

SpringBoot Actuators

The Problem?

How can we monitor and manage our application?

How can I check the health of the application?

Solution: SB Actuators

- Exposes endpoints to monitor and manage the application.
 - REST endpoints are automatically added to our application
 - No need to write any additional code
- Add the dependency to pom.xml

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

- All the endpoints will be added prefixed with:
/actuator

/actuator/health -> To get the health information about our application.

localhost:8080/actuator/health

- The /info endpoint can provide information about application.

To expose the info endpoint we need to make changes in application.properties



```
management.endpoints.web.exposure.include=info
management.endpoints.web.exposure.include=* (To expose all endpoints)
```

/actuator/threaddump

- List all the threads running in our application

/actuator/mappings

- List all the request mappings for our application

What about Security?

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

We can override the default username and password

```
spring.security.user.name=Prasad  
spring.security.user.password=1234
```

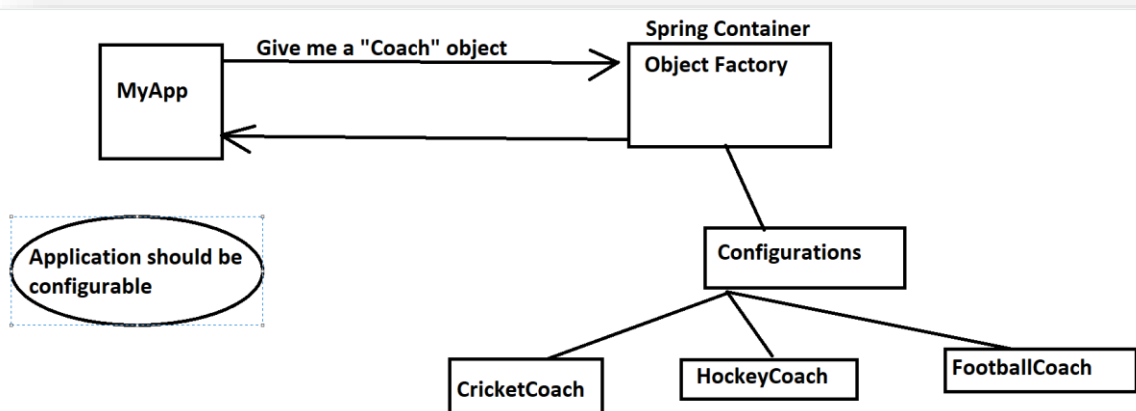
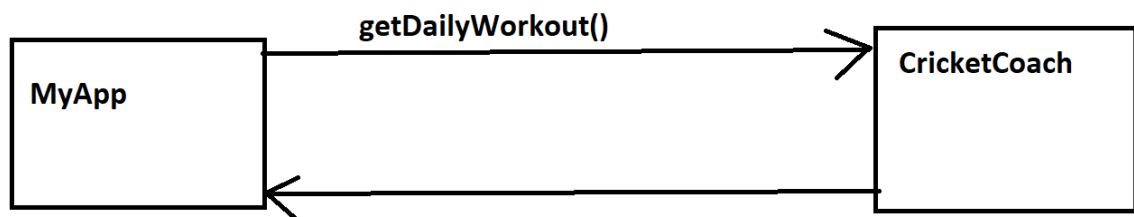
Running springboot application without IDE

1st Step: `mvnw package` command

2nd Step: `java -jar` <name of our project's JAR file>

Inversion of Control

The approach of outsourcing the construction and management of objects.



1. Spring Container

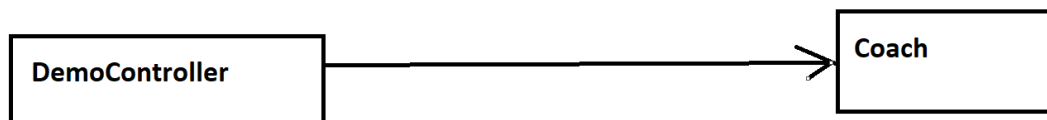
- Primary Purpose

- Create and manage (Inversion of Control)
- Inject the object dependency(dependency injection)

Spring Dependency Injection

- The dependency inversion principle - the client delegates to another object - the responsibility of providing its dependencies.

Example:



- Coach will provide daily workouts
- The democontroller wants to use the coach
 1. New Helper: Coach
 2. This is dependency
- Need to inject this dependency into the controller

Injection types

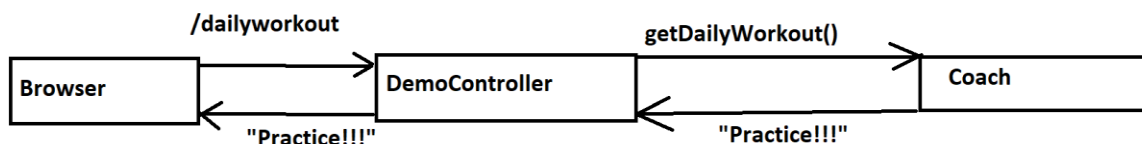
- There are many types of injection in spring
- Will take only two
 1. Constructor Injection
 2. Setter Injection

When to use one?

- Constructor Injection
 - will use it when we have all the required dependencies
- Setter Injection
 - Will use it when we have some optional dependencies.

What is Spring Autowiring?

- For DI, spring uses autowiring
- Autowiring example
 - Injecting a coach interface
 - 1. Spring will scan for @Component or a class which is annotated with @Component
 - 2. Will ask does any one implements coach interface??
 - 3. If so, lets inject them. For Ex. cricketCoach



Step1. Define the dependency interface and class
(interface coach,cricketCoach implements coach)

Step2. Create DemoController

Step3. Create a constructor in DemoController class
for injections