



Java avancé

Les Servlets

Objectifs

- Comprendre le rôle d'une Servlet La Servlet s'appelle un contrôleur
- Comprendre le cycle de vie d'une Servlet
- Savoir exploiter une requête HTTP
- Savoir générer une réponse HTTP



Les Servlets

- C'est une classe Java

jmdoudou:

Une servlet est un programme qui s'exécute côté serveur en tant qu'extension du serveur.

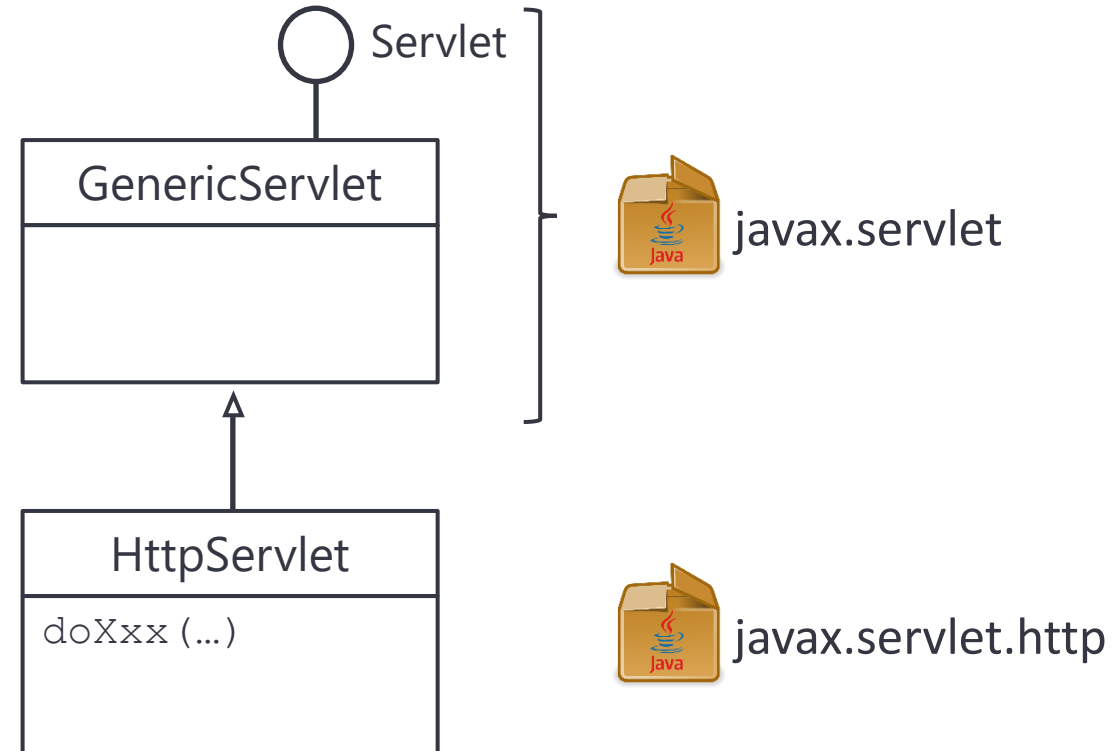
Elle reçoit une requête du client, elle effectue des traitements et renvoie le résultat.

La liaison entre la servlet et le client peut être directe ou passer par un intermédiaire comme par exemple un serveur http.



