

TP n°7 V n°1

Titre du TP : Servlet Context Listener Filter CDI

Date : 21/03/2024

Nom : Le Franc

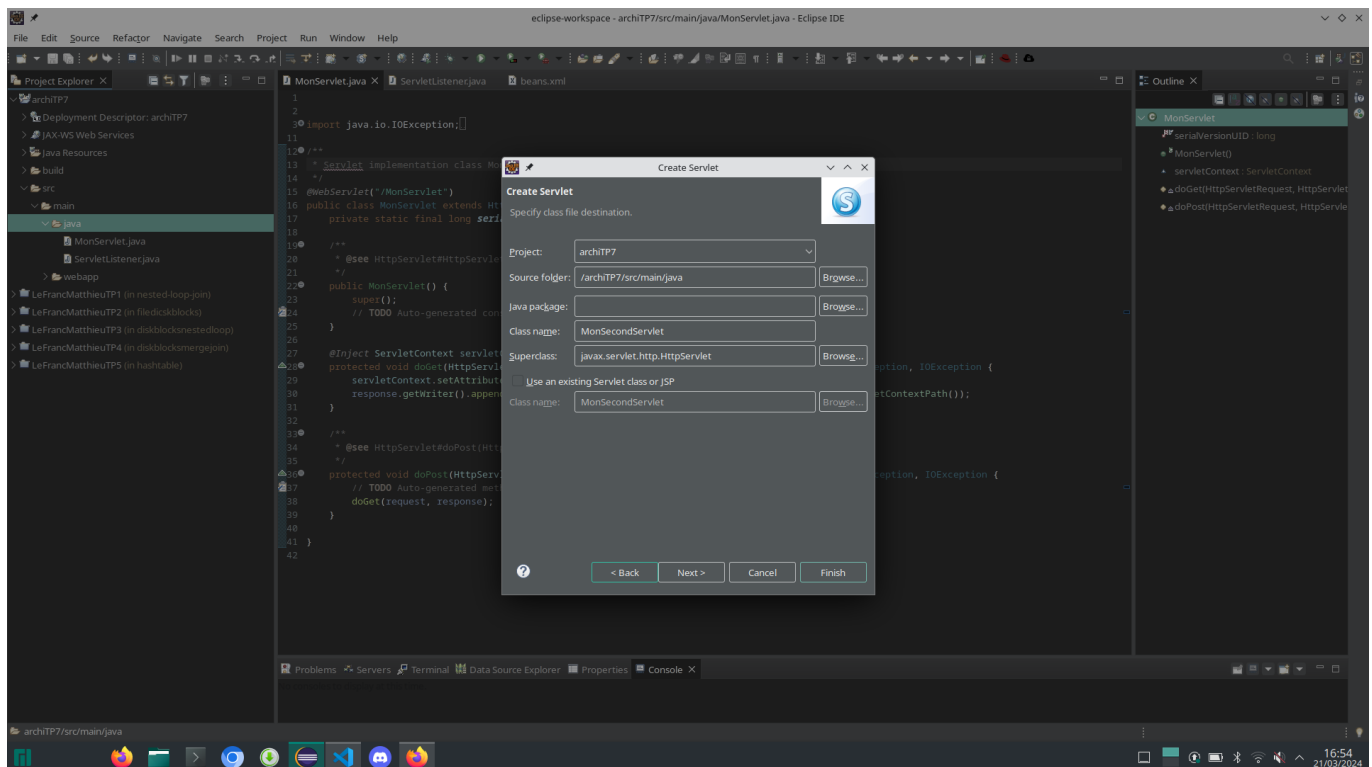
Prénom : Matthieu

N° étudiant : 71800858

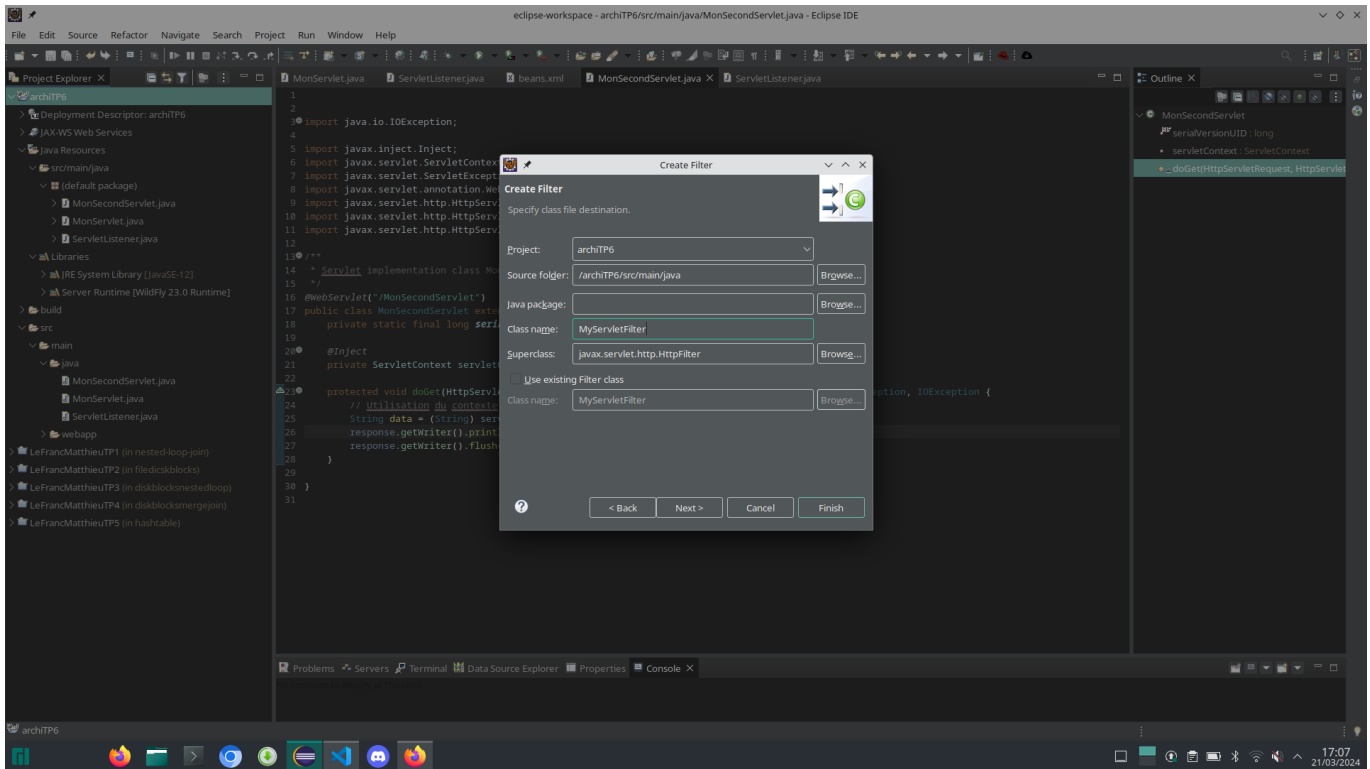
email : matthieu.le-franc@etu.u-paris.fr

Explications

Pour le TP de cette semaine, j'ai repris le TP6 sur Servlet Context Listener. N'ayant pas, dans la version du TP6, ajouté la communication entre Servlet, j'ai également refait cette partie. On crée donc un deuxième Servlet (en plus de celui du TP6).

