone

Domicile :

Portable :

Professionnel :

Adresse:

Code postal:  Ville:

Adresse e-mail :

Sur cet écran, on peut également revenir à la page précédente grâce au bouton "Retour".

## Directives techniques

### Vue

- Les pages seront réalisées en JSP.
- Libre à vous d'utiliser des JSP "basiques" ou EL/JSTL. Mais votre fil directeur doit toujours être : "le moins de code Java possible dans les pages JSP".

### MVC or not MVC ?

Vous devrez réaliser (et livrer) **trois** versions de l'application :

#### a) Version 1.0

- Dans cette version, vous utiliserez zéro servlet.
- Tout doit se faire dans les pages JSP et/ou des classes Java.
- Les paramètres de connexion seront gérés en base.

#### b) Version 2.0

- Dans cette version, vous mettrez en œuvre un aiguillage des requêtes via une servlet que vous nommerez **Controleur.java**
- Les accès à la base et le traitement des requêtes se feront dans la servlet.
- Les paramètres de connexion seront gérés dans le contexte.
- En résumé, tout sera géré par des JSP, une servlet, un ou des Java Beans, un ou des classes Java simples (POJO).

#### c) Version 3.0 (les EJB entrent en scène)

- La V 3.0 sera une évolution de la V 2.0 au niveau de l'accès en base.
- Vous garderez la servlet **Controleur.java**, cependant toutes les opérations de persistance seront traitées via des **EJB et l'API JPA** (réf. cours d'Archi + Support de cours JEE + ...).
- La V3.0 est iso-fonctionnalités avec la V2.0
- Les paramètres de connexion seront toujours gérés dans le contexte.

## Précisions complémentaires

- ✓ Le SGBD à utiliser est obligatoirement **Java DB** (Derby). Injectez au moins 4 enregistrements dans votre base.
- ✓ Le code doit être commenté au maximum.
- ✓ 3 versions = 3 projets dans NetBeans = 1 fichier .zip.
- ✓ Dès que votre groupe a terminé le projet, tout votre espace de travail doit être zippé et déposé sur Campus.
- ✓ Ce mini projet est à rendre au plus tard le **vendredi 19/12 à la fin du cours**.
- ✓ Vous devrez travailler en groupe de 3 étudiants max.
- ✓ Le DE final comportera 50% de questions sur que vous aurez réalisé dans le cadre de ces travaux pratiques. Vous devrez être à même d'expliquer chaque API utilisée et de justifier tous vos choix techniques.