

# L'API Servlet



P.Mathieu

IUT-A Lille

<http://www.iut-a.univ-lille.fr>  
prenom.nom@univ-lille.fr

22 octobre 2019

- 1 Web dynamique
- 2 JEE : La Servlet
- 3 Fabriquer une URL
- 4 Le traitement des paramètres

- Génération de pages HTML à la volée
- Génération de pages spécifiques à la personne se connectant
- Génération de pages issues d'un calcul ou fonction de données
- Ex : Commerce électronique, Annuaire, Intranet, etc ...

Principe :

- Le code est exécuté sur le serveur
- Le résultat est envoyé au client
- **Le client ne reçoit que le résultat, pas le code de génération !**

CGI technique classique et "ancestrale"

ASP (VB) ancienne technique Microsoft

PHP

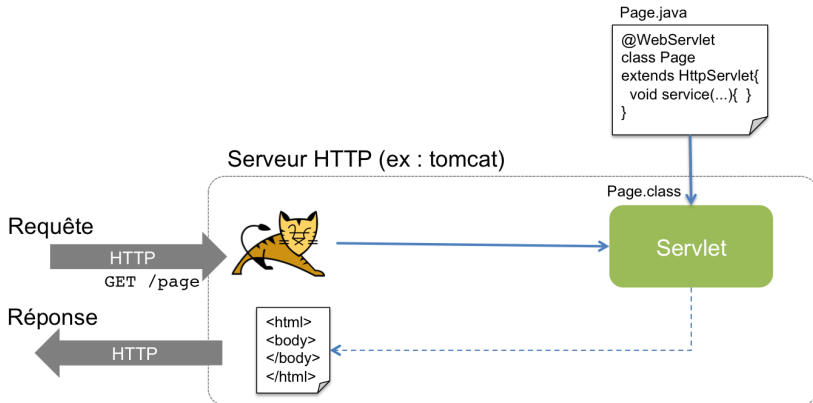
Zope (python)

Rails (Ruby)

JEE (Java)

.NET "dot net" (C#) copie de JEE par Microsoft

...



Définie dans deux paquetages : `javax.servlet` et `javax.servlet.http`

Interfaces	Classes
<code>HttpServletMapping</code> <b><code>HttpServletRequest</code></b> <b><code>HttpServletResponse</code></b> <b><code>HttpSession</code></b> <code>HttpSessionActivationListener</code> <code>HttpSessionAttributeListener</code> <code>HttpSessionBindingListener</code> <code>HttpSessionContext</code> <code>HttpSessionListener</code> ...	<code>Cookie</code> <code>HttpFilter</code> <b><code>HttpServlet</code></b> <code>HttpServletRequestWrapper</code> <code>HttpServletResponseWrapper</code> <code>HttpSessionBindingEvent</code> <code>HttpSessionEvent</code> <code>HttpUtils</code>

Contenu dans `servlet-api.jar` fourni par les serveurs WEB.

1 Web dynamique

2 JEE : La Servlet

3 Fabriquer une URL

4 Le traitement des paramètres

- Une servlet est un simple objet, identifié par une URL
- La servlet est écrite en Java. Elle doit évidemment être compilée.
- Une servlet n'a pas de `main`
- La servlet `extend HttpServlet` et implémente une méthode de service qui renvoie du HTML quand elle est invoquée
- Elle est rangée dans un “contexte”, répertoire qui regroupe les objets d'une même application
- Elle possède un nom public et un nom privé
- L'url d'appel contient `nom du contexte + nom public de la servlet`



```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;

@WebServlet("/servlet-First")
public class First extends HttpServlet {
    public void service( HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        res.setContentType("text/html;charset=UTF-8");
        PrintWriter out = res.getWriter();
        out.println( "<html><body>" );
        out.println( "<h1>Test de ma Servlet</h1>" );
        out.println( "</body></html>" );
    }
}
```

`http://127.0.0.1/contexte/servlet-First`

- importation des paquetages `javax.servlet.*` et `javax.servlet.http.*`
- Une servlet hérite de la classe `HttpServlet`.
- Elle implémente la méthode `doGet` ou `doPost` selon la requête HTTP à traiter  
(La méthode `service` redirige vers `doGet` ou `doPost` en fonction de la requête).
- Deux objets fabriqués par le conteneur sont passés en paramètre à ces méthodes :  
`HttpServletRequest` et `HttpServletResponse`
- La méthode `getWriter()` sur l'objet `HttpServletResponse` fournit le flux de sortie (`getOutputStream` pour du binaire).

```
import java.util.Date;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;

@WebServlet("/servlet-MyDate")
public class MyDate extends HttpServlet
{
    public void service( HttpServletRequest req, HttpServletResponse res )
        throws ServletException, IOException
    {
        res.setContentType("text/html;charset=UTF-8");
        PrintWriter out = res.getWriter();
        out.println( "<html><body>" );
        out.println( "<h1>Nous sommes le " + new Date() + "</h1>" );
        out.println( "</body></html>" );
    }
}
```

