Application?

An application program is a computer program designed to carry out a specific task other than one relating to the operation of the computer itself, typically to be used by end-users.

REST – Representational State Transfer

Controller?

the controller class is responsible for processing incoming REST API requests, preparing a model, and returning the view to be rendered as a response. The controller classes in Spring are annotated either by the @Controller or the @RestController annotation.

Actuators?

Monitoring our app, gathering metrics, understanding traffic, or the state of our database become trivial with this dependency.

The main benefit of this library is that we can get production-grade tools without having to actually implement these features ourselves.

Actuator is mainly used to expose operational information about the running application — health, metrics, info, dump, env, etc.

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

@Bean

public SecurityWebFilterChain securityWebFilterChain(

ServerHttpSecurity http) {

return http.authorizeExchange()

.pathMatchers("/actuator/\*\*").permitAll()

.anyExchange().authenticated()

.and().build();

}

Predefined Endpoints:

/auditevents lists security audit-related events such as user login/logout. Also, we can filter by principal or type among other fields.

/beans returns all available beans in our BeanFactory. Unlike /auditevents, it doesn't support filtering.

/conditions, formerly known as /autoconfig, builds a report of conditions around autoconfiguration.

/configprops allows us to fetch all @ConfigurationProperties beans.

/env returns the current environment properties. Additionally, we can retrieve single properties.

/flyway provides details about our Flyway database migrations.

/health summarizes the health status of our application.

/heapdump builds and returns a heap dump from the JVM used by our application.

/info returns general information. It might be custom data, build information or details about the latest commit.

/liquibase behaves like /flyway but for Liquibase.

/logfile returns ordinary application logs.

/loggers enables us to query and modify the logging level of our application.

/metrics details metrics of our application. This might include generic metrics as well as custom ones.

/prometheus returns metrics like the previous one, but formatted to work with a Prometheus server.

/scheduledtasks provides details about every scheduled task within our application.

/sessions lists HTTP sessions given we are using Spring Session.

/shutdown performs a graceful shutdown of the application.

/threaddump dumps the thread information of the underlying JVM