Swagger

Swagger is a popular open-source framework used for building APIs. It provides a suite of tools that allow developers to design, document, and interact with APIs in a standardized way

By Eslam Khder

Swagger

Add dependency

```
<dependency>
    <groupId>org.springdoc</groupId>
    <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
    <version>2.2.0</version>
</dependency>
```

By Eslam Khder

Swagger

http://localhost:9090/v3/api-docs

http://localhost:9090/swagger-ui/index.html

By Eslam Khder

Swagger

```
.requestMatchers(HttpMethod.GET, ...patterns: "/swagger-ui/**", "/v3/api-docs*/**").permitAll()
```

By Eslam Khder

Swagger

```java
@Override
protected boolean shouldNotFilter(HttpServletRequest request) throws ServletException {
    // get  the url from the request
    String path = request.getRequestURI();
    if (path.contains("login") || path.contains("create-client")
            || path.contains("/v3/api-docs") || path.contains("/swagger-ui")) {
        return true;
    }

    return false;
}
```

By Eslam Khder

Generics

generics refer to a feature that allows you to write classes, interfaces, and methods with placeholders for the types

Generics

🚀 What Are Generics?

Generics allow you to define classes, interfaces, and methods where the type of data they operate on is specified as a **parameter**. This lets you write code that can work with any data type **without losing type safety**.

By Eslam Khder

Generics

✅ Why Use Generics?

1. **Type safety:** Catch type errors at compile time.
2. **Code reusability:** Write one class or method that works with many types.
3. **Eliminates casting:** No need for manual casting.

By Eslam Khder

Generics

📦 Basic Syntax of Generics

java Copy Edit

```java
class Box<T> {
    private T value;

    public void set(T value) {
        this.value = value;
    }

    public T get() {
        return value;
    }
}
```

- `T` is a **type parameter**. You can use any letter, but common ones are:
  - `T` for Type
  - `E` for Element (used in collections)
  - `K`, `V` for Key and Value (used in maps)

By Eslam Khder

Generics

```
Box<String> stringBox = new Box<>();
stringBox.set("Hello");
String value = stringBox.get();
```

By Eslam Khder

Generics

java

Copy Edit

```java
class NumberBox<T extends Number> {
    private T num;

    public void set(T num) {
        this.num = num;
    }

    public T get() {
        return num;
    }
}
```

By Eslam Khder

Generics

Generics with Methods

java Copy Edit

```java
public class GenericMethod {
    public static <T> void printArray(T[] array) {
        for (T element : array) {
            System.out.println(element);
        }
    }
}
```

Usage:

java Copy Edit

```java
Integer[] intArray = {1, 2, 3};
String[] strArray = {"A", "B", "C"};
GenericMethod.printArray(intArray);
GenericMethod.printArray(strArray);
```

By Eslam Khder

Generics

✅ Example: A Simple Generic `Pair<K, V>` Class

java

Copy Edit

```java
public class Pair<K, V> {
    private K key;
    private V value;

    public Pair(K key, V value) {
        this.key = key;
        this.value = value;
    }

    public K getKey() {
        return key;
    }

    public V getValue() {
        return value;
    }

    public void printPair() {
        System.out.println("Key: " + key + ", Value: " + value);
    }
}
```

By Eslam Khder

Generics

java

Copy Edit

```java
public class Main {
    public static void main(String[] args) {
        Pair<String, Integer> studentAge = new Pair<>("Ali", 22);
        studentAge.printPair(); // Output: Key: Ali, Value: 22

        Pair<Integer, String> product = new Pair<>(101, "Laptop");
        product.printPair(); // Output: Key: 101, Value: Laptop
    }
}
```

By Eslam Khder

Generics

java

Copy Edit

```java
Map<String, Integer> map = new HashMap<>();
map.put("Ali", 22); // "Ali" is K (key), 22 is V (value)
```

By Eslam Khder