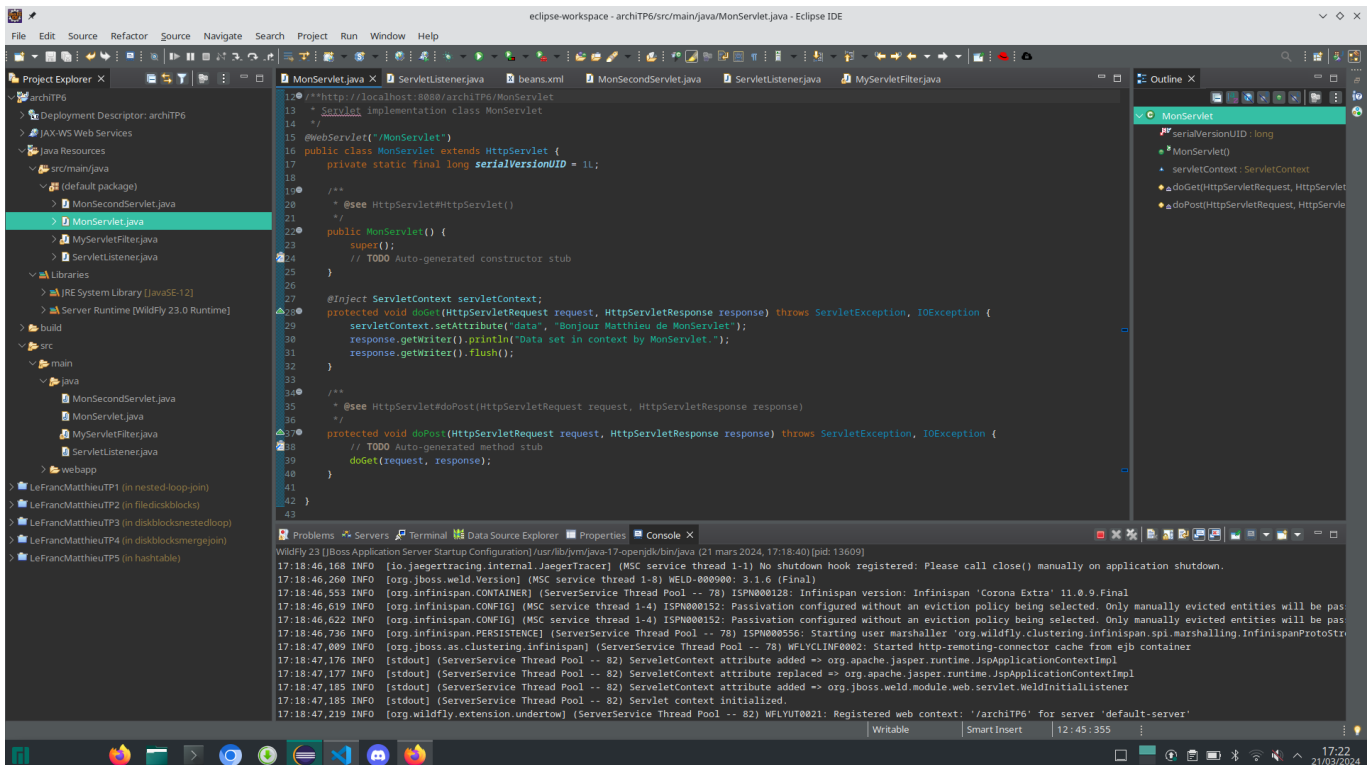


On crée également le Filter nécessaire. Tout se crée assez facilement grâce à eclipse web project donc comme pour les autres composants, on crée la class grâce à l'interface graphique.