Les Servlets

- C'est une classe Java



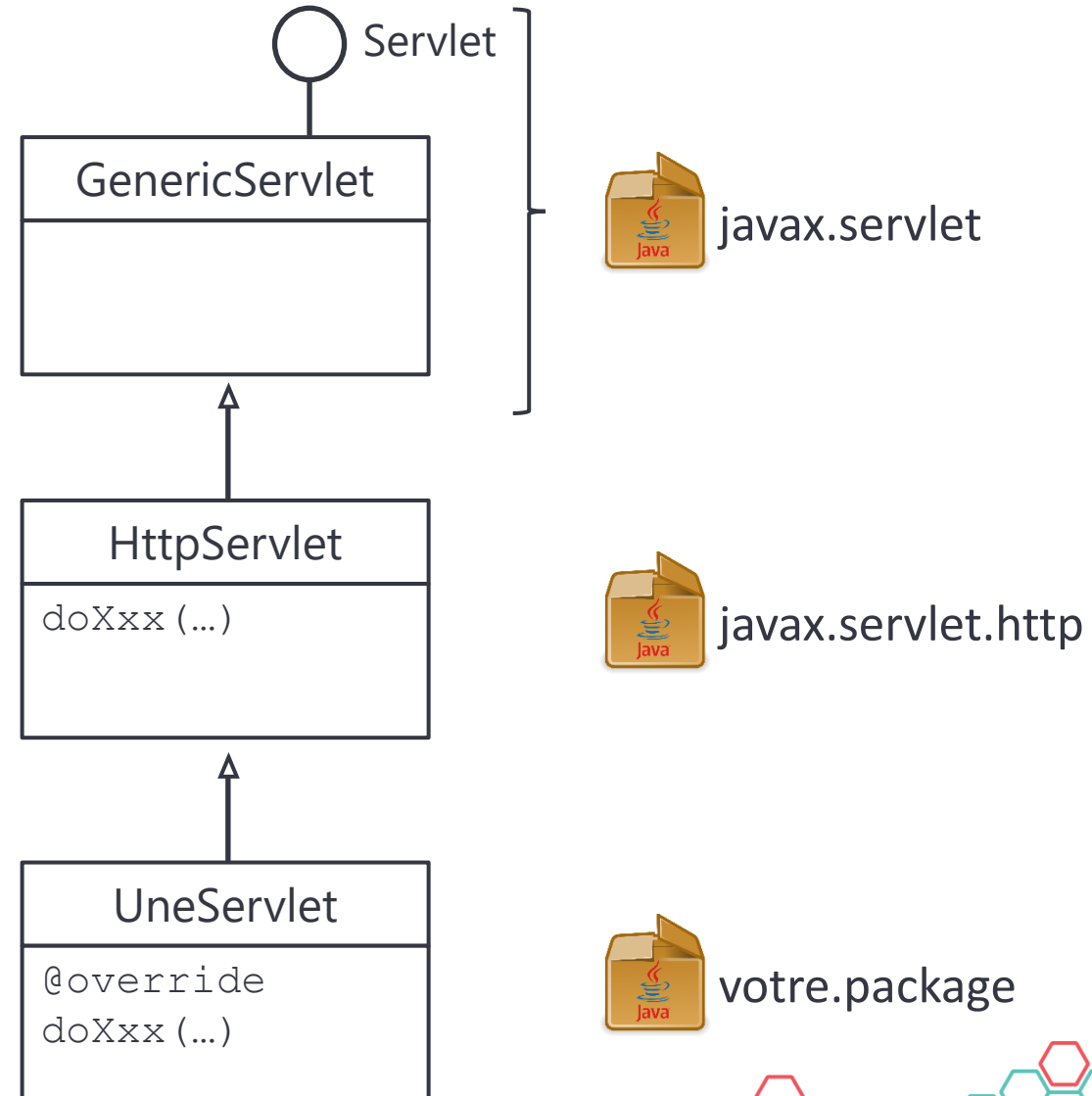
mail: souvent FMTP

UDP: contrairement aux requêtes HTTP: l'UDP ne vérifie pas qu'on a bien reçu la réponse du serveur (moins sécurisé)



