

## TP 10. Servlet

### Exercice 1: Création d'un projet d'application web avec Netbeans

1. Créez un nouveau projet menu **File > New Project**
2. Dans l'assistant de création de projets choisissez la catégorie **Java Web**
3. Choisissez **Web Application** comme type de projet
4. Cliquez sur le bouton **Next** pour passer à la deuxième étape de l'assistant de création.
5. Donnez un nom a votre projet (par exemple MaPremiereAppliWeb et choisissez un répertoire pour l'enregistrer.
6. Cliquez à nouveau sur le bouton **Next** pour passer à la troisième étape de l'assistant de création.
7. Sélectionnez le serveur (conteneur de servlets) sur lequel vous souhaitez déployer votre application web lors des tests (vous prendrez le serveur Tomcat 7.x ou 8.x intégré à Netbeans) ainsi que la version JEE à laquelle vous voulez que votre projet soit conforme (JEE7 ou une version antérieure).
8. Cliquez sur le bouton **Finish** (on travaillera sans Framework d'application web). Netbeans crée dans votre espace de travail les fichiers de votre projet.

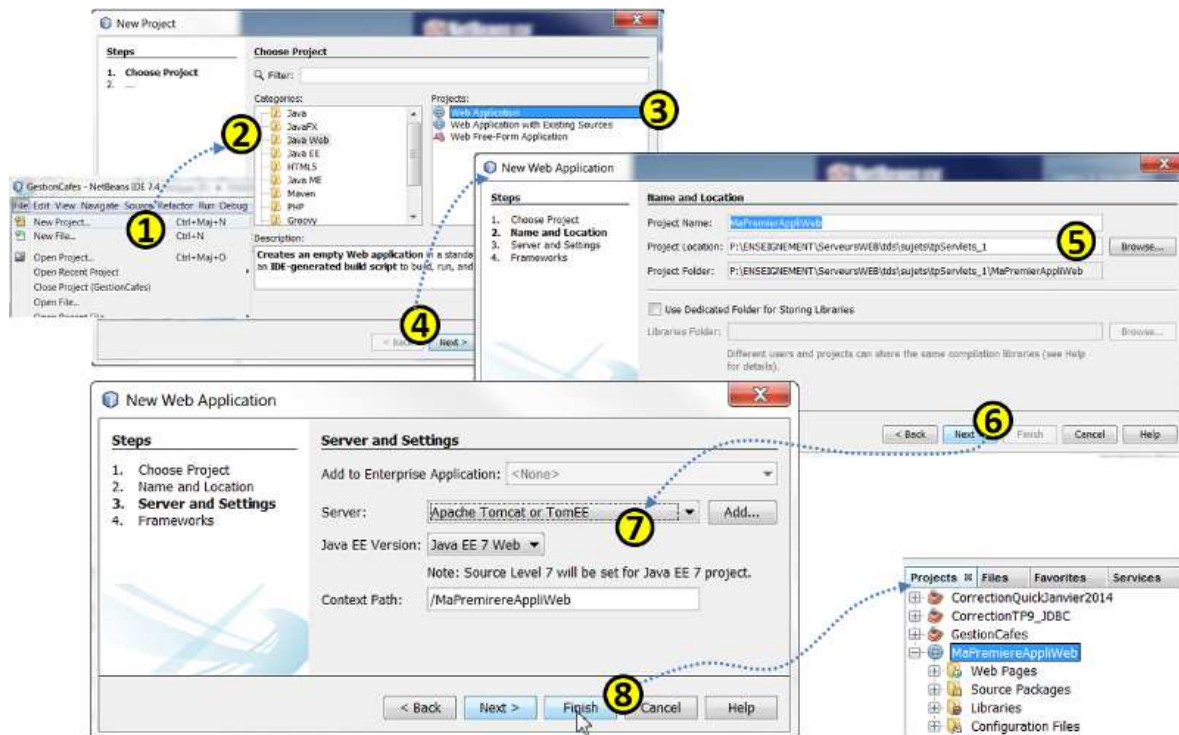


Figure 1: Création d'un nouveau projet d'application web

Un projet d'application web a été créé, la figure 2 vous montre sa structure.