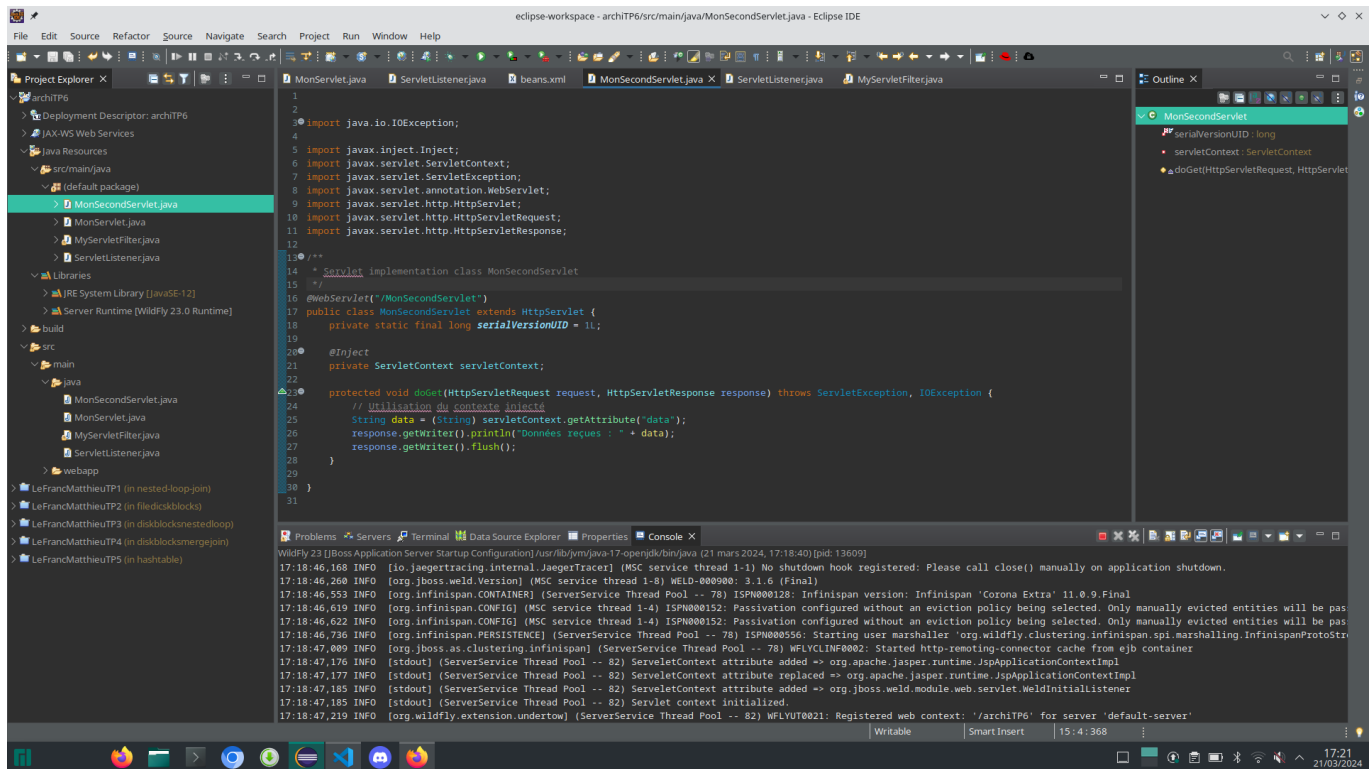


Maintenant, on écrit le code des Servlet pour qu'ils communiquent entre eux et affichent les informations que l'on souhaite.