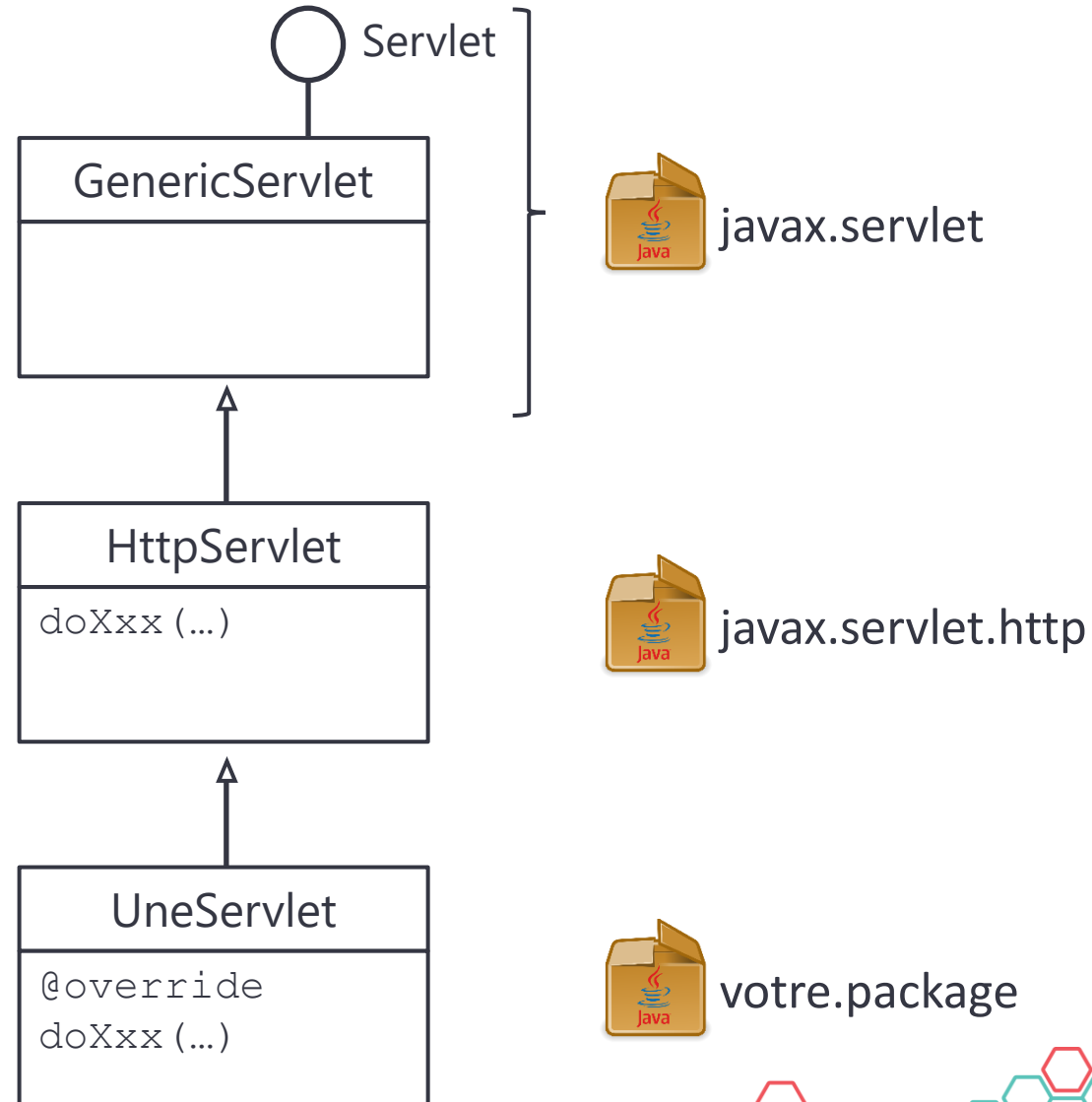
Les Servlets

- C'est une classe Java



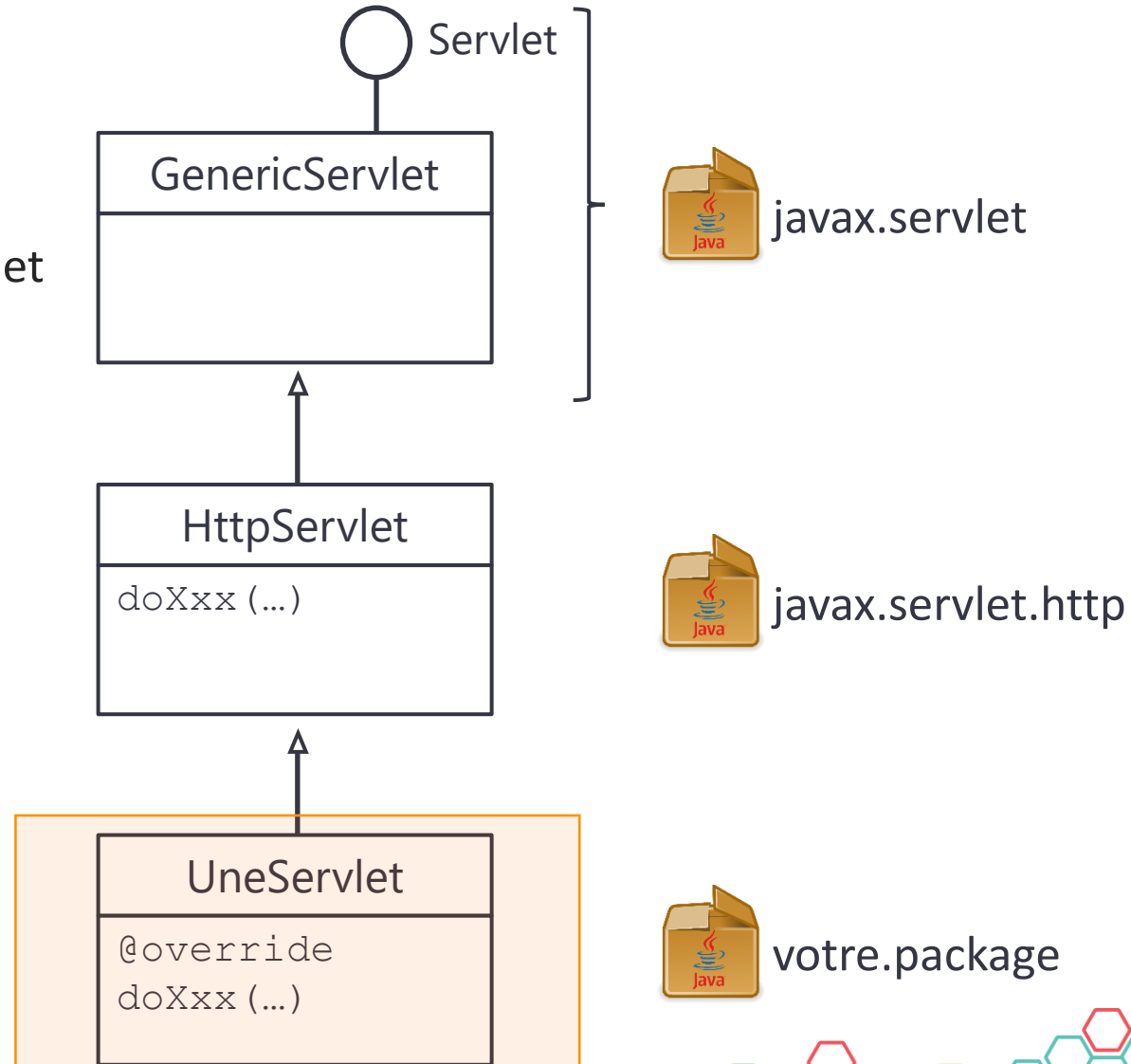
Les Servlets

- C'est une classe Java
- Nos propres Servlets hériteront systématiquement de HttpServlet



Les Servlets

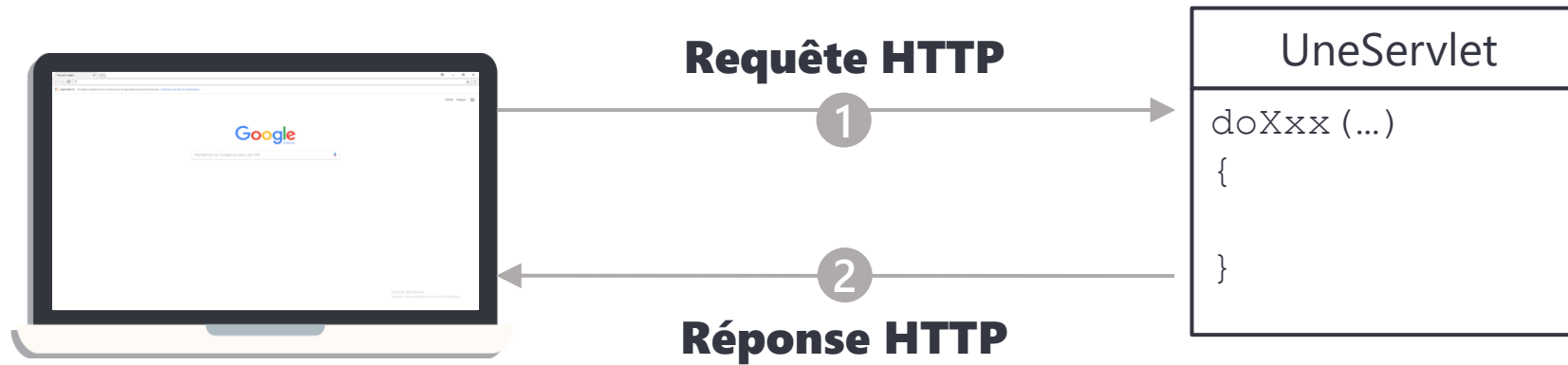
- C'est une classe Java
- Nos propres Servlets hériteront systématiquement de HttpServlet



Les méthodes « doGet » et « doPost » permettent de réagir aux requêtes HTTP de type GET et POST respectivement.

permet 98% des traitements

Les Servlets



La création d'une servlet

```
public class UneServlet extends HttpServlet  
{
```

throws permet de lancer une exception dans Java

```
    @Override
```

```
    protected void doGet(...) throws ServletException, IOException
```

```
    {
```

```
        //Générer la réponse à une requête de type GET
```

```
    }
```

```
    @Override
```

```
    protected void doPost(...) throws ServletException, IOException
```

```
    {
```

```
        //Générer la réponse à une requête de type POST
```

