Sistemas Distribuidos e Internet
Desarrollo web con Node.js

Ejercicios Complementarios
Curso 2023/2024

Contenido

1 Introducción

Una de las condiciones que el alumno ***tiene que cumplir*** para no ir a la prueba de autoría es realizar al menos el ***50% de los ejercicios complementarios*** propuestos. La realización del porcentaje mínimo exigido se realizará en el en horario de clase, pudiendo realizar el resto de los ejercicios en casa.

Para cada uno de los ejercicios realizado el alumno deberá realizar un commit en GitHub donde se indique la sesión y el número del ejercicio correspondiente. El mensaje del commit correspondiente se realizará siguiendo la siguiente nomenclatura:

Commit Message -> "SDI–sesiónX-Ejercicio-complementarioY"

Donde:

- X – indica el número de la sesión
- Y – indica el número del ejercicio complementario

Ejemplo:

Commit Message -> "SDI–IDGIT-sesión10-ejercicio-complementario1"

Este documento es copyright (C) 2024 Edward Rolando Núñez Valdez, Jordán Pascual y Enrique De la Cal

Está prohibida su distribución sin el consentimiento expreso y por escrito de los autores.

2 Sesión 10 – Desarrollo ágil con Node.js

Ejercicio 1 (50%). En este ejercicio se refinará el proceso de compra, añadiendo una serie de comprobaciones que mejorará la experiencia del usuario. Los objetivos son:

- Evitar que un usuario pueda volver a comprar una canción.
- Los autores no pueden comprar sus canciones.

Requisitos:

- En el caso que el usuario sea autor o haya comprado la canción, en la vista **song.twig** no se mostrará el enlace para comprar la canción. En su lugar, se mostrará la etiqueta <audio> para que puedan reproducir la canción (hay un ejemplo de uso de esta etiqueta en el purchase.twig).
- Si el usuario conectado no ha comprado la canción o no es el autor, debe aparecer el enlace de comprar.
- Para lograr lo anterior además de modificar la vista correspondiente hay que modificar el método GET **"/songs/:id"**
- **Validación en el servidor**: Solamente con el cambio anterior, sería posible comprar una canción por cualquier usuario introduciendo la URL de compra en el navegador. Por tanto, deben añadirse las comprobaciones necesarias en POST **'/songs/buy/:id'** para que se cumplan los dos objetivos planteados en el ejercicio.

Nota: Subir el código a GitHub en este punto. Commit Message → "SDI–IDGIT-sesión10-ejercicio-complementario1"

Ejercicio 2 (10%). Realizar los cambios necesarios para mostrar una página de error personalizada.

Requisitos:

- Modificar la vista **error.twig** en la carpeta views, si es necesario.
- El mensaje de error o notificación debe proporcionarse a la vista como parámetro.
- En lugar de mostrar el error o la notificación en la página actual se deberá redirigir a la página de error creada.

Nota: Subir el código a GitHub en este punto. Commit Message → "SDI–IDGIT-sesión10-ejercicio-complementario2"

Ejercicio 3 (30%). Actualmente se están pasando los mensajes de error o de información a la vista mediante parámetros en la URL (por ejemplo: login? message =Email o password incorrecto" messageType=alert-danger). En el presente ejercicio se plantea realizar los

Este documento es copyright (C) 2024 Edward Rolando Núñez Valdez, Jordán Pascual y Enrique De la Cal

Está prohibida su distribución sin el consentimiento expreso y por escrito de los autores.

cambios necesarios para que el mensaje de error y tipo de error viajen en sesión dentro de un objeto "errores", cuyo contenido será:

- message: contendrá el mensaje de error.
- messageType: si es de tipo alert-danger o alert-info.

Requisitos

- En el método que detectemos el error, añadiremos a la sesión el mensaje de **error** y el **tipo** de mensaje.
- El error debe mostrarse únicamente en la primera vista que se envíe al usuario. Es decir, en todos los métodos del controlador habrá que **pasar las variables de error en sesión (si existen) a variables (que enviaremos a la vista), dejando la sesión limpia de variables de error**. Esto evitará que se propague el mismo error en diferentes peticiones.

Nota: Subir el codigo a GitHub en este punto. Commit Message → "SDI–IDGIT-sesión10-ejercicio-complementario3"

Ejercicio 4 (10%). En el ejercicio anterior se plantea realizar los cambios necesarios para que el mensaje de error y tipo de error viajen en sesión dentro de un objeto "errores". En este ejercicio se propone utilizar el manejador de errores definido en el app.js para gestionar los errores de la aplicación.

Requisitos

- En el método que detectemos el error lanzamos el error con un mensaje que indique al usuario que error a ocurrido y redirija a la página de error.
- En el caso que sea mensaje de información o de alerta debería enviar el mensaje a la página que estamos redirigiendo.
- **Ejemplo de un error:** hacer una petición de consulta para mostrar una canción con un ***ID no válido***. http://localhost:8081/songs/5e78a81ed290591b64ae1aae222
- **Ejemplo de un mensaje de alerta:** Intentar comprar una canción en la que el usuario autenticado es el autor.
- **Ejemplo de un mensaje de información:** Registrarse como usuario e indicar al usuario que se ha registrado con éxito.

Nota: Subir el codigo a GitHub en este punto. Commit Message → "SDI–IDGIT-sesión10-ejercicio-complementario4"

Este documento es copyright (C) 2024 Edward Rolando Núñez Valdez, Jordán Pascual y Enrique De la Cal

Está prohibida su distribución sin el consentimiento expreso y por escrito de los autores.