Premier Servlet

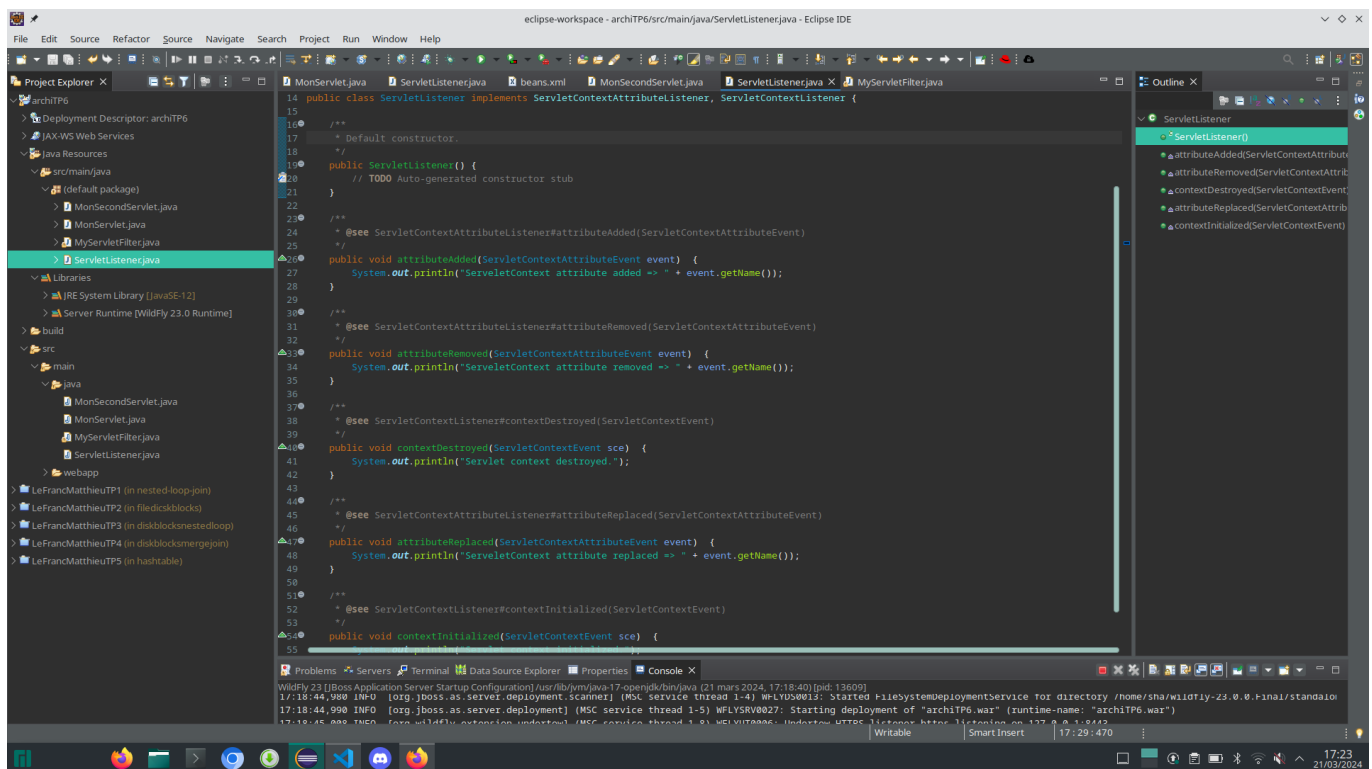


Second Servlet



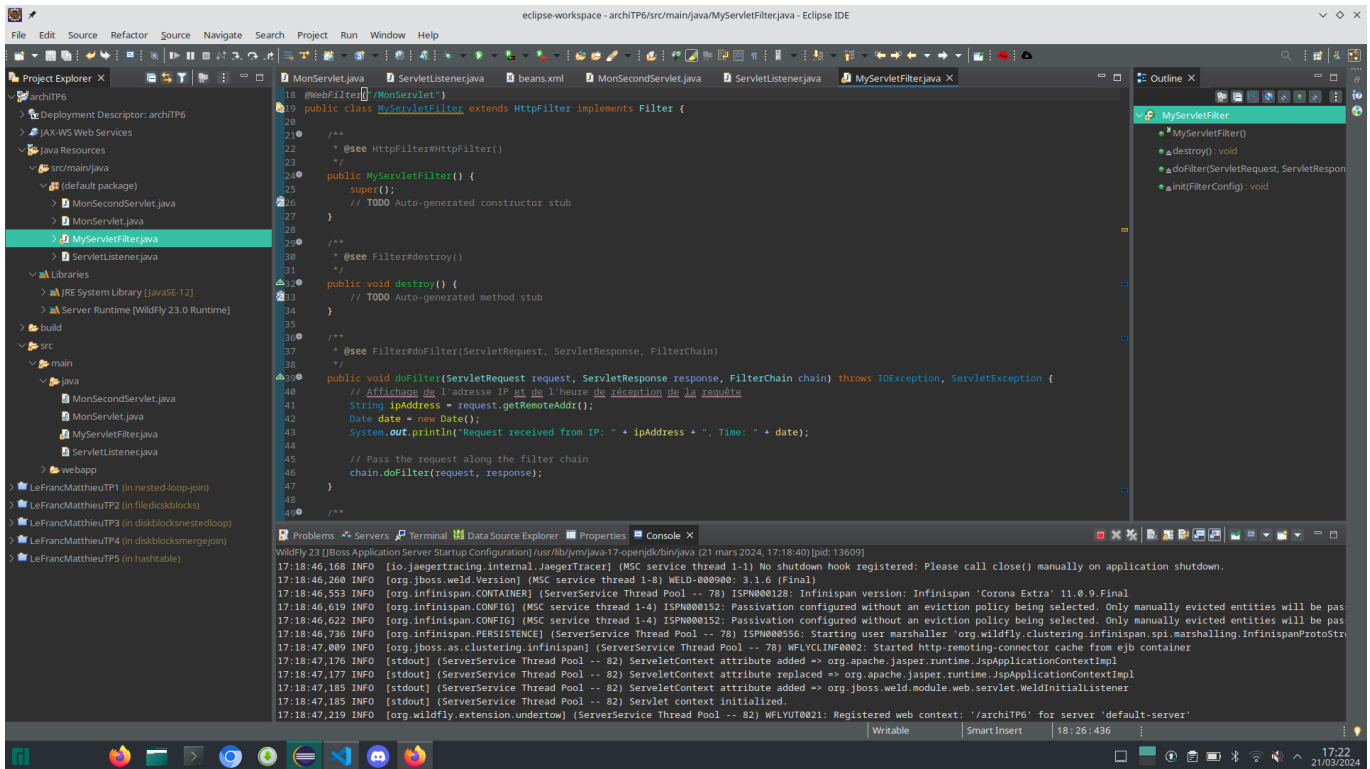
```
1
2
3 import java.io.IOException;
4
5 import javax.inject.Inject;
6 import javax.servlet.ServletContext;
7 import javax.servlet.ServletException;
8 import javax.servlet.annotation.WebServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11
12
13 /**
14  * Servlet implementation class MonSecondServlet
15  */
16 @WebServlet("/MonSecondServlet")
17 public class MonSecondServlet extends HttpServlet {
18     private static final long serialVersionUID = 1L;
19
20     @Inject
21     private ServletContext servletContext;
22
23     /**
24      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
25      */
26     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
27         // Utilisation du contexte injecté
28         String data = (String) servletContext.getAttribute("data");
29         response.getWriter().println("Données reçues : " + data);
30         response.getWriter().flush();
31     }
32 }
```

On écrit aussi l'implémentation des méthodes du Listener que nous avons déjà créé au TP précédent.