`http://127.0.0.1/contexte/servlet-MyDate`

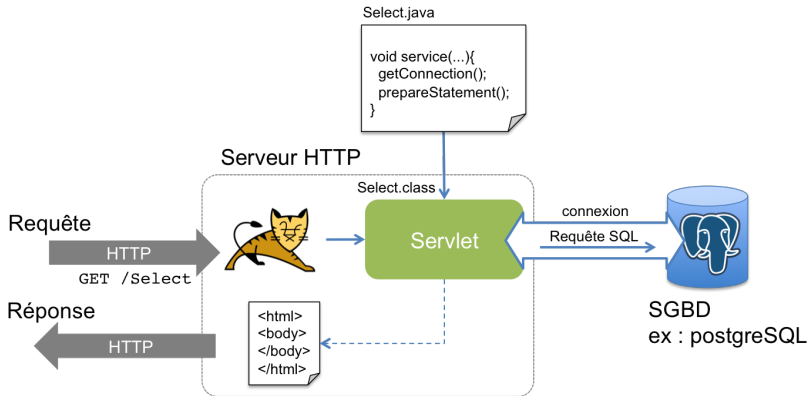
- Adresse absolue (à éviter)

`href=http://localhost/contexte/page.html`

- Adresse relative (indépendante du déploiement)

`href=../page.html`

L'usage d'une URL relative est toujours préférable pour faciliter le déploiement

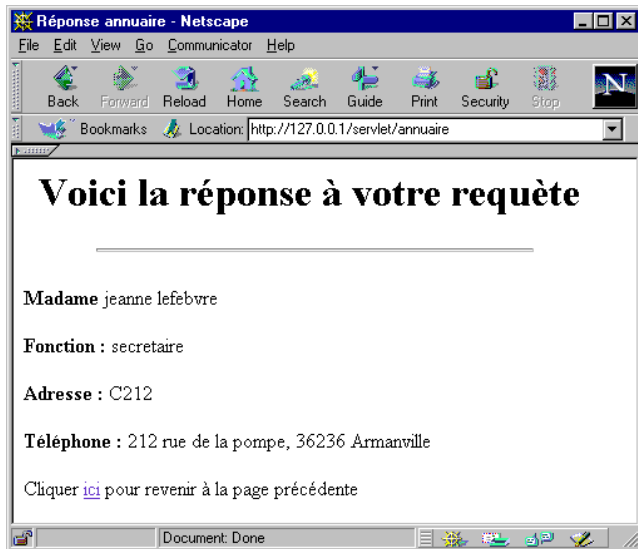


# Servlet + Base de Données

```
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.*;

@WebServlet("/Servlet-Select")
public class Select extends HttpServlet
{
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        res.setContentType("text/html;charset=UTF-8");
        PrintWriter out = res.getWriter();
        out.println( "<html><body>");
        Connection con;
        try { // ... récupération de la connexion
            String query = "select NOM,PRENOM,AGE from CLIENTS";
            PreparedStatement ps = con.prepareStatement(query);
            ResultSet rs = stmt.executeQuery(query);
            while (rs.next())
            {
                String nom = rs.getString("nom");
                String p = rs.getString("prenom");
                int a = rs.getInt("age");
                out.println(n + " " + p + " " + a + "<br>");
            }
            out.println( "</body></html>");
        }
        // ... fermeture et gestion des exceptions
    }
}
```







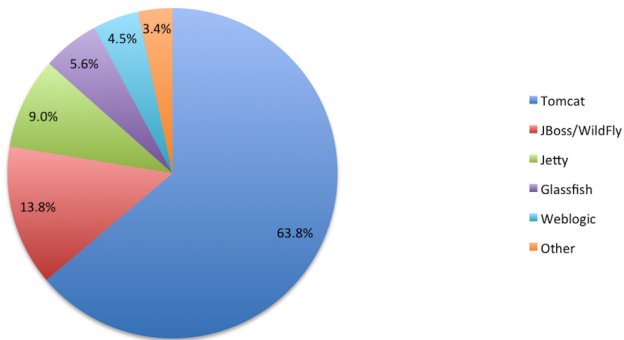
## Serveur JEE de référence réalisé par le consortium Apache

