RequestDispatcher

(Octubre de 2005, modificado en Abril de 2007)

Introducción

Un servlet/JSP puede invocar de modo directo a un recurso de la web. La idea es poder reenviar la petición (request) a dicho recurso. Para hacer esto necesitamos un **RequestDispatcher**, que es un referencia que encapsula el recurso. En el siguiente ejemplo el servlet tiene como "dispatcher" otro servlet (sc es un **ServletContext** que se obtiene en init()):

```
RequestDispatcher dispatcher = sc.getRequestDispatcher("/servlet/request_imagen");
```

Observar que el argumento corresponde con el **url-pattern** de web.xml. La creación del dispatcher se podría escribir de otra forma:

```
RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/servlet/request_imagen");
```

Pero se puede observar que este método en ocasiones falla y la documentación de Sun recomienda la primera forma: sc. **getRequestDispatcher(arg)**. El argumento es la ruta del recurso, que **debe comenzar con "/"** y es interpretado como relativo al contexto actual. Conviene usar **getContext()** para obtener un RequestDispatcher de recursos en contextos externos. getRequestDispatcher(arg) retorna null si el ServletContext no puede retornar un RequestDispatcher.

La invocación al recurso se puede hacer de dos formas:

- **Incluir** el recurso en el flujo de salida. La salida de "/servlet/request_imagen" se incluye en la salida del primer servlet.

```
dispatcher.include(request, response);
```

- **Redirigir** (forward) la petición al recurso. Funcionalmente semejante a sendRedirect(). Se trata de **redirigir la petición** a otro componente:

```
dispatcher.forward(request, response);
```

  La **diferencia entre sendRedirect() y forward() (o también include())** es:
  - En sendRedirect() la petición acaba bajo el control del segundo servlet. La URL que se puede ver en el navegador es la del segundo servlet.
  - En forward() o include() la petición se controla por el primer servlet. **La URL que se puede ver en el navegador es la del primer servlet**.

El ejemplo

Ejemplo de RequestDispatcher, se usan dos métodos: include() o forward() hacia un segundo servlet (Request_Imagen). Escoja uno:

include Enviar

El código del ejemplo se puede ver a continuación:

```
public class Request_Dispatcher extends HttpServlet {

        private ServletContext sc;                // Contexto del servlet

        /**************************************************************************
         * INIT: se obtiene el contexto
         **************************************************************************/
        public void init(ServletConfig config) throws ServletException {
                sc = config.getServletContext(); // Obtengo contexto del servlet
        }

        /**************************************************************************
         * POST
         **************************************************************************/
        public void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                response.setContentType( "text/html" );
                PrintWriter out = response.getWriter();

                //// Inicio de página
                out.println("<html>");
                out.println("<head><title>Ejemplo de RequestDispatcher</title></head>");
                out.println("<body bgcolor=\"#FFFF9D\"><FONT color=\"#000080\" FACE=\"Arial,
Helvetica,Times\" SIZE=2>");
                out.println("<CENTER><H3>Ejemplo de RequestDispatcher</H3></CENTER><HR>");
                out.println("<p>Este servlet (el primero) usará un RequestDispatcher (otro
servlet)</p>");

                /********
                 * En vez de usar getServletContext().getRequestDispatcher() aplico sc.
getRequestDispatcher(),
                 * donde sc se obtiene en init().
                 * La razón es que el primer método en ocasiones falla, de hecho, la
documentación de SUN
                 * recomienda el segundo método.
                 *********/
                RequestDispatcher dispatcher = sc.getRequestDispatcher("/servlet/
request_imagen");
                if (dispatcher != null) {
                        //// En función del param usa include() o forward()
                        if ( request.getParameter("metodo").equals("include"))
                                dispatcher.include(request, response);
                        else
                                dispatcher.forward(request, response);
                }
                else
                        out.println("<p>No se ha encontrado RequestDispatcher</p>");

                out.println("<p>Final del primer servlet</p>");
                out.println("</font></body></html>");
        }

        /**************************************************************************
         * GET: reenvia a doPost
         **************************************************************************/
        public void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                doPost(request, response);
```

```
        }

}
```

El dispatcher no tiene ningún misterio:

```
public class Request_Imagen extends HttpServlet {
        public void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                response.setContentType( "text/html" );
                PrintWriter out = response.getWriter();
                out.println("<p>Inicio del dispatcher</p>");
                out.println( "<p><img border='2' src='" + request.getContextPath() +"/java/
servlets/imagenes/libro.jpg'></p>");
                out.println("<p>Fin de dispatcher. Esta imagen ha sido mostrada por el
dispatcher</p>");
        }
}
```

Volver al índice