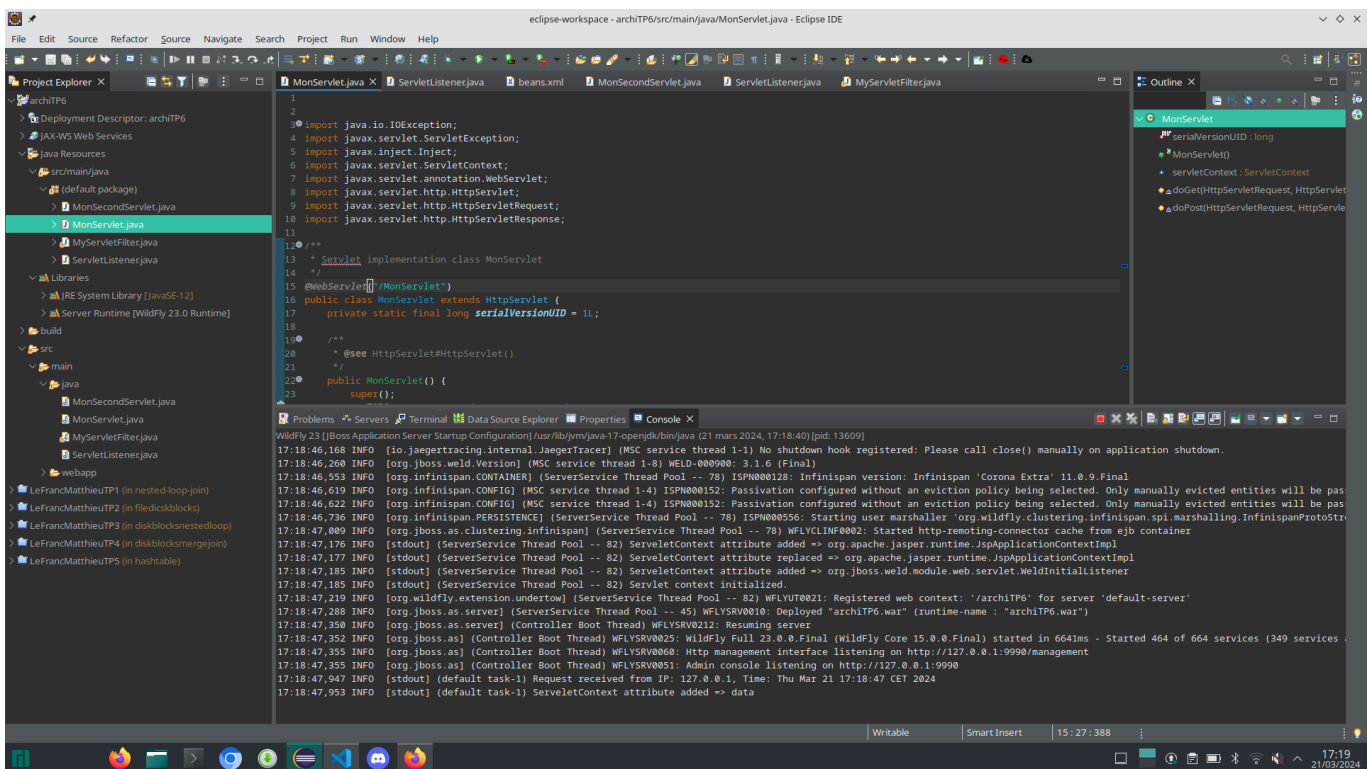


```
1
2
3 public class ServletListener implements ServletContextAttributeListener, ServletContextListener {
4
5     /**
6      * Default constructor.
7      */
8     public ServletListener() {
9         // TODO Auto-generated constructor stub
10     }
11
12     /**
13      * @see ServletContextAttributeListener#attributeAdded(ServletContextAttributeEvent)
14      */
15     public void attributeAdded(ServletContextAttributeEvent event) {
16         System.out.println("ServletContext attribute added => " + event.getName());
17     }
18
19     /**
20      * @see ServletContextAttributeListener#attributeRemoved(ServletContextAttributeEvent)
21      */
22     public void attributeRemoved(ServletContextAttributeEvent event) {
23         System.out.println("ServletContext attribute removed => " + event.getName());
24     }
25
26     /**
27      * @see ServletContextAttributeListener#attributeReplaced(ServletContextAttributeEvent)
28      */
29     public void attributeReplaced(ServletContextAttributeEvent event) {
30         System.out.println("ServletContext attribute replaced => " + event.getName());
31     }
32
33     /**
34      * @see ServletContextListener#contextDestroyed(ServletContextEvent)
35      */
36     public void contextDestroyed(ServletContextEvent sce) {
37         System.out.println("Servlet context destroyed.");
38     }
39
40     /**
41      * @see ServletContextListener#contextInitialized(ServletContextEvent)
42      */
43     public void contextInitialized(ServletContextEvent sce) {
44         // TODO Auto-generated method stub
45     }
46 }
```

