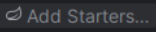


Assignment - Spring Boot

Q1) Create a sample spring boot application and configure H2 in memory database dependency.

```
<properties>
  <java.version>21</java.version>
</properties>
<dependencies> 
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>

  <dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <scope>runtime</scope>
  </dependency>

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

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>
</dependencies>
```

Q2)Configure few properties under application properties file and read there value under Main class.

```
package com.akash.springboot;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Application implements CommandLineRunner {


    @Value("${app.name}")
    private String appName;

    @Value("${app.version}")
    private String appVersion;

    @Override
    public void run(String... args) throws Exception {
        System.out.println("App Name : " + appName);
        System.out.println("App Version : " + appVersion);
    }

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

}
```



```
spring.application.name=SpringBoot
spring.h2.console.enabled=true
spring.h2.console.path=/h2-console

spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=akash
spring.datasource.password=Pass_word0

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.jpa.hibernate.ddl-auto=update

# Custom properties
app.name=EmployeeApp
app.version=1.0.0
```

Q3) Create a Employee table (id, name, designation) under H2 database and inserts few sample records under that table at startup.

```
package com.akash.springboot.config;

import com.akash.springboot.model.Employee;
import com.akash.springboot.repository.EmployeeRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.stereotype.Component;

@Component
public class DataInitializer implements CommandLineRunner {

    private final EmployeeRepository repository; 4 usages

    public DataInitializer(EmployeeRepository repository) {
        this.repository = repository;
    }

    @Override
    public void run(String... args) {
        repository.save(new Employee(id: null, name: "Alice", designation: "Developer"));
        repository.save(new Employee(id: null, name: "Bob", designation: "Tester"));
        repository.save(new Employee(id: null, name: "Charlie", designation: "Manager"));
        System.out.println("Sample employees added to H2 database.");
    }
}
```

```
package com.akash.springboot.model;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
public class Employee {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name; 3 usages
    private String designation; 3 usages

    public Employee() {} // Required by JPA

    public Employee(Long id, String name, String designation) { 3 usages
        this.id = id;
        this.name = name;
        this.designation = designation;
    }

    // Getters and setters
    public Long getId() { return id; } no usages
    public void setId(Long id) { this.id = id; } no usages

    public String getName() { return name; } no usages
    public void setName(String name) { this.name = name; } no usages

    public String getDesignation() { return designation; } no usages
    public void setDesignation(String designation) { this.designation = designation; } no usages
}
```

```
public class Employee {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name; 3 usages
    private String designation; 3 usages

    public Employee() {} // Required by JPA

    public Employee(Long id, String name, String designation) { 3 usages
        this.id = id;
        this.name = name;
        this.designation = designation;
    }

    // Getters and setters
    public Long getId() { return id; } no usages
    public void setId(Long id) { this.id = id; } no usages

    public String getName() { return name; } no usages
    public void setName(String name) { this.name = name; } no usages

    public String getDesignation() { return designation; } no usages
    public void setDesignation(String designation) { this.designation = designation; } no usages
}
```

```

package com.akash.springboot.repository;

import com.akash.springboot.model.Employee;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository no usages
public interface EmployeeRepository extends JpaRepository<Employee, Long> {
}