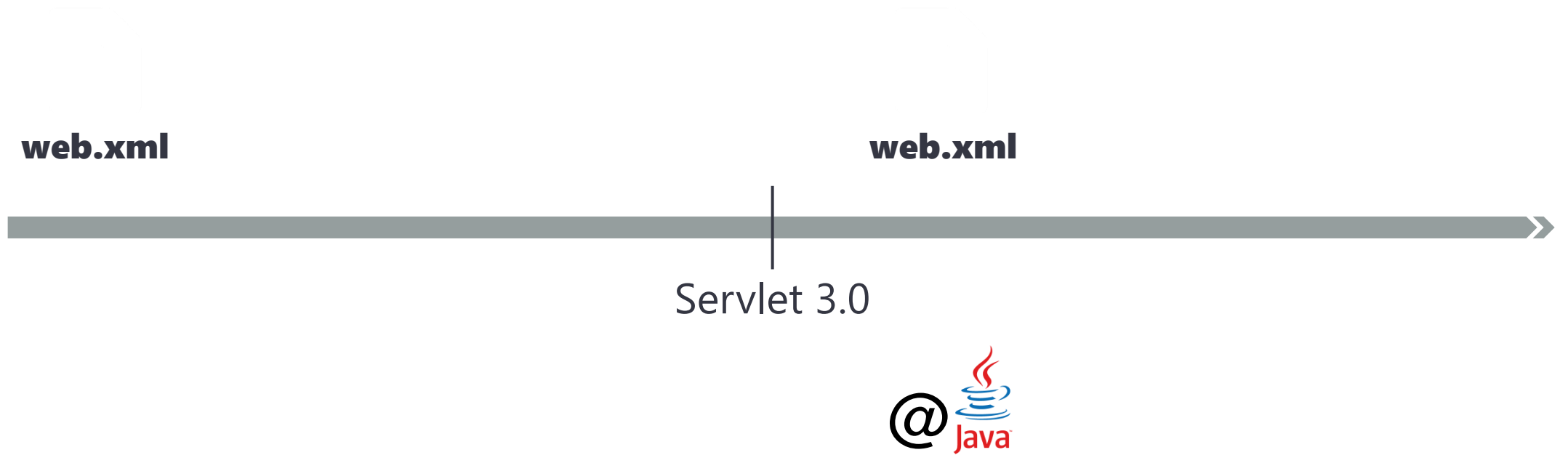
```
    }
```

```
}
```



Le paramétrage de l'URL

- Les servlets



Le paramétrage dans le web.xml

`<!-- Déclaration d'une servlet -->`

`<servlet>`

`<servlet-name>UneServlet</servlet-name>`

`<servlet-class>fr.eni.demo.servlets.UneServlet</servlet-class>`

`</servlet>`

`<!-- Association de la servlet à une ou des URL -->`

`<servlet-mapping>`

`<servlet-name>UneServlet</servlet-name>` même nom que ci-dessus

`<url-pattern>/url/de/la/servlet</url-pattern>` * on commence tout le temps par un /

`<url-pattern>...</url-pattern>` l'url dans les dernières c'est peut-être différent

`</servlet-mapping>`



* utilisation du caractère joker

`<url-pattern>/debut/url/*</url-pattern>`

rarement utilisé: redirection tjrs vers la même page



Le paramétrage par annotation

```
@WebServlet
(  
    name="UneServlet",  
    urlPatterns={"/url/de/la/servlet", "..."}  
)  
public class UneServlet extends HttpServlet  
{  
    ...  
}
```

tableau de string (les différentes url)

on aura une servlet par page

```
@WebServlet("/debut/url/*")  
public class UneServlet extends HttpServlet  
{  
    ...  
}
```



Démo

Notre première Servlet

principe des servlet:

SRP:

