Métodos y Técnicas de Programación Avanzada

Memoria del Trabajo Final

Gonzalo Senovilla Minguela, Miguel Vítores Vicente
UEMC-2º Ingeniería Informática
Métodos y Técnicas de Programación Avanzada

Índice

Descripciones de Clases

Package interfaz

Principal.java

Es donde iniciamos el proceso principal del programa instanciando en el main un objeto de tipo Ventana.

Ventana.java

Se trata de una clase que hereda de la ventana de java swing, JFrame. En este contenedor se almacenan un JPanel que incluye dos botones, uno para arrancar y otro para parar el servidor, y un JScrollPane que incluye un objeto de tipo PanelDirectorios donde se muestran los alias del servidor web.

Tiene un menú Archivo con las opciones de importar un nuevo directorio, eliminar un directorio del servidor, renombrarlo, editar la configuración de uno de los directorios y finalmente salir de la aplicación de control del servidor.

- *importarDir*: Copia el directorio seleccionado del pc del administrado a la carpeta de ficheros del servidor y pide una configuración inicial para la carpeta, así como elegir un alias.
- *eliminarDir*: Elimina el directorio seleccionado de la carpeta de ficheros del servidor.
- *renameDir*: Renombra la carpeta seleccionada, así como su fichero “alias”.conf.
- *editarConf*: Seleccionamos una carpeta y nos pide su nueva configuración.

PanelDirectorios.java

Hereda de JTextPane y sirve para añadir los directorios del servidor a la interfaz de usuario como texto para que pueda visualizarlos el administrador al principio de la ejecución, así como actualizar ese texto si se añade, elimina o renombra algún alias desde la IU.

Para actualizar usamos *super.revalidate()* seguido de *super.repaint(),* lo cual nos daba problemas anteriormente con un panel de radioButtons que parecía una mejor opción que esta, pero nos decantamos por JTextPane porque se actualizaba correctamente si se modificaba algún alias.

Package practicafinalmtpa

Constants.java

Incluye una serie de constantes de rutas de carpetas como ficheros, la papelera para probar el delete, los mimes o ips baneadas del servidor.

HTTPProcesamiento.java

Es una interfaz con métodos que se implementan en AdministradorPeticion para el procesamiento de peticiones dependiendo de los verbos y para la lectura de estas.

MiServidor.java

Se ejecuta un hilo de MiServidor cuando se arranca el servidor desde la IU. Tiene un ThreadPoolExecutor con un número máximo de trabajadores configurable.

Primero se cargan las configuraciones de cada directorio del servidor y después se esperan conexiones y se comprueba que la ip que intenta conectarse no se encuentra en el fichero de ips baneadas. Si no es así se atiende la petición y se le manda un objeto AdministradorPeticion a un hilo trabajador para que se procese dicha petición.

AdministradorPeticion.java

Esta clase es la encargada del procesado de las peticiones que llegan al servidor, implementa Runnable ya que va a ser utilizada mediante multiprocesamiento.

Las funciones principales de esta clase son las de leer la petición, procesarla, generar la respuesta y enviar dicha respuesta. Además, es la encargada de mantener cargada la configuración de la aplicación web que se esté atendiendo en ese momento.

Mediante el método procesarPeticion la clase identifica el verbo de la petición y dependiendo de este se ejecuta la función correspondiente al verbo (procesarVERBO()).

- *procesarGet*: Lee el recurso de la petición, lo busca en el directorio de aplicación, si lo encuentra, lee los bytes del fichero y genera una respuesta con código 200 “OK” y el fichero pedido por el cliente; si no lo encuentra genera una respuesta 404 “Not Found”. Finalmente se envía la respuesta al cliente.
- *procesarDelete*: Lee el recurso que la petición quiere que se borre y lo intenta borrar mediante la función borrarFichero, si la esta función devuelve true se genera una respuesta con código 202 “Accepted” si por el contrario no se ha podido borrar el archivo se crea un 404 “Not Found”y por último se envía la respuesta.
- *procesarPut*: Lee el recurso, comprueba si existe en el directorio web, si existe se ejecuta crearFichero y dependiendo del retorno, si true nueva respuesta 200 “OK”, si no, 501 “Not Implemented”. Si no existiese en el directorio web dependiendo del retorno de crearFichero se generaría un 201 “Created” o un 400 “Bad Request” y por último en cualquier caso se envía la respuesta.
- *procesarPost*: Recibe el formulario de la petición codificado en x-www-form-urlencoded, lo procesa y lo almacena en la base de datos del servidor. Devuelve una respuesta 301 “Moved Permanently”

PeticionHTTP.java

Instanciamos PeticionHTTP con un conjunto de bytes que es el mensaje que recibimos del navegador y nos quedamos con aspectos que nos interesan de la petición, como el verbo, la cabecera, el cuerpo si hay, etc. Así con esta información se podrá responder a la petición debidamente. El método que filtra el mensaje es *requestBreakdown.*

RespuestaHTTP.java

RespuestaHTTP genera una respuesta acorde a una petición. A partir de esta petición puede devolver un tipo de contenido si hay que devolver algo en el cuerpo y también puede devolver una respuesta al cliente en formato comprimido gzip si en el fichero de configuraciones de ese directorio está así estipulado.

WebAppConfig.java

Esta clase es utilizada como un registro para las configuraciones de las distintas páginas webs, de esta forma se mejora la reutilización de código en el futuro ya que se pueden añadir configuraciones sin alterar las firmas de las funciones ya existentes.

A fecha de hoy cuenta con los atributos:

- Alias al que pertenece dicha configuración.
- Índex del directorio web asociado a este alias.
- Página de error especificada para el directorio web asociado a este alias.
- Boolean gzip, por si la aplicación web quiere la respuesta comprimida.
- Boolean dirStruct que se encarga de mostrar el árbol de directorios de una aplicación en caso de no existir un index.

LogPeticionUsuario.java

Se encarga de registrar cuándo, qué página del servidor y quién ha solicitado una página del servidor. Esta información se almacena en la carpeta log.

Diagrama de Clases