

[TP-1] [4^{ème} Année G.I][Part-I] : [Servlet]

Création d'un projet Web Dynamique :

1. Le projet web dynamique aura comme nom **'Tp1ServletGroupeXX'**.
2. La page d'accueil de votre projet aura comme nom **'index.html'**.
3. Le **'web.xml'** doit contenir index.html comme la seul page d'accueil.
4. Le contenu de votre page d'accueil est :

The screenshot shows a web browser window with the address bar displaying 'http://localhost:8080/Tp1Servlet/'. The page has an orange background and contains a registration form. The form includes a dropdown menu for 'Civilité' with 'Monsieur' selected, a text input for 'Nom' containing 'BERRICH', and a text input for 'Prénom' containing 'Jamal'. At the bottom of the form are two buttons: 'Envoyer le formulaire' and 'Tout effacer'.

5. Le but de la deuxième partie est d'implémenter une Servlet qui permet, après avoir rempli le formulaire et cliquer sur le bouton 'Envoyer le formulaire', de retourner le résultat suivant :

The screenshot shows a web browser window with the address bar displaying 'http://localhost:8080/Tp1Servlet/Formulaire?civile=Monsieur&nom=B'. The page has an orange background and displays the message 'Enregistrement de vos coordonnées effectué' in large yellow text. Below this, it says 'Bonjour Monsieur Jamal BERRICH.' in yellow text.

Création de la Servlet : Technique de base.

1. La Servlet aura comme nom **'Formulaire.java'**.
2. On laisse pour le moment la configuration par défaut de la Servlet et elle ne doit ne doit surchargé aucune méthode.
3. La Servlet contiendra la méthode **'getFormulaire'**, son but est générer le résultat de mon appel de la Servlet **'Formulaire'**.

```
protected void getFormulaire(HttpServletRequest request, HttpServletResponse response)
```

4. Pour l'implémentation, la Servlet doit surcharger une seule méthode qui fait appel à ma méthode 'getFormulaire'.
5. Le nouveau url d'accès à cette Servlet sera :



6. On va essayer de refaire le même traitement (4) mais cette fois avec les deux méthodes '**doGet**' et '**doPost**'.

Création de la Servlet : Technique avancée.

1. On va essayer de compter le nombre des utilisateurs qui font appel à cette Servlet, pour cela elle va contenir l'attribue suivant :

```
private int count;
```

2. A l'initialisation de la Servlet, l'attribue 'count' doit avoir la valeur 0 à partir du fichier 'web.xml'.
3. A chaque connexion, on aura le message 'Vous êtes l'utilisateur N : xx'.
4. Dans cette étape on va essayer d'afficher pour la demande de chaque utilisateur le nom et le prénom et de l'utilisateur qui a été connecté juste avant lui depuis son poste. Pour cela je vais utiliser la notion des Cookies.
5. La méthode '**enregistrerUtilisateur**' permet d'enregistrer l'utilisateur dans un cookie et l'envoyer vers le navigateur du client.

```
protected void enregistrerUtilisateur(HttpServletRequest request, HttpServletResponse response)
```

7

6. La méthode '**afficherUtilisateur**' permet de récupérer le nom et le prénom de l'utilisateur enregistré dans les cookies.

```
protected void afficherUtilisateur(PrintWriter out)
```

