

# **Desarrollo rápido de aplicaciones.**

## **Actividad Frontend + Backend.**

Jesús Fornieles Muñoz

# Índice:

<b>2. Actividades</b>	<b>3</b>
<b>2.1. Realizar la sección Get Data from a Server del Tour of Heroes</b>	<b>3</b>
<b>2.2. Obtener la imagen de los super héroes de la api de Marvel</b>	<b>3</b>
1. Añadir un campo de imagen a la entidad Hero	3
2. Crear un servicio MarvelService para consumir la API de Marvel	3
3. Integramos la obtención de la imagen en el flujo de creación y actualización de héroes	4

## 2. Actividades

### 2.1. Realizar la sección Get Data from a Server del Tour of Heroes

Para comenzar la práctica realizamos el paso 6. Get Data from a Server del tutorial aplicando los cambios mencionados en el informe.

### 2.2. Obtener la imagen de los super héroes de la api de Marvel

Para realizar esta parte se han hecho una serie de pasos en el backend de spring:

#### 1. Añadir un campo de imagen a la entidad Hero

- Modifiqué la clase Hero para incluir un campo image.
- También actualicé el DTO y el mapper para soportar este nuevo campo.

```
private Long id;  
private String name;  
private String image;
```

#### 2. Crear un servicio MarvelService para consumir la API de Marvel

Se implementó la clase MarvelService con un método getHeroImage que:

- Construye la URL de la API de Marvel usando el nombre del héroe y los parámetros de autenticación.
- Realiza una petición HTTP a la API de Marvel.
- Procesa la respuesta JSON para extraer la URL de la imagen del héroe.
- Si no encuentra imagen, devuelve una imagen por defecto.

```

1 package com.example.demo.util;
2
3 import java.security.MessageDigest;
4 import java.security.NoSuchAlgorithmException;
5
6 public class MarvelApiUtil {
7
8     public static String generateHash(String timestamp, String privateKey, String publicKey) {
9         try {
10             String input = timestamp + privateKey + publicKey;
11             MessageDigest md = MessageDigest.getInstance("MD5");
12             byte[] messageDigest = md.digest(input.getBytes());
13
14             StringBuilder hexString = new StringBuilder();
15             for (byte b : messageDigest) {
16                 String hex = Integer.toHexString(0xff & b);
17                 if (hex.length() == 1) {
18                     hexString.append('0');
19                 }
20                 hexString.append(hex);
21             }
22             return hexString.toString();
23         } catch (NoSuchAlgorithmException e) {
24             throw new RuntimeException(message: "Error generating Marvel API hash", e);
25         }
26     }
27 }

```

### 3. Integrar la obtención de la imagen en el flujo de creación y actualización de héroes

- En el controlador, al crear o actualizar un héroe:
  - Se llama a `marvelService.getHeroImage(heroDTO.getName())` para obtener la imagen desde la API de Marvel.
  - Se asigna la URL de la imagen al campo `image` del héroe antes de guardarlo en la base de datos.

```

50 @PutMapping("/{id}")
51 public HeroDTO updateHero(@PathVariable Long id, @RequestBody HeroDTO heroDTO) {
52     Hero existingHero = heroRepository.findById(id)
53         .orElseThrow(() -> new RuntimeException(message: "Hero not found"));
54
55     existingHero.setName(heroDTO.getName());
56     existingHero.setPowers(heroMapper.toEntity(heroDTO).getPowers());
57
58     // Get hero image from Marvel API
59     String imageUrl = marvelService.getHeroImage(heroDTO.getName());
60     existingHero.setImage(imageUrl);
61
62     Hero updatedHero = heroRepository.save(existingHero);
63     return heroMapper.toDto(updatedHero);
64 }

```

Con esos pasos cuando creamos héroes dentro del front ahora estos aparecen con una foto extraída de la api, proveniente de spring:

# Tour of Heroes

## My Heroes

Hero name:

Add hero

## Messages

Clear messages

HeroService: fetched heroes  
HeroService: deleted hero id=3458  
HeroService: added hero w/ id=3502  
HeroService: fetched hero id=3502  
HeroService: fetched heroes  
HeroService: added hero w/ id=3503  
HeroService: fetched hero id=3503  
HeroService: fetched heroes  
HeroService: deleted hero id=3503  
HeroService: added hero w/ id=3504  
HeroService: fetched hero id=3504  
HeroService: fetched heroes  
HeroService: added hero w/ id=3505  
HeroService: fetched hero id=3505

# Tour of Heroes

## My Heroes

Hero name:

Add hero

3506

Thor

X

## Messages

Clear messages

HeroService: fetched heroes  
HeroService: deleted hero id=3458  
HeroService: added hero w/ id=3502  
HeroService: fetched hero id=3502  
HeroService: fetched heroes  
HeroService: added hero w/ id=3503  
HeroService: fetched hero id=3503  
HeroService: fetched heroes  
HeroService: deleted hero id=3503  
HeroService: added hero w/ id=3504  
HeroService: fetched hero id=3504

# Tour of Heroes

Dashboard | Heroes

go back

## THOR Details



ID: 3506

Hero name:

Superpowers