

K

[Start here <<](#) [Data Structure](#) [Algorithm](#) [Java](#) [Spring Boot](#) [JPA and Hibernate](#) [RESTful API](#)

Custom Software Experts

Database Software Programs

Reliable and Affordable since 1994.

genesisdatabases.com

[OPEN](#)

Registration + Login Tutorial and Example with Spring Boot, Spring Security, Spring Data JPA, Hibernate, HSQL, JSP and Bootstrap

Last modified @ 05 May 2019

[# Security](#)

[# Spring Boot](#)

This tutorial will walk you through the process of creating a simple User Account Registration + Login Example with Spring Boot, Spring Security, Spring Data JPA, Hibernate, HSQL, JSP and Bootstrap.

If you are new to Spring Boot or Spring Data JPA, it would be best to try the following tutorials

- [Spring Boot Hello World Example with JSP](#)
- [JPA/Hibernate Many To Many Bidirectional Relationship Mapping](#)

What you'll build

Log in

The screenshot shows a browser window with the title "Log in with your account". The address bar displays "localhost:8080/login?error". The main content is a "Log in" form. The "Username" field contains "hellokoding" and the "Password" field contains a series of dots ("....."). Below the form, a red error message reads "Your username and password is invalid." A blue "Log In" button is centered below the fields, and a "Create an account" link is located at the bottom.

Log out

The screenshot shows a browser window with the title "Log in with your account". The address bar displays "localhost:8080/login?logout". The main content is a "Log in" form. Above the form, a message reads "You have been logged out successfully." The "Username" and "Password" fields are empty. A blue "Log In" button is centered below the fields, and a "Create an account" link is located at the bottom.

Register account

Create your account

hellokoding

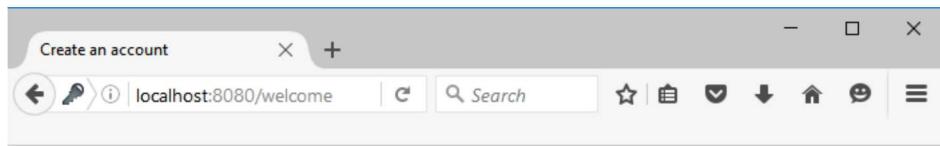
Someone already has that username.

Try one with at least 8 characters.

These passwords don't match.

Submit

Welcome



Welcome hellokoding | [Logout](#)

What you'll need

- JDK 8+ or OpenJDK 8+
- Maven 3+

Stack

- Spring Boot 2 and Spring Security
- Spring Data JPA, Hibernate 5 and HSQL
- JSP

Custom Software Experts - Database Softw Programs

Ad Reliable and Affordable since 1994.

genesisdatabases.com

[Learn more](#)

Init project structure and dependencies

Project structure

```

src
└── main
    └── java
        └── com
            └── hellokoding
                └── auth
                    ├── model
                    │   ├── Role.java
                    │   └── User.java
                    ├── repository
                    │   ├── RoleRepository.java
                    │   └── UserRepository.java
                    ├── service
                    │   ├── SecurityServiceImpl.java
                    │   ├── SecurityService.java
                    │   ├── UserDetailsServiceImpl.java
                    │   ├── UserServiceImpl.java
                    │   └── UserService.java
                    ├── validator
                    │   └── UserValidator.java
                    ├── web
                    │   └── UserController.java
                    └── WebApplication.java
                    └── WebSecurityConfig.java
    └── resources
        ├── application.properties
        └── validation.properties
    └── webapp
        ├── resources
        │   ├── css
        │   │   ├── bootstrap.min.css
        │   │   └── common.css
        │   └── js
        │       └── bootstrap.min.js
        ├── login.jsp
        ├── registration.jsp
        └── welcome.jsp

```

pom.xml

Project dependencies

[pom.xml](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLoca
  <modelVersion>4.0.0</modelVersion>
  <artifactId>auth</artifactId>
  <name>auth</name>
  <description>auth</description>
  <packaging>war</packaging>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.1.1.RELEASE</version>
  </parent>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <java.version>1.8</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>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-security</artifactId>
    </dependency>

    <dependency>
      <groupId>org.hsqldb</groupId>
      <artifactId>hsqldb</artifactId>
      <scope>runtime</scope>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-tomcat</artifactId>
      <scope>provided</scope>
    </dependency>

    <dependency>
      <groupId>org.apache.tomcat.embed</groupId>
      <artifactId>tomcat-embed-jasper</artifactId>
      <scope>provided</scope>
    </dependency>

    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>jstl</artifactId>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
```

```

<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>

```



Custom Software Experts - Database Software Programs

Ad Reliable and Affordable since 1994.

genesisdatabases.com

[Learn more](#)

Define JPA Entities and Repositories

JPA and Hibernate Entities

JPA Entity is defined with `@Entity` annotation, represent a table in your database.

User.java

```

package com.hellokoding.auth.model;

import javax.persistence.*;
import java.util.Set;

@Entity
@Table(name = "user")
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String username;

    private String password;

    @Transient
    private String passwordConfirm;

