

Servlet API

Les servlets

La Servlet API est une brique fondamentale de la spécification JavaEE. Elle permet de développer des composants gérant des requêtes HTTP. L'instanciation des servlets n'est pas gérée par le programmeur, mais par ce que l'on appelle le conteneur de servlets. Il s'agit d'une partie du serveur d'application qui orchestre le cycle de vie des servlets ainsi que les threads nécessaires à la gestion des requêtes.

Pour créer une servlet, il suffit de créer une classe qui étend la classe `HttpServlet` et de lui affecter l'annotation `@WebServlet` avec en paramètre la route qui est gérée par la servlet.

```
@WebServlet("/helloworld")
public class HelloWorldServlet extends HttpServlet {

}
```

Dans une servlet vous pouvez redéfinir certaines méthodes, notamment `doGet` et `doPost`. Ces méthodes sont appelées par le conteneur lorsqu'une requête HTTP est faite sur la route de la servlet. `doGet` gère les requêtes avec la méthode HTTP GET et `doPost` gère les requêtes avec la méthode HTTP POST.

```
@WebServlet("/helloworld")
public class HelloWorldServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

    }
}
```

Les paramètres de ces méthodes, représentent respectivement la requête HTTP et la réponse HTTP.

Objet requête

Paramètres

L'objet requête permet de récupérer des paramètres soit passés dans l'URL de la requête dans le cas d'une requête HTTP GET, ou paramètres d'un formulaire (passés dans le Body de la requête au format FormData) :

```
String monParam = request.getParameter("nomDuParam");
```

Session

L'objet requête permet également de récupérer la session du client. Dans la session, il est possible de stocker des variables et de les récupérer. La session est propre à un client.

```
// Récupérer la session
HttpSession session = request.getSession();

// Ecrire une variable dans la session
session.setAttribute("profil", profil);

// Récupérer une variable de session
ProfilUtilisateur profil = (ProfilUtilisateur) session.getAttribute("profil");
```

Objet réponse

L'objet réponse permet d'écrire du texte dans le Body de la réponse HTTP :

```
response.getOutputStream().println("Hello world");
```

On peut également faire une redirection vers une autre route de l'application :

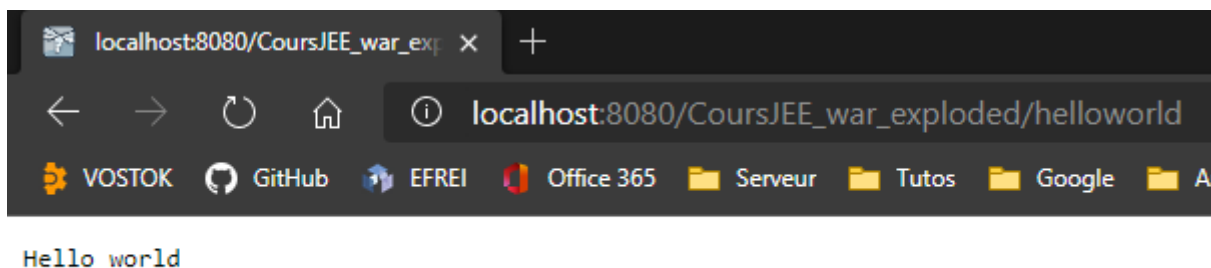
```
response.sendRedirect("/helloworld");
```

Première Servlet

Vous pouvez désormais créer une servlet "Hello world" :

```
@WebServlet("/helloworld")
public class HelloWorldServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        response.getOutputStream().println("Hello world");
    }
}
```

Ensuite lancez le projet, une page web dans votre navigateur devrait apparaître. Ajoutez `/helloworld` à la fin de l'URL et vous devriez voir le message hello world :



Récupération d'un paramètre GET

Pour récupérer un paramètre GET, on utilise la méthode `getParameter` de la requête :

```
@WebServlet("/helloworld")
public class HelloWorldServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        String name = request.getParameter("name");
        response.getOutputStream().println("Hello " + name + "!");
    }
}
```

Relancez votre projet et ajoutez un paramètre GET dans l'URL !

Soumission d'un formulaire

Nous allons maintenant voir comment gérer une soumission de formulaire avec une servlet. Dans le dossier `webapp`, vous devriez trouver une page `index.jsp`. Considérer là comme une simple page HTML, nous verrons les JSP dans la partie d'après. Si elle n'est pas présente créez simplement une page `index.html`. Écrivez maintenant un formulaire POST simple :

```
<form method="post" action="/helloworld">

  <input type="text" name="monParam"/>
  <button type="submit">Envoyer</button>

</form>
```

Dans ce formulaire, l'action est mappée sur l'URL de notre servlet. Il faut donc que notre servlet implémente la méthode `doPost` pour gérer les requêtes POST. On peut ensuite récupérer le paramètre POST de la même façon que les paramètres GET :

```
@WebServlet("/helloworld")
public class HelloWorldServlet extends HttpServlet {

    [...]

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        String name = request.getParameter("monParam");
        response.getOutputStream().println("Hello " + name + "!");
    }
}
```