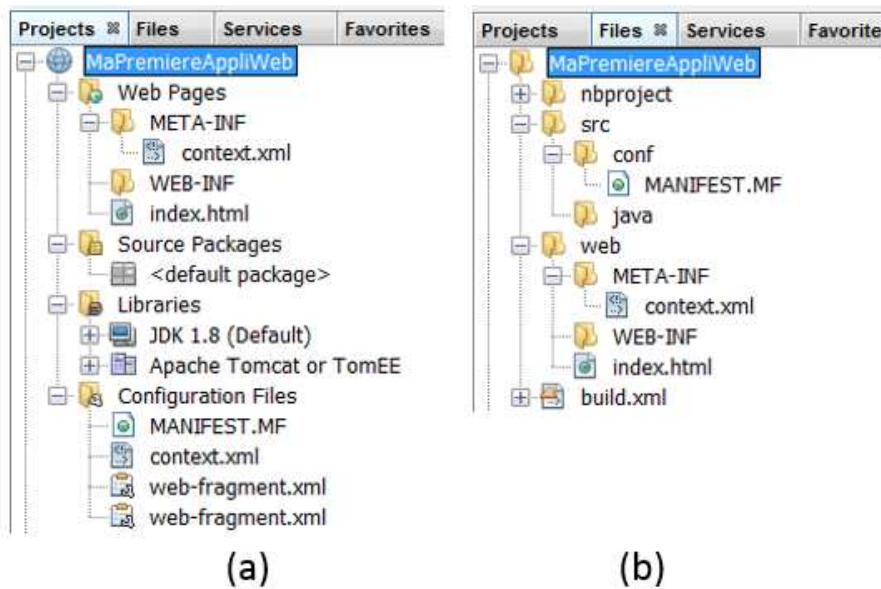


Figure 2: Structure d'un projet d'application web (a) selon la vision **Projets**, (b) selon la vision **système de fichiers**

Parmi les fichiers créés se trouve le fichier index.html, page d'accueil du site.

## Exercice 2: Exécution de l'application web

1. Modifiez le titre de la page et le contenu index.html afin qu'elle affiche le message suivant : "Page d'accueil de ma 1<sup>ère</sup> application web" ou tout autre message de votre choix.
2. Exécutez l'application (faites un clic droit sur le projet et choisissez l'item **Run** du menu contextuel).

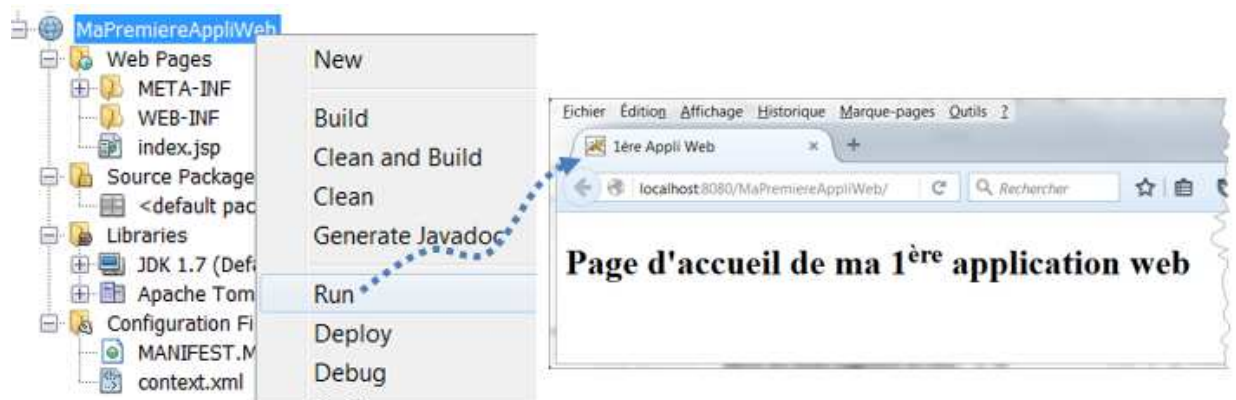


Figure 3: Exécution de l'application.

L'exécution de l'application se déroule selon les étapes suivantes :

1. Le serveur Tomcat auquel est associé le projet est automatiquement lancé (si ce n'est déjà fait)
2. L'application est compilée et est déployée sur celui-ci

3. Le navigateur web par défaut de votre système ouvre un onglet sur une page dont l'url est celle de votre application web. Si votre système possède plusieurs navigateurs web, vous pouvez choisir parmi ceux-ci celui qui sera utilisé pour tester vos applications web à l'aide de l'item **Options** du menu **outils** (voir figure 4).

