
Práctica de DBCS

Curso 2021/22

Laboratorio Práctica 1: Servicios REST

Introducción

Esta primera práctica está centrada en los conceptos vistos en el laboratorio hasta ahora. Vamos a construir un Servicio Web de tipo REST, y un cliente web para él.

Descripción

Para esta primera práctica se necesita construir una aplicación web con las siguientes características:

- Tener una API REST con Spring
- Tener una aplicación web desarrollada con Angular que conecte con la API REST
- Tener como persistencia una base de datos MySQL/MariaDB

Esta aplicación permitirá realizar las acciones CRUD para la gestión básica de usuarios. Teniendo esto en cuenta, la API debe cumplir la siguiente interfaz:

- GET /users: Devolverá una lista de todos los usuarios existentes en la base de datos.
- GET /users/{id}: Devolverá el usuario con el id especificado.
- POST /users: Creará un nuevo usuario en la base de datos.
- PUT /users/{id}: Modificará el usuario con el id especificado.
- DELETE /users/{id}: Borrará el usuario con el id especificado.
- GET /users?enable=false: Devolverá la lista de los usuarios inhabilitados.
- PUT /users/enable?user_id=Lista_Ids_Usuarios_A_Activar. Activará (modificará el campo “enabled” de la tabla) de aquellos usuarios cuyo “id” esté incluido en la lista

El modelo del usuario tendrá los siguientes atributos:

- id: Identificador del usuario.
- name: Apodo del usuario.
- firstName: Nombre del usuario.
- lastName: Apellido del usuario.
- email: Correo electrónico del usuario.
- password: Contraseña del usuario.
- enabled: Estado del usuario.
- createdAt: Fecha de creación del usuario.
- updatedAt: Fecha de modificación del usuario.

Desarrollo de la práctica

Se deberán realizar las acciones necesarias para crear la API y la aplicación web funcionales con el modelo e interfaces plateadas. Se deberá, además, utilizar y guardar los datos en la base de datos MySQL/MariaDB conectada. La aplicación web cliente conectará e interactuará con la API para poder realizar su función. Para esto, se debe hacer uso de lo aprendido durante los tutoriales de las sesiones anteriores de laboratorio.

Por la parte del cliente web (front), la aplicación Angular deberá ser funcional. Contará con una página principal que mostrará una lista de usuarios, una página para crear nuevos usuarios y una página para editar los usuarios existentes. También se valorará la usabilidad y diseño. Se recomienda utilizar alguna librería como Bootstrap, Material, etc. para la maquetación. Se puede revisar la documentación de Bootstrap sobre componentes y layout en la siguiente [página](#). La aplicación deberá tener los modelos, servicios, componentes, módulos y configuración de rutas necesarios.

El código debe estar correctamente estructurado en proyectos, carpetas, etc., tal y como se ha indicado en los tutoriales.

Se debe realizar el tratamiento de excepciones.

Es necesario pensar bien la estructura, el control de versiones, etc., puesto que esta práctica es la base sobre las que se trabajará en las siguientes.

Todas las funcionalidades añadidas que aporten seguridad en la información y buenas prácticas serán valoradas positivamente.

Evaluación

Se evaluará:

- Código creado: valorando por parte del profesor el material entregado
- Funcionamiento: mediante defensa oral del proyecto durante las siguientes sesiones de laboratorio.

Se valorará:

- Claridad en la organización de la aplicación.
- Comentarios explicativos en el código.
- Funcionalidades añadidas.

Entrega

Se subirá todo el código en el Gitlab de la escuela. Se dará acceso al repositorio (rol de *maintanier*) sobre el que se esté trabajando al profesor.

Documentación a entregar a través del campus virtual:

- Descripción de cómo poner en marcha la aplicación.
- Diagrama de componentes de la aplicación donde se vea la estructura de componentes y de Angular.

Límite de entrega del código y la documentación: ver documento de planificación.