Maintenant on ajoute du code au filter. On ajoute l'annotation `@WebFilter` avec comme attribut le pattern de l'url que l'on souhaite filtrer. On implémente la méthode `doFilter()` où on récupère l'adresse ip de `ServletRequest`, la date et on affiche tout ça dans la sortie standard.

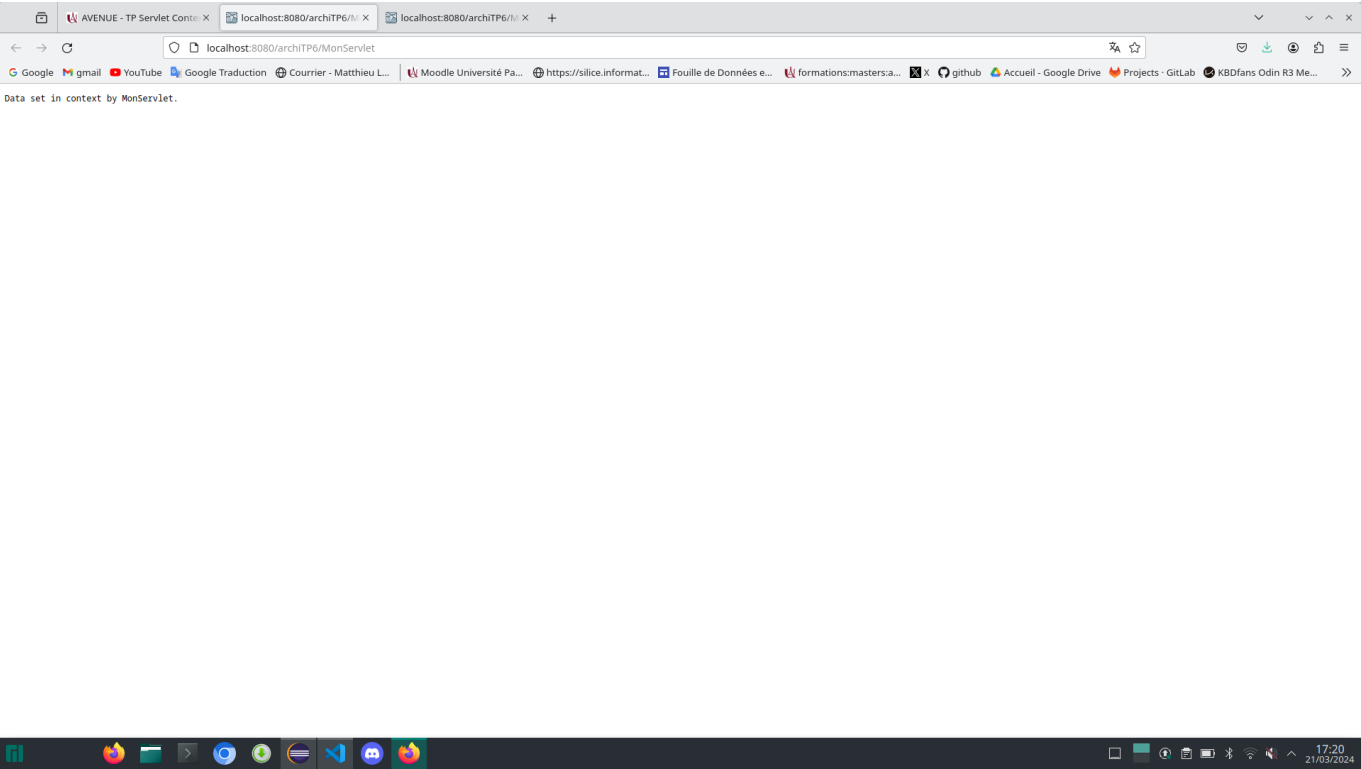


Désormais, on peut lancer le serveur Wildfly et voir que tout fonctionne correctement, ci-dessous les affichages dans la console.