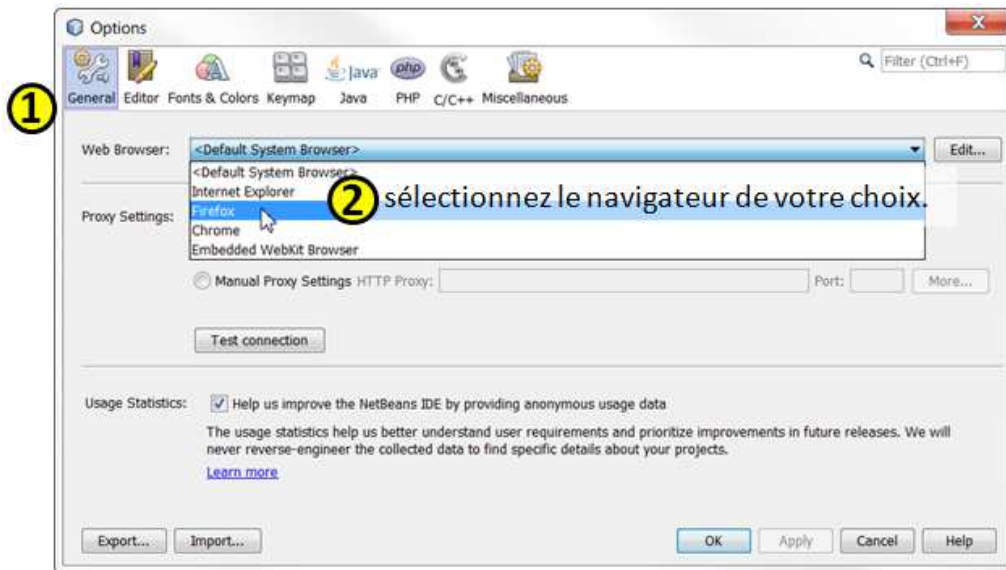


Figure 4: Choix du navigateur par défaut.

## 2. Une première Servlet

Nous allons maintenant créer notre première Servlet qui affichera la date courante du système.