```

```

2025-07-21T06:29:08.419+05:30 INFO 15146 --- [SpringBoot] [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available
2025-07-21T06:29:08.449+05:30 INFO 15146 --- [SpringBoot] [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory
2025-07-21T06:29:08.561+05:30 WARN 15146 --- [SpringBoot] [ restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by
2025-07-21T06:29:08.830+05:30 INFO 15146 --- [SpringBoot] [ restartedMain] o.s.b.a.h2.H2ConsoleAutoConfiguration : H2 console available at '/h2-console'
2025-07-21T06:29:08.852+05:30 INFO 15146 --- [SpringBoot] [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port
2025-07-21T06:29:08.857+05:30 INFO 15146 --- [SpringBoot] [ restartedMain] o.s.b.a.e.web.EndpointLinksResolver : Exposing 1 endpoint beneath base path
2025-07-21T06:29:08.900+05:30 INFO 15146 --- [SpringBoot] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) wi
2025-07-21T06:29:08.908+05:30 INFO 15146 --- [SpringBoot] [ restartedMain] com.akash.springboot.Application : Started Application in 2.588 seconds
Sample employees added to H2 database.
2025-07-21T06:29:09.154+05:30 INFO 15146 --- [SpringBoot] [on(3)-127.0.0.1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet
2025-07-21T06:29:09.154+05:30 INFO 15146 --- [SpringBoot] [on(3)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-07-21T06:29:09.155+05:30 INFO 15146 --- [SpringBoot] [on(3)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms

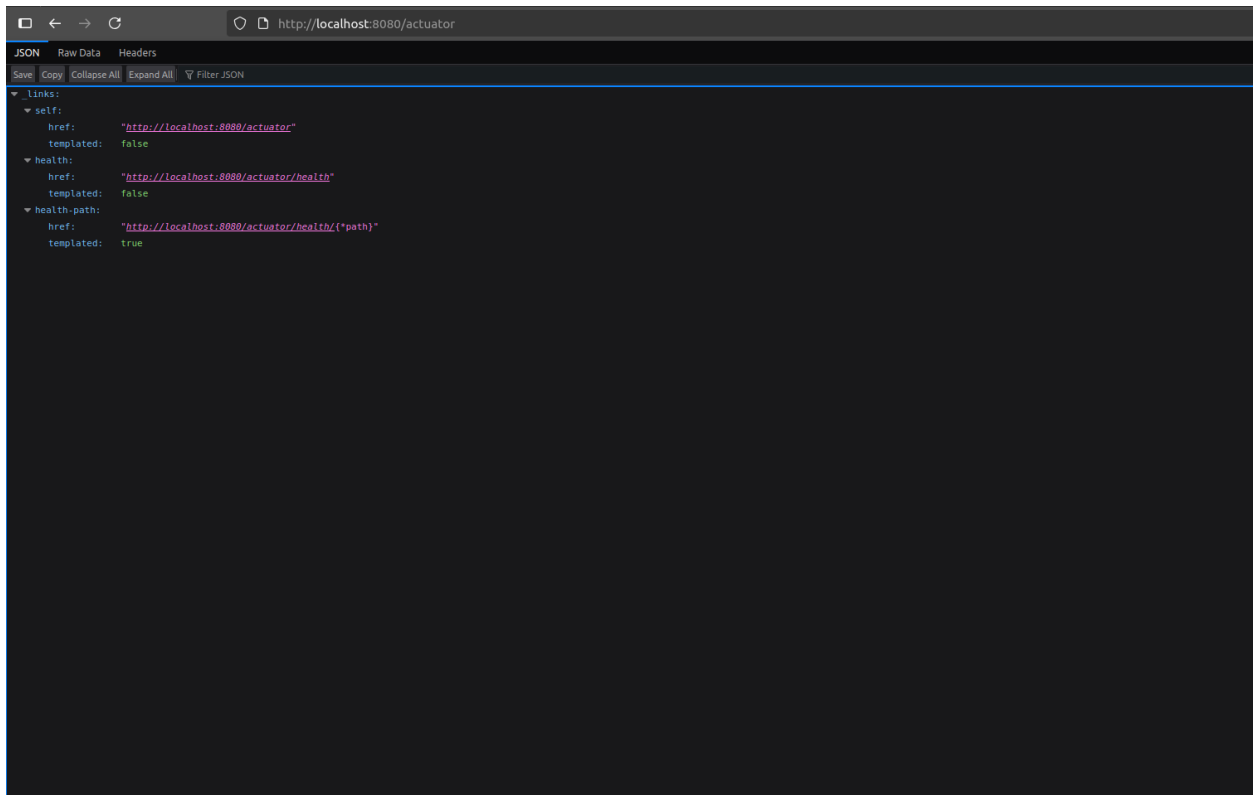
```

