**SPRING ACTUATOR**

* Spring Actuator là một module của Spring Boot cung cấp các công cụ giám sát và quản lý ứng dụng, bao gồm cả việc theo dõi và kiểm soát các service con (microservices) bên trong hệ thống của chúng ta
* Spring Actuator cung cấp một tập hợp các endpoint HTTP sẵn có, giúp chúng ta thu thập thông tin về trạng thái của ứng dụng và các service con một cách dễ dàng. Các endpoint này thường được sử dụng cho mục đích giám sát và quản lý ứng dụng trong môi trường product
* Một số endpoint quan trọng trong Spring Actuator:
  + */actuator/health*: Endpoint này cung cấp thông tin về trạng thái sức khỏe của ứng dụng và các service con. Nó thường được sử dụng bởi các công cụ và hệ thống giám sát để kiểm tra xem ứng dụng hoạt động bình thường hay có vấn đề gì xảy ra
  + */actuator/info*: Endpoint này cho phép chúng ta cung cấp các thông tin tùy chỉnh về ứng dụng, ví dụ như thông tin phiên bản, môi trường, mô tả ứng dụng, …
  + */actuator/metrics*: Endpoint này cung cấp các số liệu thống kê về hiệu suất của ứng dụng và các service con, chẳng hạn như thời gian phản hồi, lượng yêu cầu đã xử lý, …
  + */actuator/auditevents*: Endpoint này cho phép chúng ta truy vấn các sự kiện kiểm soát được ghi lại trong ứng dụng, giúp chúng ta theo dõi các hoạt động đáng chú ý
  + */actuator/env*: Endpoint này cung cấp thông tin về các biến môi trường trong ứng dụng
  + */actuator/loggers*: Endpoint này cho phép chúng ta quản lý cấu hình ghi log của ứng dụng, cho phép chúng ta bật/tắt log và thay đổi cấp độ log động
  + */actuator/beans*: Endpoint này cung cấp thông tin về các bean đã được tạo trong ứng dụng, giúp chúng ta hiểu rõ hơn về cấu trúc bean của ứng dụng
  + */actuator/threaddump*: Endpoint này cung cấp một threaddump của ứng dụng, giúp chúng ta phân tích các vấn đề liên quan đến luồng (thread) trong ứng dụng
* Demo:

A black text on a white background

Description automatically generated

A black text on a white background

Description automatically generated

A screenshot of a computer

Description automatically generated