

Spring Boot Actuator - Security



What about Security?

What about Security?

- You may **NOT** want to expose all of this information

```
{  
  "contexts": {  
    "application": {  
      "beans": {  
        "endpointCachingOperationInvokerAdvisor": {  
          "aliases": [],  
          "scope": "singleton",  
          "type": "org.springframework.boot.actuate.endpoint.invoker.cache.Cach  
          "resource": "class path resource [org/springframework/boot/actuate/au  
          "dependencies": [  
            "environment"  
          ]  
        },  
        "defaultServletHandlerMapping": {  
          "aliases": [],  
          "scope": "singleton",  
          "type": "org.springframework.web.servlet.HandlerMapping",  
          "resource": "class path resource [org/springframework/boot/autoconfig  
          "dependencies": []  
        },  
        "org.springframework.boot.autoconfigure.web.servlet.WebMvcAutoConfigura  
          "aliases": [],  
          "scope": "singleton",  
          "type":  
        }  
      }  
    }  
  }  
}
```

What about Security?

- You may **NOT** want to expose all of this information
- Add Spring Security to project and endpoints are secured :-)

What about Security?

- You may **NOT** want to expose all of this information
- Add Spring Security to project and endpoints are secured :-)

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

What about Security?

- You may **NOT** want to expose all of this information
- Add Spring Security to project and endpoints are secured :-)

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

/health and /info
are still available
You can disable them
if you want

Secured Endpoints

Secured Endpoints

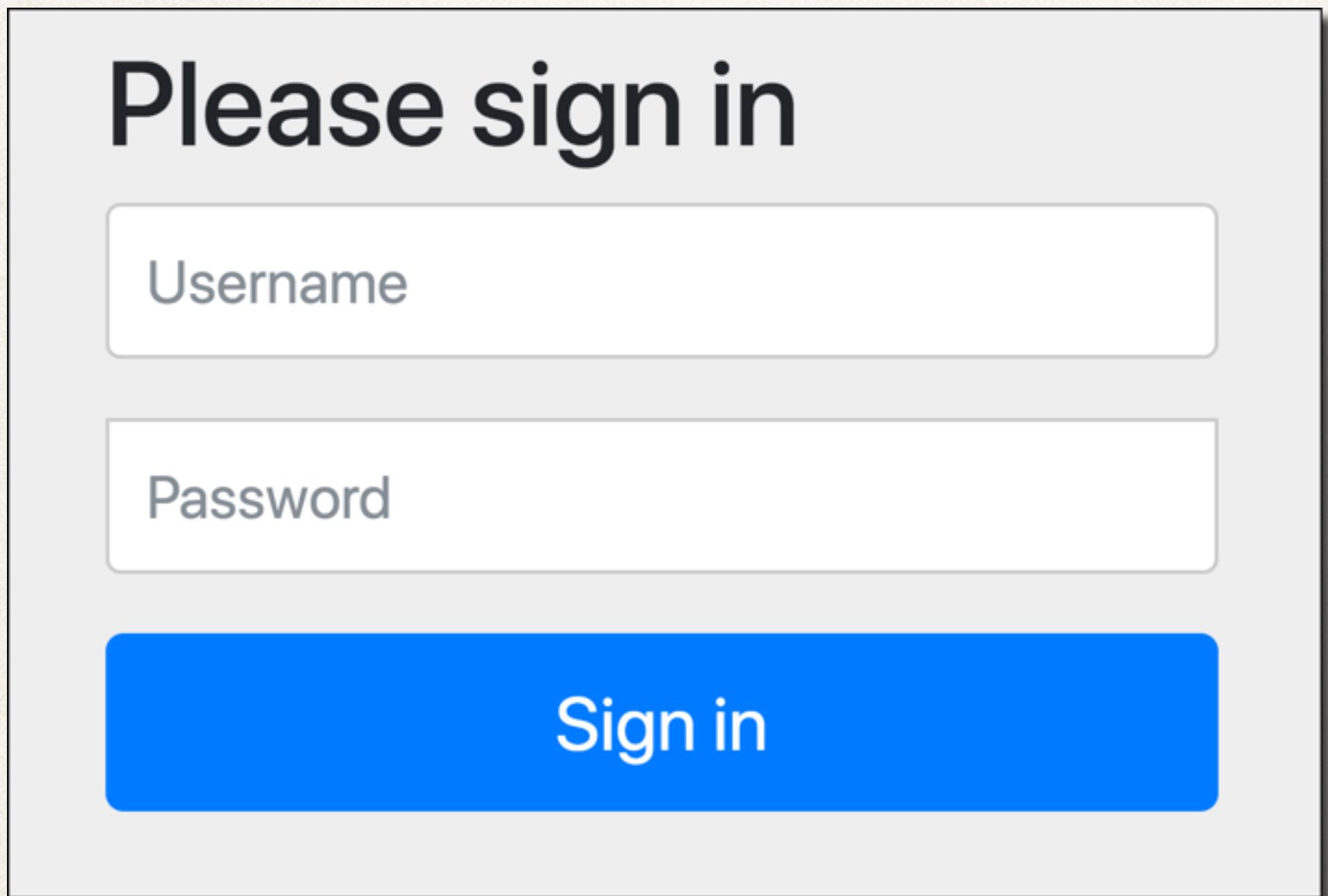
- Now when you access: **/actuator/beans**