    @ManyToMany
    private Set<Role> roles;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }
}

```

```

public String getUsername() {
    return username;
}

public void setUsername(String username) {
    this.username = username;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

public String getPasswordConfirm() {
    return passwordConfirm;
}

public void setPasswordConfirm(String passwordConfirm) {
    this.passwordConfirm = passwordConfirm;
}

public Set<Role> getRoles() {
    return roles;
}

public void setRoles(Set<Role> roles) {
    this.roles = roles;
}
}

```

[Role.java](#)

```

package com.hellokoding.auth.model;

import javax.persistence.*;
import java.util.Set;

@Entity
@Table(name = "role")
public class Role {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    @ManyToMany(mappedBy = "roles")
    private Set<User> users;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }
}

```

```

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public Set<User> getUsers() {
    return users;
}

public void setUsers(Set<User> users) {
    this.users = users;
}
}

```

`@Table` maps the entity with the table. If no `@Table` is defined, the default value is used: the class name of the entity.

`@Id` declares the identifier property of the entity.

`@ManyToMany` defines a many-to-many relationship between 2 entities.

`mappedBy` indicates the entity is the inverse of the relationship.

Spring Data JPA Repositories

Spring Data JPA contains some built-in `Repository` implemented common functions to work with database such as `findOne`, `findAll` and `save`

UserRepository.java

```

package com.hellokoding.auth.repository;

import com.hellokoding.auth.model.User;
import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> {
    User findByUsername(String username);
}

```

RoleRepository.java

```

package com.hellokoding.auth.repository;

import com.hellokoding.auth.model.Role;
import org.springframework.data.jpa.repository.JpaRepository;

public interface RoleRepository extends JpaRepository<Role, Long>{
}

```

Custom Software Experts - Database Software Programs

Ad Reliable and Affordable since 1994.

genesisdatabases.com

[Learn more](#)

Define Services

Spring Security's UserDetailsService

To implement login/authentication with Spring Security, we need to implement

`org.springframework.security.core.userdetails.UserDetailsService` interface

UserDetailsServiceImpl.java

```
package com.hellokoding.auth.service;

import com.hellokoding.auth.model.Role;
import com.hellokoding.auth.model.User;
import com.hellokoding.auth.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.authority.SimpleGrantedAuthority;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import java.util.HashSet;
import java.util.Set;

@Service
public class UserDetailsServiceImpl implements UserDetailsService{
    @Autowired
    private UserRepository userRepository;

    @Override
    @Transactional(readOnly = true)
    public UserDetails loadUserByUsername(String username) {
        User user = userRepository.findByUsername(username);
        if (user == null) throw new UsernameNotFoundException(username);

        Set<GrantedAuthority> grantedAuthorities = new HashSet<>();
        for (Role role : user.getRoles()){
            grantedAuthorities.add(new SimpleGrantedAuthority(role.getName()));
        }

        return new org.springframework.security.core.userdetails.User(user.getUsername(), user.getPassword(), grantedAuthorities);
    }
}
```

Security Service

We create `SecurityService` to provide current logged-in user and auto login user after registration

SecurityService.java

```
package com.hellokoding.auth.service;

public interface SecurityService {
    String findLoggedInUsername();

    void autoLogin(String username, String password);
}
```

SecurityServiceImpl.java

```
package com.hellokoding.auth.service;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.authentication.AuthenticationManager;
import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.stereotype.Service;

@Service
public class SecurityServiceImpl implements SecurityService{
    @Autowired
    private AuthenticationManager authenticationManager;

    @Autowired
    private UserDetailsService userDetailsService;

    private static final Logger logger = LoggerFactory.getLogger(SecurityServiceImpl.class);

    @Override
    public String findLoggedInUsername() {
        Object userDetails = SecurityContextHolder.getContext().getAuthentication().getDetails();
        if (userDetails instanceof UserDetails) {
            return ((UserDetails)userDetails).getUsername();
        }
        return null;
    }

    @Override
    public void autoLogin(String username, String password) {
        UserDetails userDetails = userDetailsService.loadUserByUsername(username);
        UsernamePasswordAuthenticationToken usernamePasswordAuthenticationToken = new UsernamePasswordAuthenticationTok
            authenticationManager.authenticate(usernamePasswordAuthenticationToken);

        if (usernamePasswordAuthenticationToken.isAuthenticated()) {
            SecurityContextHolder.getContext().setAuthentication(usernamePasswordAuthenticationToken);
            logger.debug(String.format("Auto login %s successfully!", username));
        }
    }
}
```

```

    }
}
}
```

User Service

Provide service for registering account

UserService.java

```

package com.hellokoding.auth.service;

import com.hellokoding.auth.model.User;

public interface UserService {
    void save(User user);

    User findByUsername(String username);
}
```

UserServiceImpl.java

```

package com.hellokoding.auth.service;

import com.hellokoding.auth.model.User;
import com.hellokoding.auth.repository.RoleRepository;
import com.hellokoding.auth.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Service;

import java.util.HashSet;