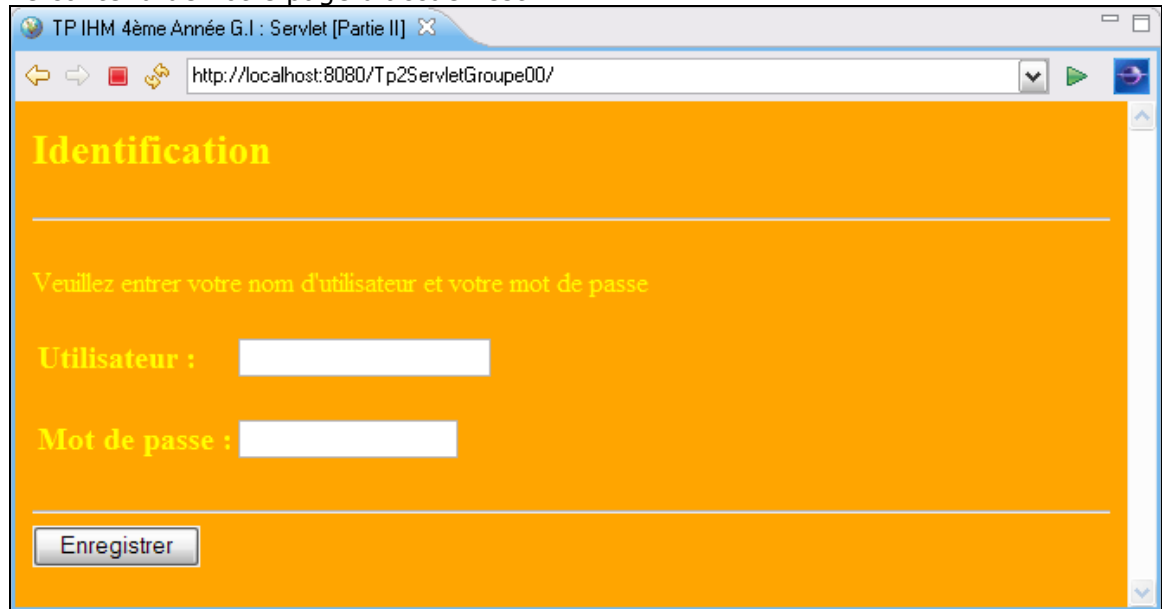
Déploiement :

1. Le déploiement sera effectué grâce à l'application 'Tomcat Manager'.
2. La génération du fichier 'war' sera effectuée par Eclipse.
3. Chaque groupe va déployer et tester le projet réalisé par un autre groupe.

[TP-2] [4^{ème} Année G.I][Part-II] : [Servlet]

Création d'un projet Web Dynamique :

1. Le projet web dynamique aura comme nom **'Tp2ServletGroupeXX'**.
2. La page d'accueil de votre projet aura comme nom **'identification.html'**.
3. Le 'web.xml' doit contenir la page identification.html comme page d'accueil.
4. Le contenu de votre page d'accueil est :



5. Le but de la deuxième partie est d'implémenter une Servlet qui permet de retourner le résultat suivant :



Création de la Servlet : Technique de base.

1. La Servlet aura comme nom **'Identification.java'**.
2. Le nom du package qui contient cette Servlet est : **'ma.supmti.identification'**.
3. On laisse pour le moment la configuration par défaut de la Servlet et elle ne doit ne doit surchargé que la méthode 'doPost' pour générer le résultat de l'appel.
4. On va faire la gestion des utilisateurs, on supposera que tous les utilisateurs sont reconnus par le système.

Création de la Servlet : Technique avancée.

1. On va essayer de compter le nombre des utilisateurs qui font appel à cette Servlet, pour cela elle va contenir l'attribue suivant :

```
private int count;
```

2. A l'initialisation de la Servlet, l'attribue 'count' doit avoir la valeur 0 à partir du fichier 'web.xml'.
3. A chaque connexion, on aura le message 'Vous êtes l'utilisateur N : xx'.
4. Dans cette étape, on va essayer d'afficher un seul numéro pour chaque utilisateur, ce numéro unique sera l'identifiant de l'utilisateur.
5. S'il s'agit d'un nouveau utilisateur l'attribue 'count' va être incrémenté par 1, sinon, on va afficher l'identifiant 'count' attribué à cet utilisateur lors de sa première connexion.
6. La méthode **'enregistrerUtilisateur'** permet d'enregistrer les coordonnées de l'utilisateur dans une session. Pour chaque utilisateur, on va enregistrer son nom de l'utilisateur, son mot de passe et l'identifiant 'count'.

```
protected void enregistrerUtilisateur(HttpSession session, HttpServletRequest request)
```

7. La méthode **'ifUtilisateurExiste'** permet de tester si l'utilisateur est déjà connecté. Le test comportera juste le nom de l'utilisateur.

```
protected boolean ifUtilisateurExiste(HttpSession session, String nomUtilisateur)
```

Déploiement :

1. Le déploiement sera effectué grâce à l'application 'Tomcat Manager'.
2. La génération du fichier 'war' sera effectuée par Eclipse.
3. Chaque groupe va déployer et tester le projet réalisé par un autre groupe.