

h2 console spring boot

H2 Console in Spring Boot

1. What is H2 Database?

H2 is a lightweight, in-memory database commonly used for **development, testing, and demos** in Spring Boot applications.

Key characteristics:

- Embedded database
- Runs in memory (no installation required)
- Fast startup
- Automatically resets when the application stops
- Provides a **web-based console** for viewing tables and running SQL queries

Spring Boot has **built-in auto configuration** for H2.

2. What is H2 Console?

The **H2 Console** is a **browser-based UI** that allows you to:

- View database tables
- Run SQL queries
- Inspect data
- Debug database issues
- Test SQL quickly

Example URL after enabling:

```
http://localhost:8080/h2-console
```



3. Architecture Overview

Spring Boot Application

```
|
| JDBC / JPA
v
H2 In-Memory Database
|
v
H2 Web Console
(Browser UI)
```



4. Step-by-Step Setup

Step 1 — Add Dependency

Maven

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

Gradle

```
<> gradle
```



Step 2 — Configure application.properties

```
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=sa
spring.datasource.password=
```



```
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
```

5. Important Properties Explained

Property	Description
<code>spring.h2.console.enabled</code>	Enables the H2 web console
<code>spring.h2.console.path</code>	URL path for console
<code>jdbc:h2:mem:testdb</code>	Creates an in-memory database
<code>username=sa</code>	Default H2 username
<code>password=</code>	Empty password by default

6. Running the Application

Start the Spring Boot application.

Open browser:

```
http://localhost:8080/h2-console
```



You will see the H2 login page.

7. H2 Console Login Configuration

Use the following values:

Field	Value
JDBC URL	<code>jdbc:h2:mem:testdb</code>
User Name	<code>sa</code>

Field	Value
Password	<i>(leave empty)</i>

Click **Connect**.

8. Example Spring Boot Entity

User Entity

```
import jakarta.persistence.*;

@Entity
public class User {

    @Id
    @GeneratedValue
    private Long id;

    private String name;

    private String email;
}
```

9. Repository

```
import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> {
    ~
    // save
}
```

10. Sample Data Insert

```
import org.springframework.boot.CommandLineRunner;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class DataLoader {

    @Bean
    CommandLineRunner loadData(UserRepository repo) {
        return args -> {
            repo.save(new User(null, "John", "john@gmail.com"));
            repo.save(new User(null, "Alice", "alice@gmail.com"));
        };
    }
}
```

11. Viewing Data in H2 Console

After application starts:

Run SQL:

</> SQL



Output Example

ID	NAME	EMAIL
1	John	john@gmail.com
2	Alice	alice@gmail.com

12. Useful SQL Commands

Create table:

```
CREATE TABLE product(  
  id INT PRIMARY KEY,  
  name VARCHAR(100)  
  ,
```

```
<> SQL
```

Insert data:

```
<> SQL
```

Query:

```
<> SQL
```

13. Common Problems & Fixes

Problem 1 — Whitelabel Error Page

Cause:

Spring Security blocking console.

Solution:

Add configuration:

```
@Bean  
SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {  
  
    http  
        .csrf(csrf -> csrf.disable())  
        .headers(headers -> headers.frameOptions(frame -> frame.disable()));  
  
    return http.build();  
}
```

Problem 2 — Database Not Found

Cause:

Wrong JDBC URL.

Correct:

```
jdbc:h2:mem:testdb
```



Problem 3 — Tables Missing

Reason:

In-memory database resets when application restarts.

Solution:

Use file database.

```
spring.datasource.url=jdbc:h2:file:./data/testdb
```



14. In-Memory vs File Database

Feature	In-Memory	File Based
Persistence	✗ No	✓ Yes
Speed	Very Fast	Fast
Restart Data	Lost	Saved
Best Use	Testing	Local Dev

15. Best Practices

✔ Use H2 for:

- Development
- Unit testing
- Prototyping

✘ Do NOT use H2 for:

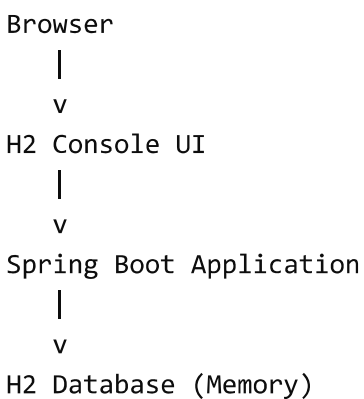
- Production systems
- Large scale applications

16. Quick Summary

Feature	Description
H2 Database	Lightweight embedded DB
H2 Console	Web UI for DB
Default URL	/h2-console
Default User	sa
Password	empty
Best Use	Development & Testing


17. Visual Flow

```
Browser
  |
  v
H2 Console UI
  |
  v
Spring Boot Application
  |
  v
H2 Database (Memory)
```



If you want, I can also generate:

- **Complete Spring Boot Project with H2 + JPA**
- **Common H2 Console Errors**
- **Spring Boot + H2 + Hibernate Example**
- **Spring Boot Interview Questions on H2**

Just tell me .



pdf download need all data displayed here

Your PDF is ready.

 **Download it here:**

[Download the PDF](#)

The document includes:

- All headings and sections
- Code blocks
- Tables
- Architecture diagrams
- SQL examples
- Spring Boot configuration
- Troubleshooting section

Formatted for **A4 (Portrait)** with proper margins and readable fonts suitable for training or documentation.

If you want, I can also generate PDFs for other Spring Boot topics like:

- **Dependency Injection**
- **Bean Scopes**
- **Spring Boot Auto Configuration**

- **Spring Data JPA**
- **Spring Security Basics**  



html download

Your HTML file is ready.



 **Download it here:**

[Download the HTML](#)

You can:

- Open it in any **web browser**
- Edit it in **VS Code / Notepad++**
- Convert it later to **PDF, DOCX, or PPT**
- Host it as **documentation or tutorial page**

If you want, I can also generate:

- **Professional styled HTML documentation**
- **Spring Boot notes website format**
- **Printable training manual HTML**
- **Complete Spring Boot documentation bundle (HTML + PDF + PPT)**  



 Spring Boot Doc Generator Auto 