Secured Endpoints

- Now when you access: **/actuator/beans**
- Spring Security will prompt for login

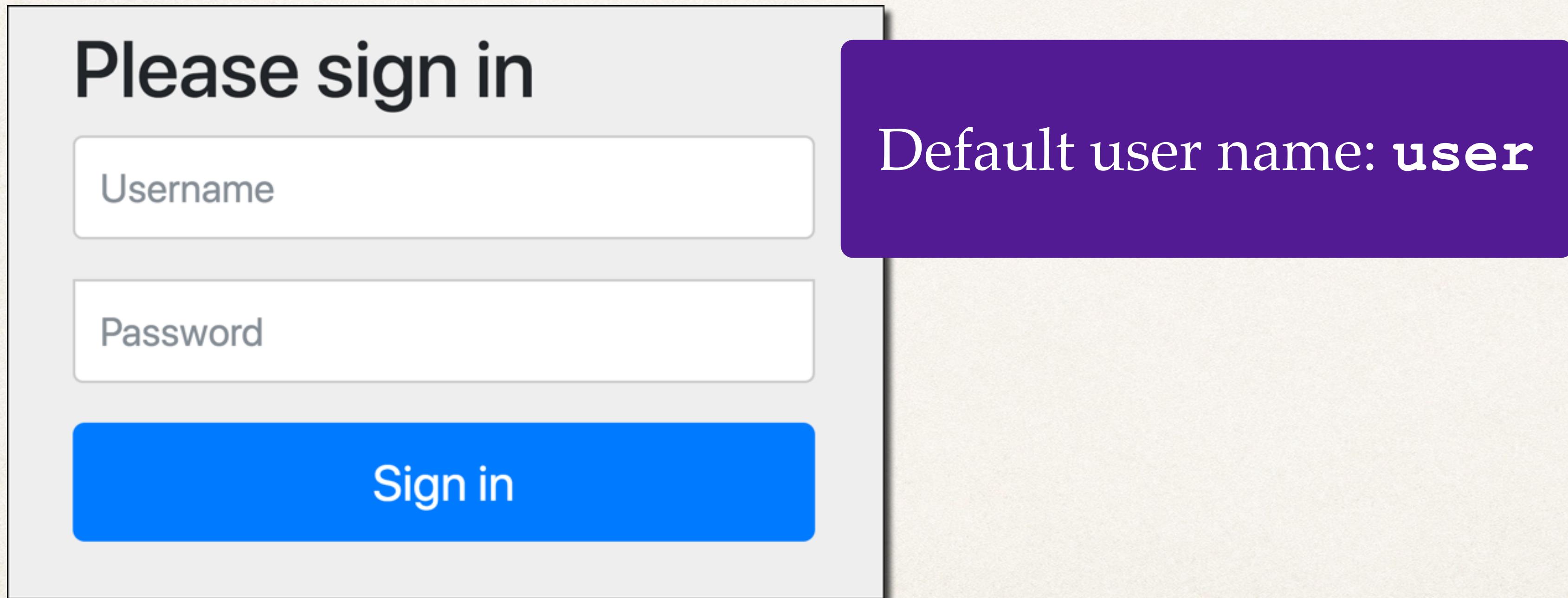
Secured Endpoints

- Now when you access: **/actuator/beans**
- Spring Security will prompt for login



