

Informe de prácticas

Entrega 03

Alumno/a: Jordi Solé García

Alumno/a: Victor Gallego Izquierdo

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Contenido

Introducción.....	1
Decisiones de diseño.....	1
Repositorios de código consultados.....	3
Bibliografía consultada.....	3

Introducción

En este informe se va a detallar la nueva implementación de la práctica 2 la cual consistía en una aplicación web. Ahora, la práctica 3, consiste en dos aplicaciones: la cliente y la del servicio REST. Esta última es la que se encargará de acceder a la base de datos y, por lo tanto, servirá a la cliente. Las únicas funciones que se seguirán teniendo en cuenta en la aplicación cliente son las que hacen referencia a los controles de sesión y al almacenamiento de las imágenes en un directorio fijo de la aplicación.

Antes de comenzar a explicar las decisiones de diseño, contestaremos las tres preguntas que se plantean en el enunciado:

a) ¿Por qué podemos acceder al servicio desde el navegador?

Al tratarse de una operación con el método GET, si la aplicación que contiene este servicio está activa, la petición al servicio se realiza mediante la URL. Por este motivo, se puede acceder desde el navegador.

b) ¿Qué método HTTP se está utilizando?

Se está utilizando el método GET

c) ¿Con qué tipo MIME? Comprueba la características de red en las opciones de desarrolladores de los navegadores (F12 en sistemas Windows y Linux).

text/html

Decisiones de diseño

Aplicación REST

Hemos decidido añadir varios packages de la aplicación web de la anterior práctica, los cuales ya explicamos en detalle en el anterior informe. Estos son:

-Package de Application: que contiene las clases de Image y User

-Package de Drivers: contienen todos los controladores explicados y necesarios de la práctica 2. Ahora es necesario que los contenga la aplicación REST ya que es esta la que se ocupa del acceso a las BBDD.

Después en el apartado de **Restful Web Services → HTTP methods**, es donde hemos implementado todos los servicios:

- Los servicios que están en el enunciado de la práctica:

- Login()
- registerImage()

- deleteImage()
- modifyImage()
- listImage()
- searchByID()
- searchByTitle()
- searchByAuthor()
- searchByCreationDate()
- searchByKeywords()

En los servicios modify y register hemos creído conveniente añadir el campo filename para una implementación más fácil acorde a como lo teníamos en la práctica anterior.

-Servicios extra

Hemos implementado dos servicios extra: registerUser y combinedSearch.

RegisterUser: Este servicio nos ofrece la posibilidad de añadir un nuevo usuario en nuestra BBDD. En resumen, realiza la función del servlet “registrarUsuario” de la práctica 2. Se realiza mediante el método POST y espera como parámetros el usuario, la contraseña y la confirmación de contraseña. En caso correcto devuelve un código de estado HTTP de tipo 200 y, si hay algún error, uno de tipo 400.

CombinedSearch: Este servicio nos ofrece la búsqueda combinada que nos ofrecía la práctica2. En este caso, como había que implementar y utilizar las operaciones de búsqueda independientes, solo se utiliza la “combinedSearch” en caso de que haya más de un campo a filtrar(sin tener en cuenta el ID”. Esto se debe a que no tiene sentido combinar ninguna búsqueda con el ID ya que este campo o coincide o no, no puede haber ningún patrón parecido. Por lo tanto, los campos posibles a combinar son: el título, la descripción, las palabras clave, el autor y la fecha de creación. Cabe recalcar que esta búsqueda se filtra haciendo una operación AND, de forma que si un campo de búsqueda ya no es satisfactorio la búsqueda será nula.

Este servicio se lleva a cabo con el método POST, ya que pueden haber campos de filtrado vacíos y con GET no se podría tratar. El resultado es una lista de imágenes vacía o no y en formato JSON.

Aplicación WEB

En esta parte de la práctica no hay mucho que explicar, ya que sigue siendo lo mismo que la anterior, simplemente que las operaciones que se llevaban a cabo en el servlet ahora se realizan haciendo una petición al servicio REST. Aún así, la aplicación web sigue teniendo el trabajo del control de sesiones y de la gestión de las imágenes en disco. A causa de esta gestión, tenemos que seguir manteniendo el package de Application ya que tiene que

acceder a la clase Image y el package de “Drivers”, ya que para eliminar alguna imagen del disco tenemos que acceder mediante su ID.

Además, hemos añadido un campo de búsqueda en el “buscarImagen.jsp” debido a que el enunciado de la práctica pedía una búsqueda mediante el ID.

Repositorios de código consultados

Lista de ejemplos de código consultados en github, gitlab, etc.

Bibliografía consultada

Códigos de estado de respuesta HTTP - HTTP | MDN. (2022, 7 noviembre).

<https://developer.mozilla.org/es/docs/Web/HTTP/Status>

Rodríguez, M. A. (2012, 17 septiembre). *Jugando con JSON en Java y la librería Gson.*

Adictos al trabajo. <https://www.adictosaltrabajo.com/2012/09/17/gson-java-json/>