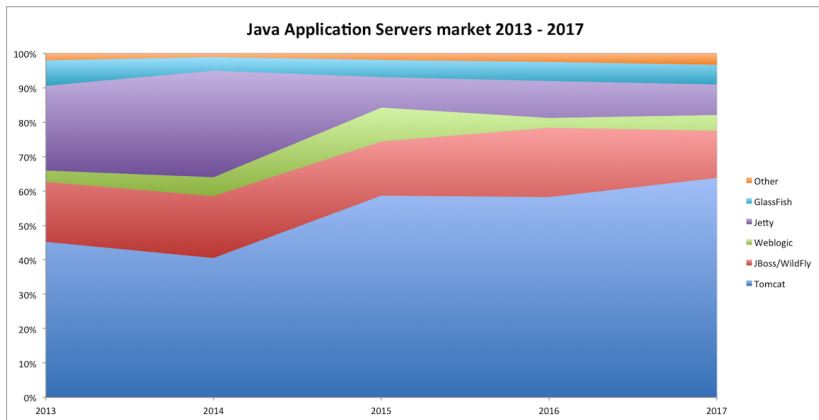
```
Tomcat
  bin
  conf
  work
  lib
    servlet-api.jar
  webapps
    contexte1
      mapage.html
      WEB-INF
        web.xml
        lib
          postgresql.jar
        classes
          First.class
    contexte2
    .....
```

<http://localhost/contexte1/page1.html>

<http://localhost/contexte1/servlet-First>

**Java application server market 2017**





1 Web dynamique

2 JEE : La Servlet

3 Fabriquer une URL

4 Le traitement des paramètres

# Fabriquer une URL

## Trois manières de fabriquer une URL

The screenshot shows the Facebook login page. Three red arrows point from text labels to specific elements on the page:

- Saisie d'URL GET**: Points to the address bar showing the URL `https://fr-fr.facebook.com`.
- Formulaire HTML GET ou POST**: Points to the login form fields, including "Adresse électronique ou téléphone", "Mot de passe", and the "Connexion" button.
- Hyperlien GET**: Points to the link "Créer une Page" at the bottom of the page.

The page content includes the Facebook logo, a welcome message, a registration form, and a link to create a page.

### ● Get

- ▶ Mode d'appel de la ligne d'URL du navigateur ou d'un hyperlien (tag `<A>`).

- ▶ Les paramètres sont concaténés à l'URI :

`http://127.0.0.1/contexte/servlet/comptes?nom=Dupond&prenom=pierre`

- ▶ L'url et ses paramètres sont visibles dans la barre d'adresses

### ● Post

- ▶ Les paramètres ne sont pas visibles de l'utilisateur.
- ▶ Uniquement utilisable à partir d'un formulaire (tag `<Form>`)
- ▶ Adapté à un grand nombre de paramètres

- Le serveur WEB reçoit une requête (GET, POST) de fourniture d'un fichier et renvoie ce fichier

```
> telnet www.google.fr 80  
> GET /index.html HTTP/1.1
```

# Fabriquer une URL

## Derrière le navigateur

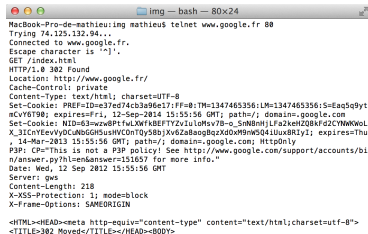
```
MacBook-Pro-de-mathieu:img mathieu$ telnet www.google.fr 80
Trying 74.125.132.94...
Connected to www.google.fr.
Escape character is '^]'.
GET /index.html
HTTP/1.0 302 Found
Location: http://www.google.fr/
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Set-Cookie: PREF=ID=e37ed74cb3a96e17:FF=0:TM=1347465356:LM=1347465356:S=Eaq5q9yt
mCvY6T90; expires=Fri, 12-Sep-2014 15:55:56 GMT; path=/; domain=.google.com
Set-Cookie: NID=63=wzw8PtfwLXWfkBEFTYZvIuloMsv7B-o_SnN8nHjLFa2keHZQ8kFd2CYNWKWoL
X_3ICnYEevVyDCuNbGGH5usHVC0nTQy58bjXv6Za8aogBqzXd0xM9nW5Q4iUux8RIyI; expires=Thu
, 14-Mar-2013 15:55:56 GMT; path=/; domain=.google.com; HttpOnly
P3P: CP="This is not a P3P policy! See http://www.google.com/support/accounts/bi
n/answer.py?hl=en&answer=151657 for more info."
Date: Wed, 12 Sep 2012 15:55:56 GMT
Server: gws
Content-Length: 218
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN

<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
```