Secured Endpoints

- Now when you access: **/actuator/beans**
- Spring Security will prompt for login



Secured Endpoints

- Now when you access: **/actuator/beans**
- Spring Security will prompt for login

The image shows a Spring Security login interface and a terminal window. The login page has fields for 'Username' and 'Password', and a 'Sign in' button. A purple callout box says 'Default user name: user'. The terminal window shows a log entry with a red box around the generated password: 'Using generated security password: 78fd68a6-c190-421d-934b-df7852fc7dc2'.

Please sign in

Username

Password

Sign in

Default user name: **user**

```
11-02 21:05:57.074 INFO 24986 --- [main] .s.s.UserDetailsSe
Using generated security password: 78fd68a6-c190-421d-934b-df7852fc7dc2
```

Check console logs
for password

Spring Security configuration

Spring Security configuration

- You can override default user name and generated password

Spring Security configuration

- You can override default user name and generated password

File: `src/main/resources/application.properties`

```
spring.security.user.name=scott  
spring.security.user.password=tiger
```

Customizing Spring Security

Customizing Spring Security

- You can customize Spring Security for Spring Boot Actuator

Customizing Spring Security

- You can customize Spring Security for Spring Boot Actuator
 - Use a database for roles, encrypted passwords etc ...

Customizing Spring Security

- You can customize Spring Security for Spring Boot Actuator
 - Use a database for roles, encrypted passwords etc ...
- Follow the same techniques as regular Spring Security

```
public class DemoSecurityConfig extends WebSecurityConfigurerAdapter {  
  
    @Autowired  
    private DataSource securityDataSource;  
  
    @Override  
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
  
        auth.jdbcAuthentication().dataSource(securityDataSource);  
    }  
    ...  
}
```

Customizing Spring Security

- You can customize Spring Security for Spring Boot Actuator
 - Use a database for roles, encrypted passwords etc ...
- Follow the same techniques

```
public class DemoSecurityConfig extends WebSecurityConfigurerAdapter {  
  
    @Autowired  
    private DataSource securityDataSource;  
  
    @Override  
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
  
        auth.jdbcAuthentication().dataSource(securityDataSource);  
    }  
    ...  
}
```

Read security info
from database

Security

Customizing Spring Security

- You can customize Spring Security for Spring Boot Actuator
 - Use a database for roles, encrypted passwords etc ...
- Follow the same techniques as regular Spring Security

```
public class DemoSecurityConfig extends WebSecurityConfigurerAdapter {  
  
    @Autowired  
    private DataSource securityDataSource;  
  
    @Override  
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
        auth.jdbcAuthentication().dataSource(securityDataSource);  
    }  
    ...  
}
```

```
@Override  
protected void configure(HttpSecurity http) throws Exception {  
  
    http.authorizeRequests()  
        .antMatchers("/actuator/**").hasRole("ADMIN")  
        ...  
}
```

Only authorize
ADMIN users

Excluding Endpoints

Excluding Endpoints

- To exclude `/health` and `/info`

Excluding Endpoints

- To exclude **/health** and **/info**

File: `src/main/resources/application.properties`

```
# Exclude individual endpoints with a comma-delimited list
#
management.endpoints.web.exposure.exclude=health,info
```

Spring Boot Actuator Documentation

Spring Boot Actuator Documentation

For more details
www.luv2code.com/actuator-docs

Development Process

Step-By-Step

Development Process

Step-By-Step

1. Edit **pom.xml** and add **spring-boot-starter-security**

Development Process

Step-By-Step

1. Edit **pom.xml** and add **spring-boot-starter-security**
2. Verify security on actuator endpoints for: **/beans** etc

Development Process

Step-By-Step

1. Edit **pom.xml** and add **spring-boot-starter-security**
2. Verify security on actuator endpoints for: **/beans** etc
3. Disable endpoints for **/health** and **/info**