### Exercice 3: Création de la Servlet HelloWorldServlet

1. Créez un nouveau package (par exemple appliweb1.servlets)

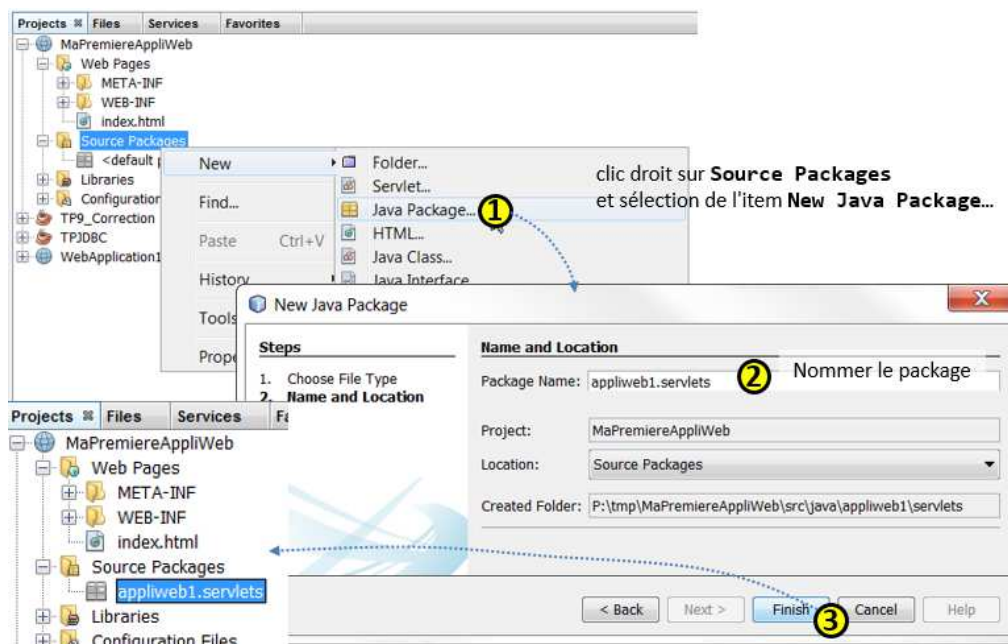


Figure 5: Création d'un package Java

2. Dans ce package, créez une nouvelle classe de Servlet nommée HelloWorldServlet.

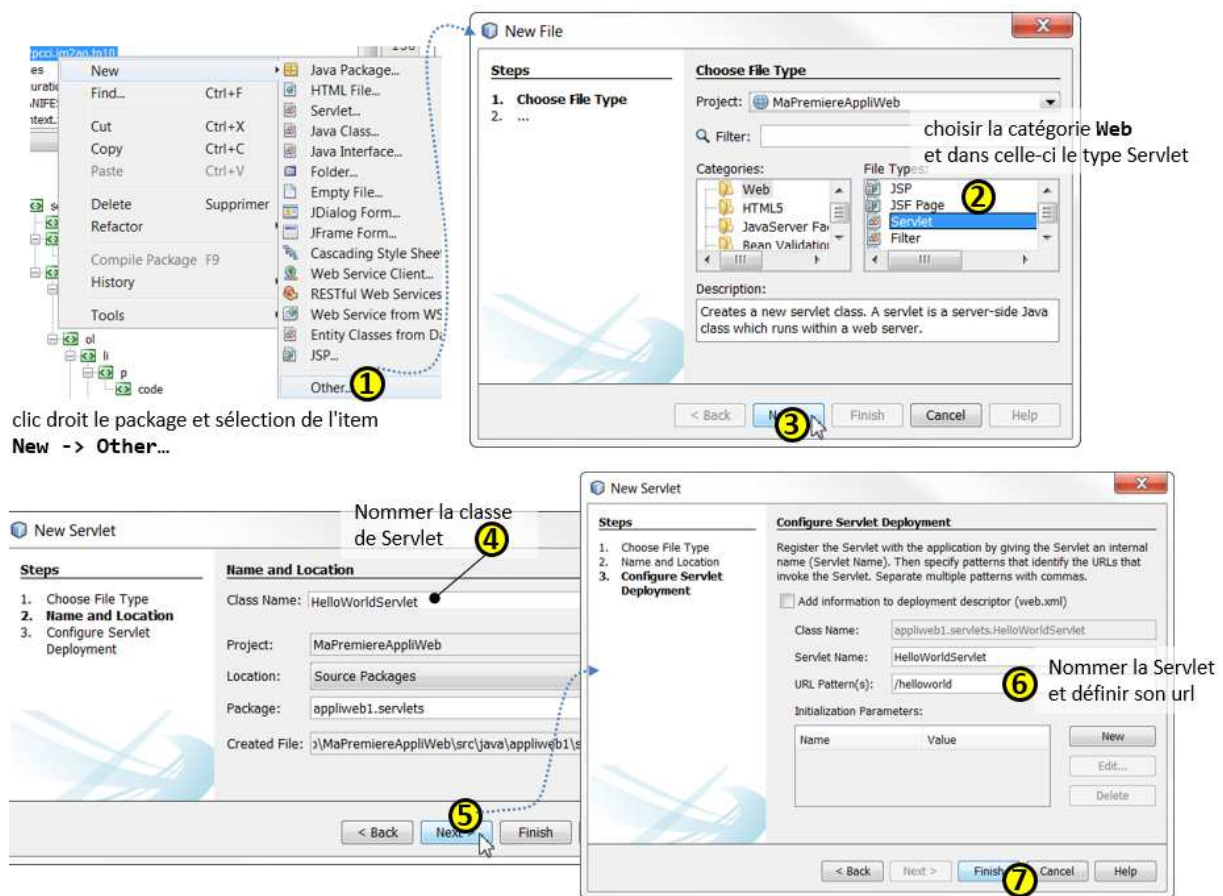


Figure 6: Création d'une Servlet

3. Modifiez le code la servlet de manière à ce que le contenu qu'elle génère affiche en plus la date courante (pour cela vous pouvez créer une instance de la classe `java.util.Date` que vous écrirez sur la sortie de la réponse de votre servlet).
4. Réexécutez votre application web. Par défaut votre navigateur est ouvert sur la page d'url `http://localhost:8080/MaPremiereAppliWeb/` qui correspond à la page `index.html`. Pour avoir la réponse générée par votre servlet `HelloWorldServlet` complétez l'url précédente avec l'url associée à votre servlet (*urlpattern* que vous avez spécifiée lors de la création de la servlet et qui est définie dans l'annotation `@WebServlet` dans le code source de votre servlet).

```
@WebServlet(name = "HelloWorldServlet", urlPatterns = {"/helloworld"})  
public class HelloWorldServlet extends HttpServlet {
```

Figure 7: annotation pour la définition de l'url de la servlet

5. Vérifiez que votre servlet produit bien les résultats escomptés, en particulier que chaque fois que vous réactualisez la page, la date du système est bien mise à jour.