- Le serveur WEB reçoit une requête (GET, POST) de fourniture d'un fichier et renvoie ce fichier

```
> telnet www.google.fr 80
> GET /index.html HTTP/1.1
```



```
MacBook-Pro-de-mathieu:img mathieu$ telnet www.google.fr 80
Trying 74.125.132.94...
Connected to www.google.fr.
Escape character is '^]'.
GET /index.html
HTTP/1.0 302 Found
Location: http://www.google.fr/
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Set-Cookie: PREF=ID=e37ed74cb3a96e17:FF=0;TM=1347465356;S=Eaq5q9yt
mCvY6T90; expires=Fri, 12-Sep-2014 15:55:56 GMT; path=/; domain=.google.com
Set-Cookie: NID=63=wz8PtfuLXWfkBEFTYZvlu0MsV78-o_SnN8nHjLfazketZQ8kfD2CYmKw0L
X_1ICnYevVy0Cn0G8SusHVConT0y58b;Xv6ZaBaqbzXdoMnWS04iux8R1y; expires=Thu
, 14-Mar-2013 15:55:56 GMT; path=/; domain=.google.com; HttpOnly
P3P: CP="This is not a P3P policy! See http://www.google.com/support/accounts/bi
n/answer.py?hl=en&answer=151657 for more info."
Date: Wed, 12 Sep 2012 15:55:56 GMT
Server: gws
Content-Length: 218
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN

<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
```

- Le client WEB (Navigateur) interprète ce fichier en fonction d'un type MIME.

```
<html>
<form action=Servlet-Bonjour method=get>
  Saisir votre nom <input name=n type=text> <br/>
  <input type=submit>
</form>
</html>
```

Ce formulaire génère une requête

`http://localhost/contexte/Servlet-Bonjour?n=paul`

# Fabriquer une URL

## Les différents champs de saisie

Trois types de champs de saisie :

- `<input>`
- `<select>`
- `<textarea>`

Nombreux input types de saisie :

text, password, datetime, datetime-local, date, month, time, week, number, email, url, search, tel, color, file, checkbox, radio, ....

Deux input types de boutons : submit, reset

**Dans tous les cas, le paramètre est passé par une chaîne de caractères HTTP**

- 1 Web dynamique
- 2 JEE : La Servlet
- 3 Fabriquer une URL
- 4 Le traitement des paramètres

# Le traitement des paramètres

## La Servlet de traitement

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;

@WebServlet("/Servlet-Bonjour")
public class Bonjour extends HttpServlet
{
    public void service( HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        res.setContentType("text/html;charset=UTF-8");
        PrintWriter out = res.getWriter();
        out.println( "<html><body>" );
        out.println( "<h1>Bonjour " + req.getParameter("n") + "</h1>" );
        out.println( "</body></html>" );
    }
}
```

Si le paramètre n'est pas présent dans la requête, `getParameter` renvoie la valeur `null`

# Le traitement des paramètres

## Trois méthodes de traitement

- `public void doGet(...)`  
répond aux requêtes HTTP de type GET
- `public void doPost(...)`  
répond aux requêtes HTTP de type POST
- `public void service(...)`  
qui appelle `doGet` ou `doPost` en fonction de la requête HTTP reçue

Redéfinir `service` empêche d'utiliser `doGet` ou `doPost`

# Le traitement des paramètres

## L'objet `HttpServletRequest`

### Méthodes principales

- `getContentType()` Type MIME de requete
- `getRemoteAddr()` IP du client
- `getMethod()` type de requete réseau
- `getRemoteUser()` login de l'utilisateur sur le poste client
- `getParameter(String)` lecture d'un paramètre
- `getHeader()` entêtes HTTP
- `getParameterNames()` lecture des noms de tous les paramètres
- `getParameterValues(String)`
- ...

# Le traitement des paramètres

## Anonymat.org

### Vos traces

Voici quelques informations qu'il est possible de collecter sur vous lorsque vous surfez sur Internet...

Vous êtes connecté à Internet avec l'adresse IP :

**91.162.91.146**

Via le serveur de votre fournisseur d'accès (ou le proxy) :

**91-162-91-146.subs.proxad.net**

Votre navigateur et votre système d'exploitation sont :

**Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:69.0) Gecko/20100101 Firefox/69.0**

En clair, votre navigateur est :

**Mozilla Firefox**

**JavaScript est activé  
Java est désactivé**

Adresse de la page web visitée avant cette page :

**Inconnue**

Nombre de pages visitées durant cette session de navigation :

**3**

En clair, le système d'exploitation de votre ordinateur est :

**Mac OS**

L'adresse de votre ordinateur indique :

### Les Sites

Messageries anonymes  
Messageries sécurisées  
Surf anonyme & proxies  
Testez votre anonymat  
Services distants

### Les Softs

Messageries anonymes  
Messageries sécurisées  
Solutions d'anonymat  
Outils de chiffrement  
Élimination des spywares  
Effacement sécurisé  
Disimulation système  
Gestionnaires de cookies  
Firewalls  
Proxies  
Stéganographie  
Utilitaires pour experts  
Utilitaires divers



 Imprimez cette page



Testé et approuvé !

ACCUEIL

EDITO

ACTUALITÉ

VOS TRACES

ANNUAIRE

OUTILS

NEWSLETTER

E-mail 

RECHERCHE



FORUM

CORRECTIFS



# Le traitement des paramètres

## Lire l'ensemble des entêtes HTTP

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
import javax.servlet.annotation.WebServlet;

@WebServlet("/servlet-Info")
public class Info extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        res.setContentType("text/html;charset=UTF-8");
        PrintWriter out = res.getWriter();
        out.println("<HTML><BODY>");
        out.println("<table border=1>");
        Enumeration enTetes = req.getHeaderNames();
        while (enTetes.hasMoreElements())
        {
            String enTete = (String) enTetes.nextElement();
            out.println("<tr><td>" + enTete
                + "</td><td>" + req.getHeader(enTete)
                + "</td></tr>");
        }
        out.println("</table>");
        out.println("</HTML>");
    }
}
```

