

Examen Segunda Evaluación

Introducción

Este proyecto tiene como objetivo crear dos APIs REST, una para registrar errores en Redis y otra para almacenar y recuperar datos de MongoDB. La aplicación Angular será la encargada de acceder a las APIs y mostrar los datos. La aplicación tendrá un sistema de login usando JWT y dos tipos de clientes, usuarios normales y admins. Los administradores tendrán acceso a todos los productos y la posibilidad de eliminarlos, mientras que los usuarios normales sólo podrán ver los productos.

Redis Logging API

La API de Redis Logging almacenará todos los logs y errores de la aplicación Angular en Redis. Es necesario que se envíen estos errores al API. Tendrá los siguientes endpoints:

Endpoint: /log

Method: POST

Request Body:

```
{  
  "message": "string",  
  "level": "string"  
}
```

Endpoint: /log

Method: GET

Response Body:

```
{  
  "message": "string",  
  "level": "string",  
  "time": "date",  
}
```

Endpoint: /log/date

Method: GET

Parameters: date

Response Body:

```
{  
  "message": "string",  
  "level": "string",  
  "time": "date",  
}
```

obtener todos los logs a partir de esa fecha.

MongoDB API

La API de MongoDB almacenará y recuperará todos los datos necesarios para la aplicación Angular. Tendrá los siguientes endpoints:

Endpoint: /products

Method: GET

Requiere autenticación

Response Body:

```
[  
  {  
    "id": "string",  
    "name": "string",  
    "description": "string",  
    "price": "number",  
    "user": {  
      "Name": "string",  
      "id": "string",  
      "dateCreation": "date"  
    }  
  }  
]
```

Endpoint: /products

Method: POST

Requiere autenticación

Request Body:

```
{  
  "name": "string",  
  "description": "string",  
  "price": "number"  
}
```

Response Body:

```
{  
  "id": "string",  
  "name": "string",  
  "description": "string",  
  "price": "number",  
  "user": {  
    "Name": "string",  
    "id": "string",  
    "dateCreation": "date"  
  }  
}
```

El user es el usuario que ha creado el producto.

Endpoint: /products/:id

Method: GET

Requiere autenticación

Response Body:

```
{  
  "id": "string",  
  "name": "string",  
  "description": "string",  
  "price": "number",  
  "user": {  
    "Name": "string",  
    "id": "string",
```

```
        "dateCreation": "date"
    }
}
```

Endpoint: /products/:id

Method: DELETE

Response Body:

```
{
  "message": "string"
}
```

También estarán los endpoints anteriormente creados de Login pero con una modificación, en BBDD se almacenará y se mostrará otro atributo en esos objetos, el atributo isAdmin: booleano.

Endpoint: /register

Method: POST

Request Body:

```
{
  "name": "rafael",
  "password": "password",
  "admin": "boolean",
}
```

Endpoint: /login

Method: POST

Request Body:

```
{
  "name": "rafael",
  "password": "password"
}
```

Response Body:

```
{
  "token": "string",
  "admin": "boolean",
}
```

```
}
```

Angular Components

Por defecto se abrirá un Login, supondremos que ya están registrados dos usuarios en BBDD, uno cliente y otro admin.

Login Component

El componente de inicio de sesión permitirá a los usuarios iniciar sesión en la aplicación Angular utilizando su correo electrónico y contraseña. Tendrá los siguientes campos:

- Email
- Password

Este componente aparecerá por defecto, después de hacer el login se tendrá que ir a otra URL <http://localhost:4200/productos> que tendrá los siguientes componentes.

Product List Component

El Product List Component mostrará una lista de todos que ha creado ese usuario. Tendrá los siguientes campos:

- Nombre del producto
- Descripción del producto
- Precio del producto

Admin Panel Component

El componente del panel de administración mostrará una lista de todos los productos con la posibilidad de eliminar productos. Tendrá los siguientes campos:

- Nombre del producto
- Descripción del producto

- Precio del producto
- Usuario creador del producto
- Botón Eliminar

Product Form Component

Este componente permitirá crear un producto, será un formulario que almacena la información usando el API de MongoDB, aparecerá en ambos usuarios.

Rúbrica

Si el programa da un error de ejecución o no es sintácticamente correcto entonces se le pondrá un 0 a la práctica. Se valorará el uso de código que siga SOLID/KISS/DRY y las buenas prácticas comentadas en clase.

Redis API	1
MongoDB API	2
Angular Login	1
Product List Component	1
Admin Panel Component	1,5
Product Form Component	1,5
Gestión errores Angular	1,5
SOLID/KISS/DRY	0.5
TOTAL	10

Entregable

El entregable será la URL del GitHub donde tendréis el proyecto incluyendo HTML, CSS y Javascript.