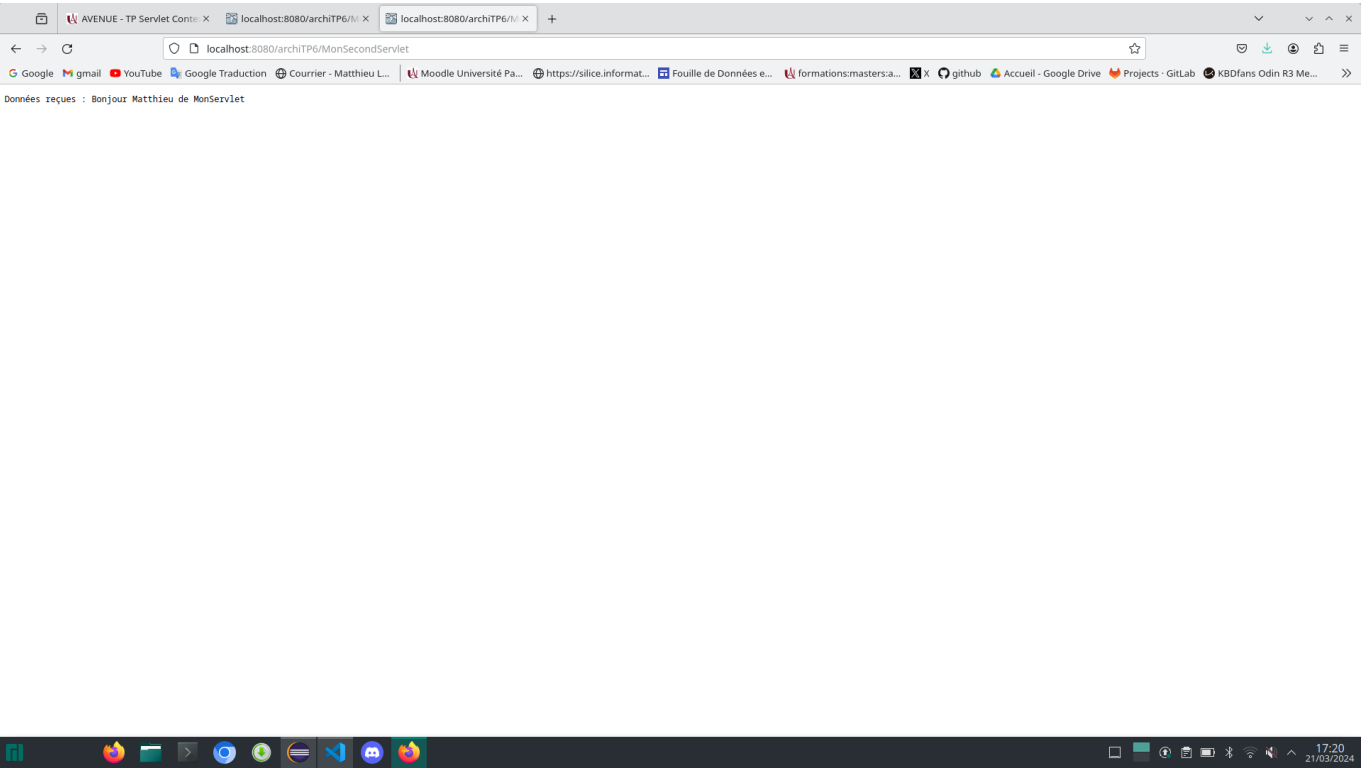


Ci-dessous les affichages des Servlets dans le navigateur.

Premier Servlet



Second Servlet



Rappels des précédents TP

Je ne l'ai pas abordé ici mais j'ai également ajouté un fichier `beans.xml` dans le dossier `WEB-INF` pour activer CDI (Context and Dependency Injection). J'avais fait ceci dans le TP6 pour que ce soit déployable.