

sakte hain, jaise ki Apache Commons, Jackson, Kestronit, UKhttp, etc.

R spring boot actuator ky hai or iska implementation kese karte he project main

Spring Boot Actuator ek production-ready feature hai, jo Spring Boot application ke health, metrics, info, log, thread dumps, etc. ke liye endpoints provide karta hai. Isse aap apne application ke status ko track kar sakte hain aur real-time monitoring ke liye APIs ko expose kar sakte hain.

Spring Boot Actuator ko Spring Boot project mein implement karne ke liye, aapko kuch simple steps follow karne honge:

1. `spring-boot-starter-actuator` dependency ko add karna: Aapko apne project ke build file mein, jaise ki pom.xml (Maven) ya build.gradle (Gradle) mein, `spring-boot-starter-actuator` dependency ko add karna hoga.
2. Endpoints ko configure karna: Aap apne application ke liye Actuator endpoints ko configure kar sakte hain. Iske liye, aapko `application.properties` file mein `management.endpoints.web.exposure.include` property ko set karna hoga.
3. Actuator ke endpoints ko access karna: Actuator ke endpoints ko access karne ke liye, aapko `http://localhost:8080/actuator` URL ko use karna hoga. Isse aap Actuator ke sabhi endpoints ki list dekh sakte hain.

Iske alawa, aap Actuator endpoints ko customise kar sakte hain aur apne application ke liye custom endpoints bhi create kar sakte hain. Actuator endpoints ko secure karne ke liye, aap Spring Security ka use kar sakte hain.

R ky aap iska example de sakte hai

Haan, main aapko ek example de sakta hoon.

Agar aap Maven ka use kar rahe hain, to aapko `pom.xml` file mein `spring-boot-starter-actuator` dependency ko add karna hoga:

```
xml
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```

Agar aap Gradle ka use kar rahe hain, to aapko `build.gradle` file mein `spring-boot-starter-actuator` dependency ko add karna hoga:

```
python
dependencies {
  implementation 'org.springframework.boot:spring-boot-starter-actuator'
}
```

Ab aapko Actuator endpoints ko configure karna hoga. Iske liye, aapko `application.properties` file mein `management.endpoints.web.exposure.include` property ko set karna hoga:

```
properties
management.endpoints.web.exposure.include=*
```

Isse Actuator ke sabhi endpoints enable ho jayenge.

Ab aap Actuator ke endpoints ko access kar sakte hain. Aap `http://localhost:8080/actuator` URL ko use karke Actuator ke sabhi endpoints ki list dekh sakte hain.

Jaise ki agar aap health check endpoint ko access karna chahte hain, to aap `http://localhost:8080/actuator/health` URL ko use kar sakte hain.

Iske alawa, aap Actuator endpoints ko customise kar sakte hain. Jaise ki agar aap custom health check endpoint banana chahte hain, to aapko `HealthIndicator` interface ko implement karna hoga:

```
java
@Component
public class CustomHealthCheck implements HealthIndicator {

  @Override
  public Health health() {
    // Custom health check logic
    return Health.up().build();
  }
}
```

Isse aap custom health check endpoint ko Actuator ke endpoints ki list mein add kar sakte hain.

Regenerate response

+ New chat

Not seeing what you expected here? Don't worry, your conversation data is preserved! Check back soon.

Upgrade to Plus

NEW

Dark mode

Updates & FAQ

Log out