

# Programació

UT11.2. JSP



# Introducció

- Ja sabem el que són els Servlets i com enviar informació des d'una pàgina web per a que ens responguen amb informació HTML generada al servidor.
- Però ara sorgeix la problemàtica que ens pot interessar que la informació de resposta no siga una pàgina HTML completa sinó part d'ella o informació que la complete.
- Per això entre d'altres motius disposem de la tecnologia JSP.



# Què és JSP?

- JSP és el acrònim de Jakarta Server Pages (anteriorment Java Server Pages)
- Les pàgines JSP normalment mesclen codi HTML, CSS o JavaScript amb codi **Java que s'executa del costat del servidor web.**
- El codi Java rep, igual que els Servlets, normalment la informació des d'un formulari, des dels paràmetres de la URL del navegador o inclús des dels atributs "request" que li envia un Servlet.
  - Posteriorment s'envia al client el codi HTML+CSS+JS resultant.



# Fitxers J S P

- Codi web (HTML, CSS o JS) amb codi Java incrustat.
- Els arxius ha de tindre l'extensió `.jsp`
- Han d'estar emmagatzemats a la carpeta de continguts web del servidor web (a la mateixa ubicació que els arxius html)



# Exemple d'arxiu J S P

Codi HTML

```
<%@page import="java.time.LocalDateTime"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Hora actual</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <h2>
      Ara mateix són les <%= LocalDateTime.now() %>
    </h2>
  </body>
</html>
```

Codi Java



# Execució de l'exemple de J S P

- Creem un arxiu anomenat `hora.jsp` amb el codi d'abans i el guardem en la mateixa carpeta que `index.html`
- Si modifiquem el codi d'`index.html` per a que l'action ara cride a la pàgina jsp creada i executem l'aplicació web. Quan fem clic en el botó d'enviament obtenim això:

**Hello World!**

**Ara mateix són les 20:42:05.705508**



# Execució de l'exemple de JSP

- És important que veges que la informació rebuda, no és la pàgina jsp (encara que a la barra d'adreces així ho indique), sinó el codi html generat per el servidor.
- Si veiem el codi font...

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Hora actual</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <h2>
      Ara mateix són les 20:42:05.705508
    </h2>
  </body>
</html>
```



# Tipus de codi més usats en un JSP

**Expressions**      ➡ `<%= una expressió %>`

**Scriptlets**      ➡ `<% codi (diverses línies) %>`

**Declaracions**      ➡ `<%! declaració de variables o mètodes %>`



# Objectes predefinits en JSP

De la mateixa manera que en un Servlet, també en el codi JSP tenim a disposició una sèrie d'objectes disponibles (no estan tots):

- **request**: és el objecte request que es rep a doPost i doGet. Classe `HttpServletRequest`
- **response**: el mateix per a response. Classe `HttpServletResponse`
- **session**: proporciona una sessió única per cada usuari. Classe `HttpSession`
- **out**: escriu contingut a la pàgina web actual. Classe `PrintWriter`
- **application**: proporciona serveis a l'aplicació web. Classe `ServletContext`. Útil per guardar atributs. Els valors persisteixen entre invocacions al servlet, accés a bases de dades, peticions a altres servlets, missatges d'error i d'altres tipus



# Rebre paràmetres a JSP

- Ja vam veure com rebre paràmetres d'un formulari html amb un Servlet.
- De la mateixa manera, fent ús de “request” i el mètode “getParameter” podem obtenir els camps del formulari o els paràmetres passats per la URL

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <ul>
      <li>Nom: <%= request.getParameter("nom") %> </li>
      <li>Cognoms: <%= request.getParameter("cognoms") %> </li>
      <li>Email: <%= request.getParameter("email") %> </li>
    </ul>
  </body>
</html>
```



# Saltar a un Servlet, HTML o JSP des de codi Java

- També es pot donar el cas que des d'un Servlet o des d'un Scriplet d'un JSP vulguem fer una redirecció.
- Per això farem servir el mètode `getRequestDispatcher` de l'objecte `request`.

Ruta relativa a la  
pàgina JSP, HTML o  
nom del Servlet

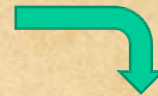
```
if(usuarioInvalido){  
    request.getRequestDispatcher("login.jsp").forward(request, response);  
}
```



# Passar informació a un Servlet/JSP des d'un Servlet/JSP

- També és habitual que necessitem enviar informació calculada o obtinguda d'un servlet a un altre (o JSP).
- Per això farem ús dels mètodes `get/setAttribute` de l'objecte `request`.

```
request.setAttribute("missatge", "Producte " + producte.getCodigo() + " afegit amb èxit");  
request.getRequestDispatcher("productes.jsp").forward(request, response);
```



```
<%  
    String avis = (String) request.getAttribute("missatge");  
    if(avis != null){  
        out.println("<div class=\"avis\">" + avis + "</div>");  
    }  
%>
```

productes.jsp