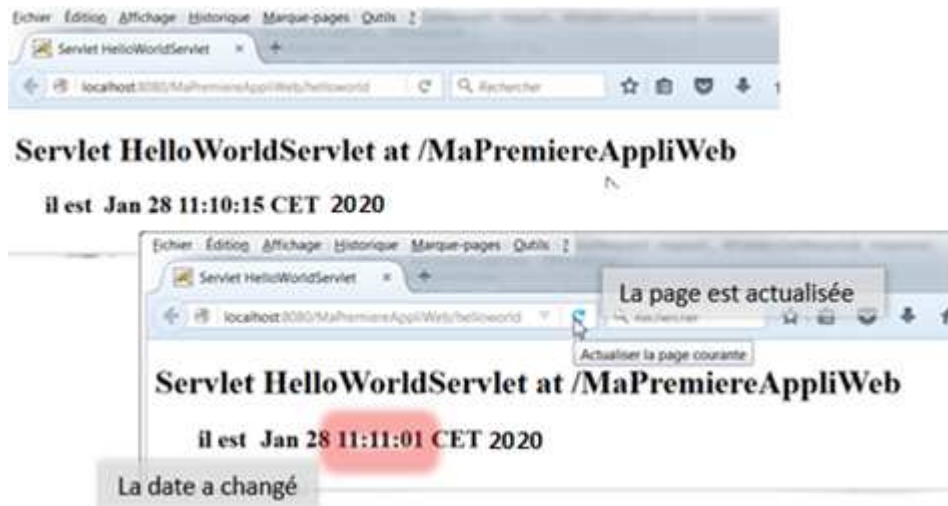


Figure 8: Exécution de la servlet HelloWorldServlet

Nous allons maintenant modifier la servlet HelloWorldServlet pour qu'elle puisse produire une réponse personnalisée en fonction des données qui lui seront transmises depuis un formulaire HTML. Ces données sont:

- un texte à afficher
- le nombre de fois où ce texte doit être affiché

La figure 9 montre le fonctionnement attendu de l'application

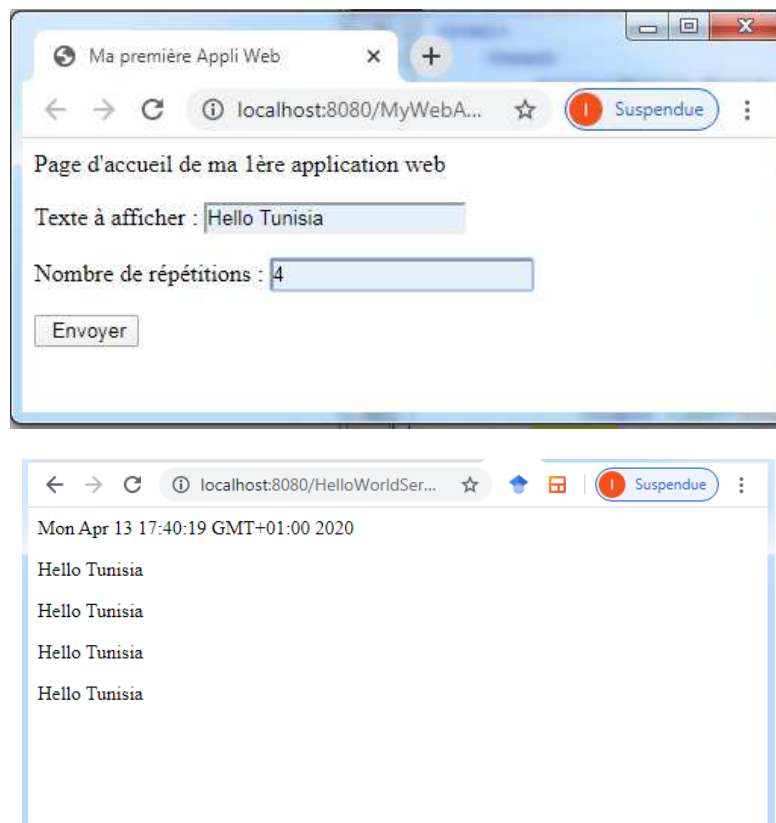


Figure 9: L'application HelloWorld avec un formulaire HTML