

Informe de práctica

Aplicaciones Distribuidas

Entrega Práctica 3

Servicio Web Rest

Alumno: Max Pasten

Alumno: Marcel Calleja

14 de noviembre de 2023

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Contenido

Introducción	1
Modificaciones y funcionalidades del servicio Web Rest	1
Bibliografía consultada	4

Introducción

Este informe contiene la documentación sobre las decisiones de diseño, operaciones y modificaciones de nuestro servicio Web Rest de almacenamiento de imágenes además de las referencias consultadas.

En este informe se describirán las modificaciones realizadas en la aplicación de cliente y en la aplicación del servicio web Rest, principalmente las operaciones extra y los distintos cambios respecto al esquema original de la aplicación del servicio web Rest.

Modificaciones y funcionalidades del Servicio Web Rest

En este apartado detallaremos cada una de las operaciones realizadas, además de las funcionalidades que se agregaron y que se modificaron.

Funcionalidad de convertir lista de objetos a JSON

Para poder enviar los datos de petición del cliente, la aplicación de servicio rest obtiene una lista de objetos de la clase Image (que contiene los atributos de las imágenes) a través de una conexión con la clase que accede a la base de datos, esta se convierte en un formato JSON y se envía al cliente. Del mismo modo, el cliente recupera el JSON enviado por el servicio Rest y lo convierte en una lista de objetos Image.

Operación de búsqueda combinada

Se agregó la operación de búsqueda combinada en la aplicación del servicio web Rest usando el siguiente endpoint:

<http://localhost:8080/RestAD/resources/jakartaee9/searchCombined>

Esta operación de búsqueda combinada necesita los parámetros obligatorios de fecha, enviando una fecha de inicio y una fecha final. Opcionalmente se pueden agregar más parámetros de búsqueda como el autor de la imagen o las palabras clave que posean.

La operación se realizó con un método POST enviando los parámetros de fecha de inicio, fecha final, autor y palabras clave con "formParam".

Operación de imágenes recientes

Se agregó la operación de imágenes recientes en la aplicación del servicio web Rest usando el siguiente endpoint:

<http://localhost:8080/RestAD/resources/jakartaee9/imageRecent>

Esta operación de imágenes recientes se usa para mostrar en el menú de la aplicación del cliente las 5 imágenes más recientes ingresadas en el sistema.

La operaci3n se realiz3 con un m3todo GET y no requiere un par3metro por el cliente.

P3gina de confirmaci3n para eliminar una imagen

Se ha a1adido una p3gina jsp de confirmaci3n para eliminar la imagen que se muestra al pulsar el bot3n de eliminar. El jsp consiste en un mensaje que pregunta si se quiere eliminar la imagen, y dos botones "S3" y "No", en caso de pulsar S3 se redirigir3 a la p3gina de la lista de im3genes o la p3gina de error, en el caso de que haya un error o no, respectivamente. En caso de pulsar No, se redirigir3 de nuevo a la p3gina de listar im3genes.

Bibliograf3a consultada

Con el objetivo de saber c3mo utilizar json en Java, para la comunicaci3n entre las dos aplicaciones:

<https://stackoverflow.com/questions/20117148/how-to-create-json-object-using-string>

Con el objetivo de saber c3mo leer el body de las responses de la aplicaci3n rest:

<https://stackoverflow.com/questions/9620683/java-fileoutputstream-create-file-if-not-exists>

Chat GTP:

Se us3 como referencia y gu3a para la conexi3n el siguiente c3digo proporcionado para el env3o de datos al servicio web rest

Java

```
private void sendToServer(String data, HttpServletResponse response) throws
IOException {
    try {
        URL url = new URL("http://localhost:8080/Server/ServerServlet");
        HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
        connection.setRequestMethod("POST");
        connection.setDoOutput(true);
        PrintWriter out = new PrintWriter(connection.getOutputStream());
        out.write("dato=" + URLEncoder.encode(data, "UTF-8"));
        out.close();
        int responseCode = connection.getResponseCode();
        if (responseCode == HttpURLConnection.HTTP_OK) {
            response.getWriter().write("Datos enviados correctamente al
servidor.");
        } else {
            response.getWriter().write("Error al enviar datos al servidor.
C3digo de respuesta: " + responseCode);
        }
    }
}
```

```
    }  
  } catch (Exception e) {  
    e.printStackTrace();  
    response.getWriter().write("Error: " + e.getMessage());  
  }  
}
```

En este ejemplo simple que se proporcionó se envía un dato usando el método post a un servidor y como respuesta del servidor un 200 OK para validar que se realizó la conexión. En la práctica se modificó este código de referencia para poder movernos entre los jsp del lado del cliente y realizar las conversiones de los Json que enviaba el servidor.