# Le traitement des paramètres

## La redirection

Parfois une Servlet n'exécute que du code, sans rien afficher

### Deux méthodes de redirection

- Via le client

```
response.sendRedirect(url)
```

- Interne au serveur

```
RequestDispatcher.forward(req, res)
```

```
RequestDispatcher rd=request.getRequestDispatcher("chemin");  
rd.forward(request, res);
```

# Une servlet d'insertion

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;

@WebServlet("/Servlet-Insert")
public class Insert extends HttpServlet
{
    public void service( HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        String n = req.getParameter("n");
        ....
        String query = "insert into users(nom) values (?)";
        PreparedStatement ps = con.prepareStatement(query);
        ps.setString(1,n);
        ps.executeUpdate();
        res.sendRedirect("Servlet-Select");
    }
}
```

# Le traitement des paramètres

L'objet `HttpServletResponse`

Méthodes principales :

- `getWriter()` pointeur vers un flux texte `PrintWriter`
- `getOutputStream()` pointeur vers un flux binaire
- `sendRedirect(url)` renvoie l'ordre au navigateur de recharger la page passée en paramètre (en Get)
- .....

# Le traitement des paramètres

## Traitement des Checkbox et RadioButton

- `radio` : exclusif
- `checkbox` : inclusif

```
<form action=servlet-Enregistrer method=get>
<p>fraise <input name=f1 type=radio value=fraise>
<p>poire  <input name=f2 type=radio value=poire>
<p>pomme  <input name=f3 type=radio value=pomme>
<input type=submit>
</form>
```

- Problème : difficile à gérer avec `getParameter()`
- C'est une mauvaise solution

# Le traitement des paramètres

## Traitement des Checkbox et RadioButton

```
<form action=servlet-Enregistrer method=get>
<p>fraise <input name=fruit type=checkbox value=fraise>
<p>poire  <input name=fruit type=checkbox value=poire>
<p>pomme  <input name=fruit type=checkbox value=pomme>
<input type=submit>
</form>
```

- avec Radio un seul parametre fruit (celui-choisi)  
getParameter()
- avec Checkbox plusieurs fois fruit=valeur  
getParameterValues()

# Le traitement des paramètres

## L'encodage

- L'encodage du fichier source (paramétré par votre éditeur)
- L'encodage indiqué dans l'entête HTTP pour le navigateur

```
resp.setContentType("text/html;charset=UTF-8");
```

A faire impérativement avant le `getWriter`

- L'encodage indiqué par le navigateur quand il interprète la page. n'est utilisé QUE si pas d'indication dans l'entête HTTP

```
<meta equiv="Content-Type" content="text/html;charset=UTF-8">
```

A placer impérativement avant `title`

**Préférez l'UTF-8 partout !**