Single

Responsability

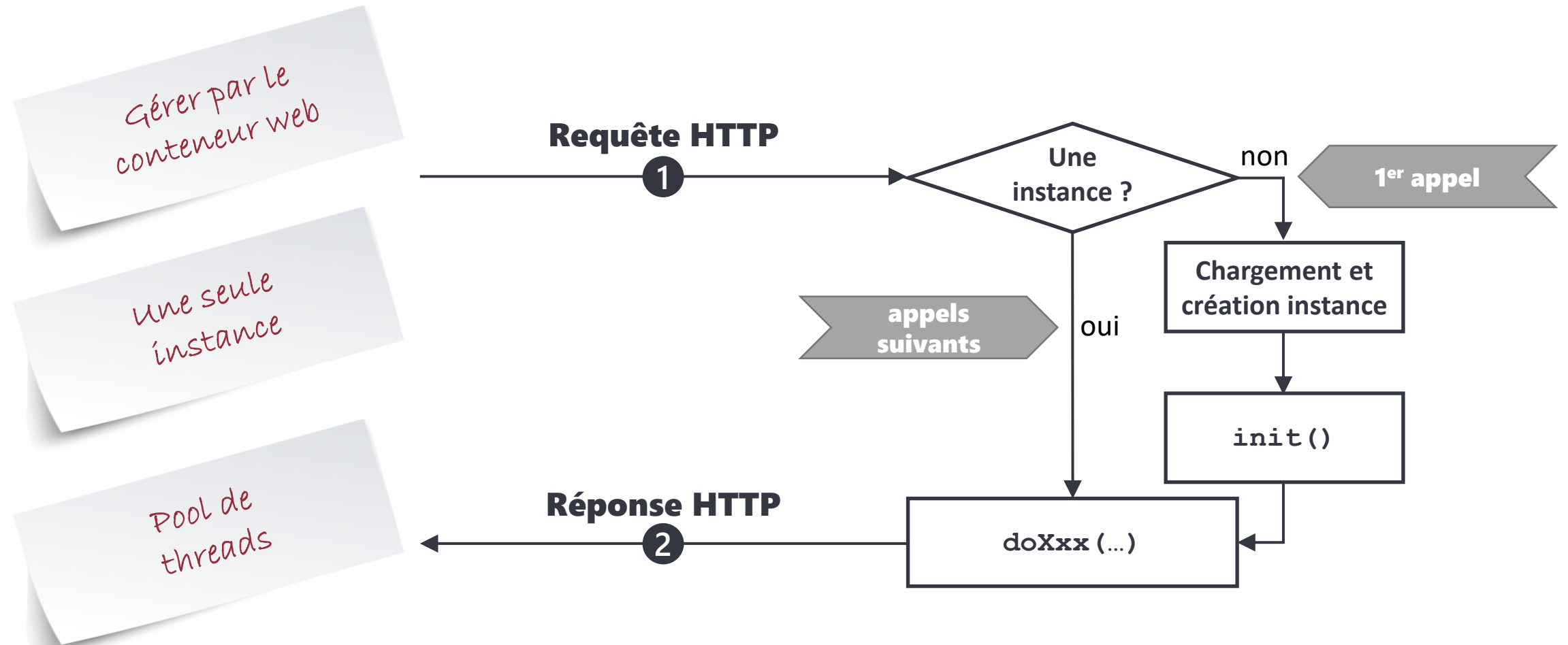
Principle

(une classe n'a qu'un seul rôle)

