Universidad Carlos III

Tecnologías informáticas para la web

Curso 2021-22

Práctica 2 - Documentación

FECHA-ENTREGA: **12/12/2021**

GRUPO: **80**

Eduardo De Andrés Tabernero - 100363553

ÍNDICE

INFORMACIÓN RELEVANTE

DESAJUSTE CON EL UPLOAD DE FOTOGRAFÍAS

Cuando se realiza alguna de subida de archivo, este se guarda en el servicio de carpetas interno pero el servidor de payara (montado con eclipse) hasta que no actualizas la carpeta dentro del propio IDE no se muestra correctamente la imagen en la web. Es por ello que a la hora de actualizar, subir productos, etc. Tienes que realizar esta acción si quieres una mejor experiencia.

EJECUCIÓN DEL SCRIPT SQL

La base de datos es creada automáticamente con un solo Script. Este se encarga tanto de generar el esquema y las tablas, como de rellenar con datos las tablas necesarias (en este caso la de usuarios/clientes y la de productos) para una mejor comprensión de cómo sería visualmente del programa.

VENTA DE PRODUCTOS

A la hora de vender un producto no se ha planteado guardar un estado “vendido” dentro de la tabla productos. Para ello tenemos la tabla ventas que se encarga de guardar las transacciones. En el caso de que el vendedor realice la venta de manera externa lo puede marcar como vendido desde la actualización del producto y en ese caso sí que se marca como vendido. Así no llenamos tanto la tabla de productos disponibles y en el caso de que haya que hacer limpieza se hace mediante selección del estado.

URL DISPATCHER

Todas las url están acotadas y pensadas para que no puedan ser utilizadas incorrectamente. Es por ello que no se muestra ninguna información relevante en ellas excepto en las opciones de ‘view’ y ‘buy’ que se muestra únicamente el id del objeto con el que queremos trabajar. Para un mayor entendimiento de cómo funcionan se adjunta un pequeño esquema con el nombre Anexo1*.

FUNCIONES Y SALIDAS DEL PROGRAMA

CLIE80I

GET ALL PRODUCTS

GET	**/products**	
Description	Devuelve todos los productos que hay en la base de datos. Contenido adicional: @RequestParam	
	200	Se devuelve el contenido (lista de productos)
	---	Se devuelve contenido vacío

SEARCH BY CATEGORY

GET	**/products/category/{category}**	
Description	Devuelve una lista de productos contenidos en una categoría. Contenido adicional: @PathVariable	
	200	Se devuelve el contenido (lista de productos)
	---	Se devuelve contenido vacío

SEARCH BY ID

GET	**/products/id/{product_id}**	
Description	Devuelve el producto que coincida con el id dado. Contenido adicional: @PathVariable	
	200	Se devuelve el contenido
	---	Se devuelve un objeto vacío (null)

SEARCH BY OWNER

GET	**/products/owner/{owner}**	
Description	Devuelve una lista de productos que pertenecen a un cliente. Contenido adicional: @PathVariabl@Validated	
	200	Se devuelve el contenido (lista de productos)
	---	Se devuelve contenido vacío

UPDATE A PRODUCT

PUT	**/update/{product_id}**	
Description	Modifica el objeto con el id que se indica y se cambia su información dentro de la base de datos. Se devuelve el 'status' no un objeto. Contenido adicional: @PathVariable @Validated @RequestBody	
	200	La actualización se ha realizado correctamente
	500	No se ha podido actualizar el producto

DELETE A PRODUCT

DELETE	**/delete/{product_id}**	
Description	Elimina el objeto con el id especificado y devuelve el 'status'. Contenido adicional: @PathVariable @Validated	
	200	La eliminación se ha realizado correctamente
	500	No se ha podido eliminar el producto

LOGIN

POST	**/Products**
Description	Introduce un nuevo elemento en la base de datos que viene

	dado por un objeto. Contenido adicional: @RequestBody	
	200	La actualización se ha realizado correctamente
	500	No se ha podido actualizar el producto

BANK 80I

GET ALL TRANSACTIONS

GET	**/transactions**	
Description	Devuelve todas las transacciones que hay en la base de datos. Contenido adicional: @RequestParam	
	200	Se devuelve el contenido (lista de transacciones)
	---	Se devuelve contenido vacío

DELETE TRANSACTION

DELETE	**/transactions**	
Description	Elimina una transacción con el id dado de la base de datos. Contenido adicional: @PathVariable @Validated	
	201	El objeto se ha eliminado correctamente
	500	No se ha podido eliminar el objeto

GET TRANSACTION BY ID

GET	**/transactions/id/{transaction_id}**
Description	Devuelve todas las transacciones que hay en la base de datos. Contenido adicional: @PathVariable

	200	Se devuelve el contenido (objeto transacción)
	---	Se devuelve contenido vacío

SAVE TRANSACTION

POST	**/transactions**	
Description	Guarda una nueva transacción en la base de datos. Contenido adicional: @PathVariable	
	200	Se ha guardado correctamente
	500	No se ha podido guardar

CLIE80I

GET ALL USERS

GET	**/users**	
Description	Devuelve todos los usuarios que hay en la base de datos. Contenido adicional: @RequestParam	
	200	Se devuelve el contenido (lista de transacciones)
	---	Se devuelve contenido vacío

GET USER BY NAME

GET	**/users/{name}**	
Description	Devuelve los usuarios que compartan el mismo nombre. Contenido adicional: @PathVariable	
	200	Se devuelve el contenido (lista de transacciones)
	---	Se devuelve contenido vacío

LOGIN BY EMAIL

GET	**/login/email**	
Description	Devuelve el usuario que sea propietario del email indicado. Contenido adicional: @PathVariable	
	200	Se devuelve el usuario
	---	Se devuelve un objeto null

GET USER BY NAME AND SURNAME

GET	**/users/{name}/{surname}**	
Description	Devuelve todos los usuarios que hay en la base de datos que compartan el nombre y apellido indicado. Contenido adicional: @PathVariable	
	200	Se devuelve el contenido (lista de transacciones)
	---	Se devuelve contenido vacío

REGISTER USER

POST	**/register**	
Description	Guarda un usuario en la base de datos. Contenido adicional: @RequestBody	
	201	Se devuelve el status usuario guardado correctamente.
	500	Se devuelve el status error en el guardado de usuario.

UPDATE USER

PUT	**/update/{email}**

Description	Actualiza un usuario en la base de datos. Contenido adicional: @PathVariable @Validated	
	201	Se devuelve el status usuario modificado correctamente.
	500	Se devuelve el status error en la modificación de usuario.

DELETE USER

DELETE	**/delete/{email}**	
Description	Elimina un usuario en la base de datos. Contenido adicional: @PathVariable @Validated	
	201	Se devuelve el status usuario borrado correctamente.
	500	Se devuelve el status error en la eliminación de usuario.