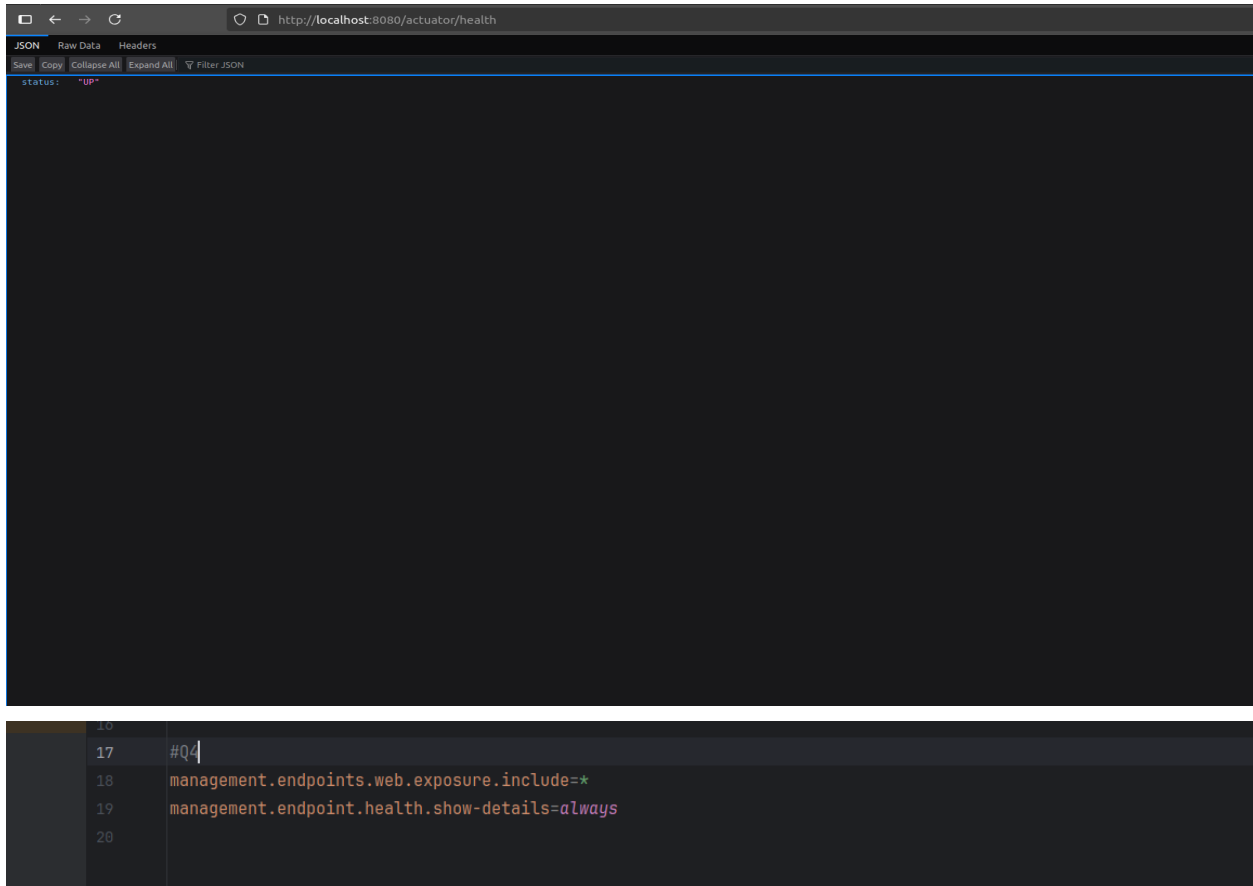
The screenshot shows the H2 console web interface. The browser address bar displays the URL: `http://localhost:h2-console/login.do?sessionId=d0249f2a1bca04056a2ecb9d82a53e91`. The interface includes a sidebar on the left with navigation links: `jdbc:h2:mem:testdb`, `EMPLOYEE`, `INFORMATION_SCHEMA`, `Users`, and `H2 2.3.232 (2024-08-11)`. The main area has a toolbar with buttons for `Run`, `Run Selected`, `Auto complete`, and `Clear`, along with a text input for the `SQL statement:`. Below the toolbar, the SQL query `SELECT * FROM EMPLOYEE;` is entered. The execution results are displayed as a table with 3 rows and 3 columns: `ID`, `DESIGNATION`, and `NAME`. The data rows are: `1 Developer Alice`, `2 Tester Bob`, and `3 Manager Charlie`. The status below the table indicates `(3 rows, 3 ms)`. An `Edit` button is located at the bottom of the results area.

ID	DESIGNATION	NAME
1	Developer	Alice
2	Tester	Bob
3	Manager	Charlie

(3 rows, 3 ms)

Q4) Enable actuator services and play around few endpoints it exposes.





Q5) Enable devtools and play around by making changes.