une servlet ne renvoie que
vers un url dans 98% des cas



Le cycle de vie d'une servlet

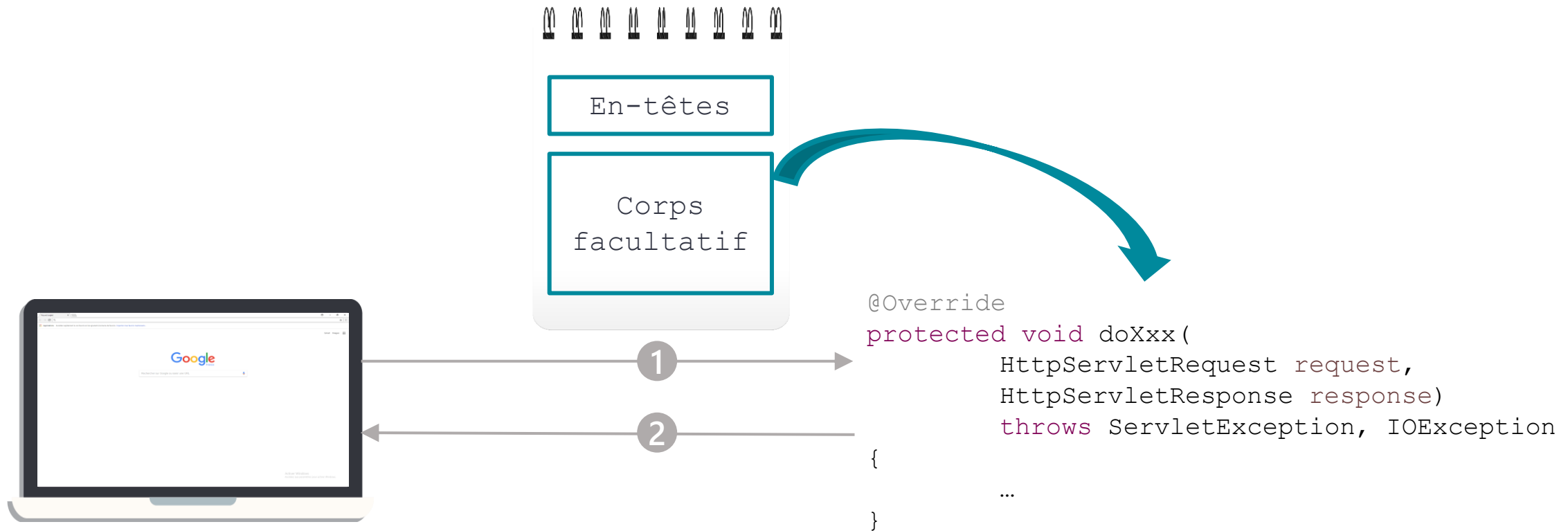


Démo

Cycle de vie d'une Servlet



Lien requête HTTP → Servlet



La lecture de la requête : l'URL

`http://www.exemples.fr:8080/DemoJavaEE/url/de/la/servlet`

HttpServletRequest
<<interface>>

```
getScheme():String  
getServerName():String  
getServerPort():int  
getContextPath():String  
getServletPath():String
```

nom du projet (le + important)

le / du Servlet



La lecture de la requête : les en-têtes principaux



```
POST /docs/ouvrage HTTP/1.1
Host: www.exemples.fr
Accept-Language: en-US
...
```

```
nom=Java%20EE&auteur=
Etienne%20CASSIN
```

HttpServletRequest
<<interface>>

```
getCharacterEncoding():String
getContentType():int
getContentType():String
getLocale():Locale
getMethod():String
...
```



La lecture de la requête : tous les en-têtes



```
POST /docs/ouvrage HTTP/1.1  
Host: www.exemples.fr  
Accept-Language: en-US  
...
```

```
nom=Java%20EE&auteur=  
Etienne%20CASSIN
```

HttpServletRequest <<interface>>

```
getHeader(String name):String  
getDateHeader(String name):Date  
getIntHeader(String name):int  
getHeaders(String name) Header de la requête HTTP  
    :Enumeration<String>  
getHeaderNames()  
    :Enumeration<String>  
...
```



La lecture de la requête : les paramètres



```
POST /docs/ouvrage HTTP/1.1
Host: www.exemples.fr
Accept-Language: en-US
...
```

```
nom=Java%20EE&auteur=
Etienne%20CASSIN
```

très important: `getParameter()`
permet de récupérer les info d'un formulaire par exemple

HttpServletRequest <<interface>>

```
getParameter(String name)           le seul vraiment utile
    :String
getParameterValues(String name)
    :String[]
getParameterNames()
    :Enumeration<String>
getParameterMap()
    :Map<String, String[]>
```

