

Servlet

Servlet

- [Résumé](#)
- [Servlet classique :](#)
- [Récupération d'informations d'un formulaire](#)
- [Redirection vers une url](#)
- [Chaînage de Servlets](#)
- [Sessions](#)
- [Principes de conception](#)
- [Liaison Servlet-Bean-JSP](#)

Résumé

- Conteneur web
- Les servlets sont exécutés côté serveur.
- Non visible si pas compilé
- Code dans /src/
- Réponds à la requête du client

Servlet classique :

```
@WebServlet("/MonServlet")
public class MonServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public MonServlet() {
```

```

        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
    HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at:
    ").append(request.getContextPath());
        faire(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
    HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}

```

doGet() et *doPost()* ne sont pas utilisées en même temps mais correspondent à l'utilisation ou non du cache.

Récupération d'informations d'un formulaire

Implémentation de la méthode *faire()* qui récupère les éléments du formulaire ainsi que les valeurs associées :

```
private void faire(HttpServletRequest request, HttpServletResponse
response)
{
    //récupération de la liste des paramètres
    Enumeration<String> params =request.getParameterNames();

    while(params.hasMoreElements())
    {
        String key = params.nextElement();
        //affichage des valeurs
        System.out.println(key + ": " +
request.getParameter(key));
    }
}
```

Redirection vers une url

```
public void redirection(String url, HttpServletResponse response)
throws IOException
{
    response.sendRedirect(url);
}
```

Chaînage de Servlets

Utilisation des chainages plutôt que la redirection dans la pratique pour garder la logique métier.
Pour se faire, il faut utiliser le *requestDispatcher*

Servlet1 appelant Servlet2 par inclusion:

```
void service(HttpServletRequest request, HttpServletResponse response)
{
    /*
     * traitement
     */

    ServletContext sc = getServletContext();
    RequestDispatcher rd = sc.getRequestDispatcher("/Servlet2");
    rd.include(request, response);
    //suite des traitements
}
```

Servlet1 appelant Servlet2 par délégation:

```
void service(HttpServletRequest request, HttpServletResponse response)
{
    /*
     * traitement
     */

    ServletContext sc = getServletContext();
    RequestDispatcher rd = sc.getRequestDispatcher("/Servlet2");
    rd.forward(request, response);
    //aucun traitement pour répondre à l'utilisateur, le servlet2 l'a
    fait
}
```

Sessions

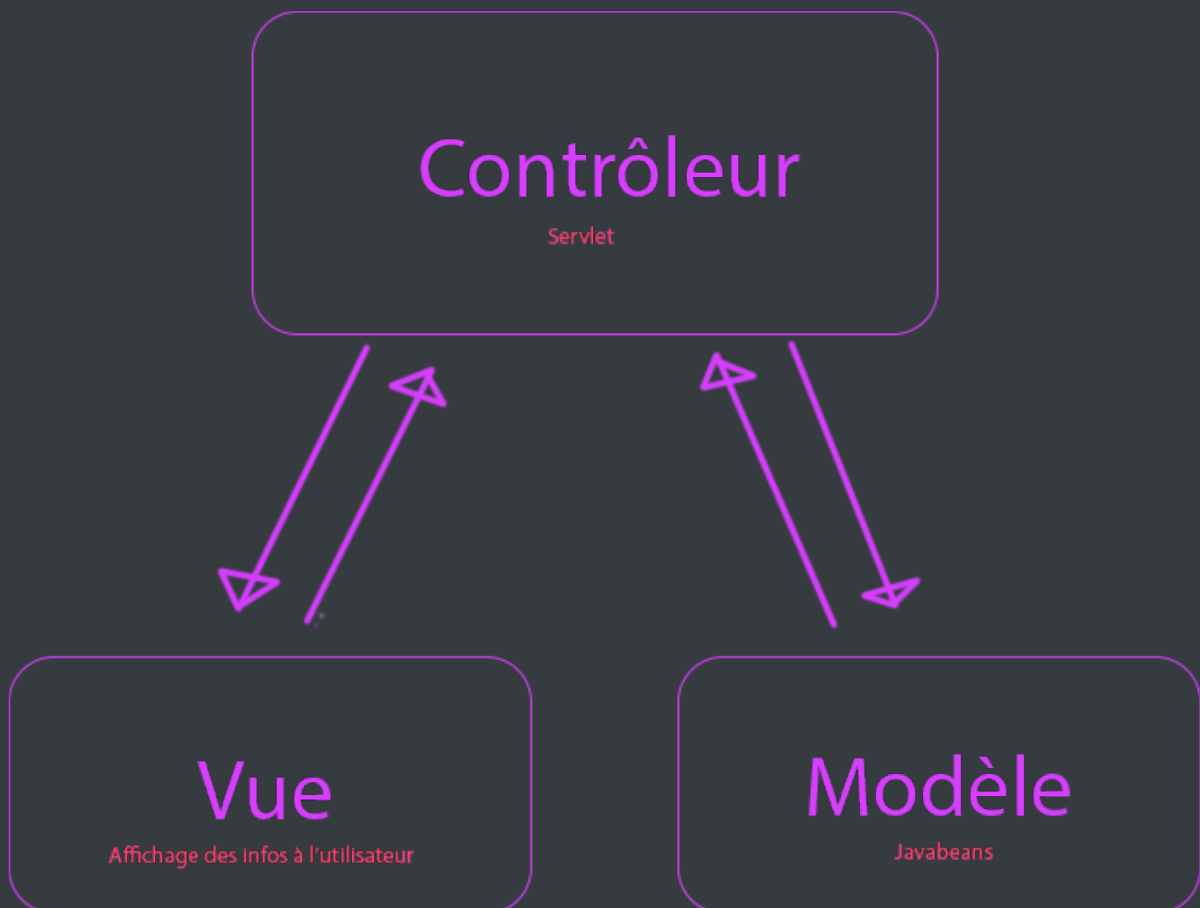
Permettent un maintien d'état entre les requêtes d'un même utilisateur. Chaque session a son propre identifiant et elles sont gérées par un cookie. Deux possibilités de fin : timeout ou déconnexion. (Ex p.127)

Obtention d'une session:

```
//la première fois
HttpSession maSession = req.getSession(true);

//les fois suivantes, la session étant créée
HttpSession maSession = req.getSession();
    //ou
HttpSession maSession = getSession(false);
```

Principes de conception



Liaison Servlet-Bean-JSP

Initialisation du bean dans le servlet. Remplissage du bean avec les données du JSP, puis affichage des résultats dans un autre JSP.

Servlet :

```
//récupération/création d'une session
HttpSession session = request.getSession();
//init du bean
InformationBean bean = new InformationBean();
//traitement sur le bean
bean.setNom(request.getParameter("nom"));
//transfert du bean dans la session
session.setAttribute("bean", bean);
//redirection vers la page d'affichage des resultats
RequestDispatcher rd = request.getRequestDispatcher("/resultat.jsp");
rd.forward(request, response);
```

JSP source :

```
<!-- Appel du servlet précédent dans la methode post -->
<form name="identificationForm" method="post" action="MonServlet">
```

JSP destination :

```
<!-- import du bean -->
<%@page import="beans.InformationBean"%>

    <!-- Instanciation locale du bean avec la valeur du bean de la
    session (nom passé en paramètre de session.setAttribute("nom", objet))
    puis affichage de la valeur -->
    <%
        InformationBean bean =
        (InformationBean)session.getAttribute("bean");
        out.println(bean.getNom());
    %>
```