HashMap=dictionnaire

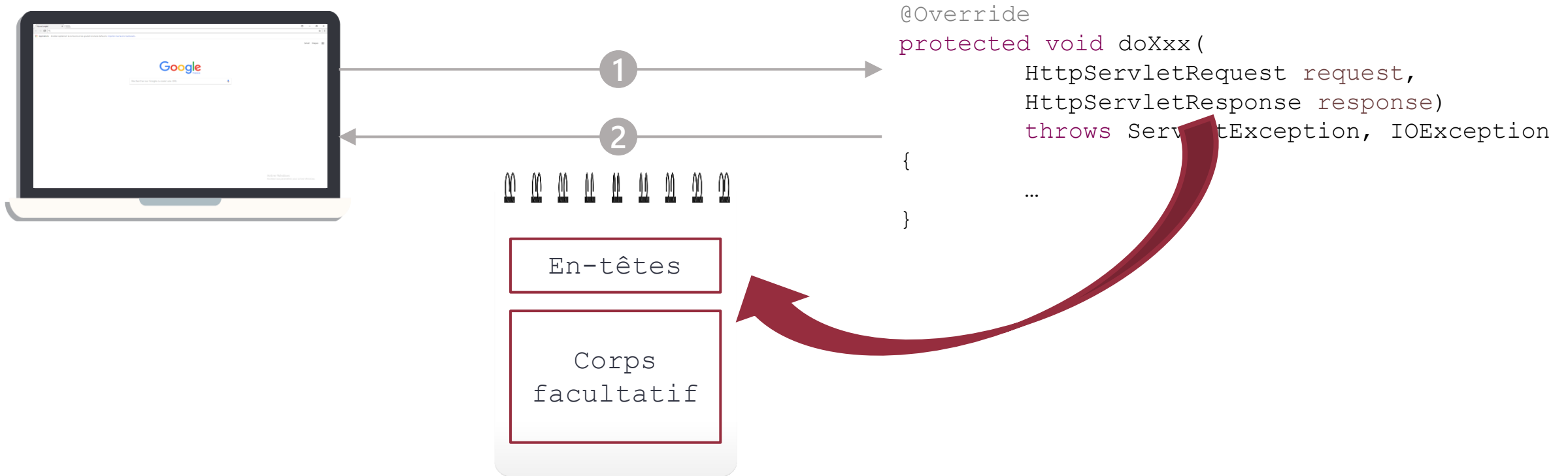


Démo

La lecture de la requête (form)



Lien servlet → réponse HTTP



L'écriture de la réponse : les en-têtes principaux



```
HTTP/1.1 200 OK  
Date: Mon, 25 Sep XXXX  
12:51:19 GMT  
Content-Type: text/html  
...
```

```
<html>  
...  
</html>
```

HttpServletResponse
<<interface>>

```
setCharacterEncoding(String charset)  
setContentLength(int len)  
setContentType(String type)  
setStatus(int status)  
setLocale(Locale locale)  
...
```



L'écriture de la réponse : focus - le code de statut

- Les serv



```
HTTP/1.1 200 OK
Date: Mon, 25 Sep XXXX
12:51:19 GMT
Content-Type: text/html
...
```

```
<html>
...
</html>
```

HttpServletResponse
<<interface>>

```
setCharacterEncoding(String charset)
setContentLength(int len)
.setContentType(String type)
setStatus(int status)
setLocale(Locale locale)
...
SC_OK (int valeur 200)
SC_NOT_FOUND (int valeur 404)
SC_...
```



L'écriture de la réponse : tous les en-têtes



```
HTTP/1.1 200 OK  
Date: Mon, 25 Sep XXXX  
12:51:19 GMT  
Content-Type: text/html  
...
```

```
<html>  
...  
</html>
```

HttpServletResponse
<<interface>>

```
setHeader(String name, String value)  
setDateHeader(String name, long date)  
setIntHeader(String name, int value)  
...
```



L'écriture de la réponse : le corps



```
HTTP/1.1 200 OK  
Date: Mon, 25 Sep XXXX  
12:51:19 GMT  
Content-Type: text/html  
...
```

```
<html>  
...  
</html>
```

HttpServletResponse <<interface>>

getWriter() : PrintWriter

Flux texte

frequent

getOutputStream() : ServletOutputStream

Flux binaire

rare



L'écriture de la réponse : le corps au format texte

- Les servlets

Utilisation

```
PrintWriter out = response.getWriter();  
out.println("Ecriture du corps de la réponse HTTP");  
//Suite écriture...  
out.close();
```

Manipulation du tampon

```
//Taille du tampon  
response.setBufferSize(size);  
//Vidage du tampon  
response.reset();  
//Forcer l'envoi du tampon  
out.flush();
```



Démo

L'écriture de la réponse



L'écriture de la réponse : la redirection permanente

- Les servlets



HttpServletResponse <<interface>>
<pre>setStatus(int status) setHeader(String name, String value) SC_MOVE_PERMANENTLY</pre>

```
response.setStatus(HttpServletResponse.SC_MOVED_PERMANENTLY);
response.setHeader("Location", "Nouvelle URL");
```

rare



L'écriture de la réponse : la redirection temporaire

- Les servlets



HttpServletResponse <<interface>>
<pre>setStatus(int status) setHeader(String name, String value) SC_MOVE_TEMPORARILY ou sendRedirect(String location)</pre>

```
response.sendRedirect("URL Temporaire");
```

frequent



L'écriture de la réponse : répondre une erreur

- Les servlets

HttpServletResponse

<<interface>>

sendError(int status)

sendError(int status, String message)

web.xml

```
<error-page>
```

```
    <error-code>500</error-code>
```

```
    <location>/erreur500.html</location>
```

```
</error-page>
```

```
response.sendError(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);
```



Démo

Rediriger la réponse



Atelier

Rechercher le nombre tiré au sort





Java avancé

Les Servlets