@Service
public class UserServiceImpl implements UserService {
    @Autowired
    private UserRepository userRepository;
    @Autowired
    private RoleRepository roleRepository;
    @Autowired
    private BCryptPasswordEncoder bCryptPasswordEncoder;

    @Override
    public void save(User user) {
        user.setPassword(bCryptPasswordEncoder.encode(user.getPassword()));
        user.setRoles(new HashSet<>(roleRepository.findAll()));
        userRepository.save(user);
    }

    @Override
    public User findByUsername(String username) {
        return userRepository.findByUsername(username);
    }
}
```

Custom Software Experts - Database Softw Programs

Ad Reliable and Affordable since 1994.

genesisdatabases.com

[Learn more](#)

Define Validator, Controllers and View Templates

Spring Validator

To provide input-data validation for `/registration` controller with Spring Validator, we implement `org.springframework.validation.Validator`. Error codes, e.g. `Size.userForm.username`, are defined by [validation.properties](#)

UserValidator.java

```
package com.hellokoding.auth.validator;

import com.hellokoding.auth.model.User;
import com.hellokoding.auth.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import org.springframework.validation.Errors;
import org.springframework.validation.ValidationUtils;
import org.springframework.validation.Validator;

@Component
public class UserValidator implements Validator {
    @Autowired
    private UserService userService;

    @Override
    public boolean supports(Class<?> aClass) {
        return User.class.equals(aClass);
    }

    @Override
    public void validate(Object o, Errors errors) {
        User user = (User) o;

        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "username", "NotEmpty");
        if (user.getUsername().length() < 6 || user.getUsername().length() > 32) {
            errors.rejectValue("username", "Size.userForm.username");
        }
        if (userService.findByUsername(user.getUsername()) != null) {
            errors.rejectValue("username", "Duplicate.userForm.username");
        }

        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "password", "NotEmpty");
        if (user.getPassword().length() < 8 || user.getPassword().length() > 32) {
            errors.rejectValue("password", "Size.userForm.password");
        }
    }
}
```

```
    if (!user.getPasswordConfirm().equals(user.getPassword())) {
        errors.rejectValue("passwordConfirm", "Diff.userForm.passwordConfirm");
    }
}
}
```

Controllers

UserController.java

```
package com.hellokoding.auth.web;

import com.hellokoding.auth.model.User;
import com.hellokoding.auth.service.SecurityService;
import com.hellokoding.auth.service.UserService;
import com.hellokoding.auth.validator.UserValidator;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.*;

@Controller
public class UserController {
    @Autowired
    private UserService userService;

    @Autowired
    private SecurityService securityService;

    @Autowired
    private UserValidator userValidator;

    @GetMapping("/registration")
    public String registration(Model model) {
        model.addAttribute("userForm", new User());

        return "registration";
    }

    @PostMapping("/registration")
    public String registration(@ModelAttribute("userForm") User userForm, BindingResult bindingResult) {
        userValidator.validate(userForm, bindingResult);

        if (bindingResult.hasErrors()) {
            return "registration";
        }

        userService.save(userForm);

        securityService.autoLogin(userForm.getUsername(), userForm.getPasswordConfirm());

        return "redirect:/welcome";
    }

    @GetMapping("/login")
    public String login(Model model, String error, String logout) {
        if (error != null)
            model.addAttribute("error", "Your username and password is invalid.");
    }
}
```

```

if (logout != null)
    model.addAttribute("message", "You have been logged out successfully.");

return "login";
}

@GetMapping={"/", "/welcome"})
public String welcome(Model model) {
    return "welcome";
}
}

```

We don't define /login POST controller, it is provided by Spring Security

JSP View Templates with Bootstrap

registration.jsp

```

<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<c:set var="contextPath" value="${pageContext.request.contextPath}"/>

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>Create an account</title>

        <link href="${contextPath}/resources/css/bootstrap.min.css" rel="stylesheet">
        <link href="${contextPath}/resources/css/common.css" rel="stylesheet">
    </head>

    <body>

        <div class="container">

            <form:form method="POST" modelAttribute="userForm" class="form-signin">
                <h2 class="form-signin-heading">Create your account</h2>
                <spring:bind path="username">
                    <div class="form-group ${status.error ? 'has-error' : ''}">
                        <form:input type="text" path="username" class="form-control" placeholder="Username"
                                    autofocus="true"></form:input>
                        <form:errors path="username"></form:errors>
                    </div>
                </spring:bind>

                <spring:bind path="password">
                    <div class="form-group ${status.error ? 'has-error' : ''}">
                        <form:input type="password" path="password" class="form-control" placeholder="Password"></form:input>
                        <form:errors path="password"></form:errors>
                    </div>
                </spring:bind>

                <spring:bind path="passwordConfirm">
                    <div class="form-group ${status.error ? 'has-error' : ''}">
                        <form:input type="password" path="passwordConfirm" class="form-control"
                                    placeholder="Confirm your password"></form:input>

```

```

        <form:errors path="passwordConfirm"></form:errors>
    </div>
</spring:bind>

    <button class="btn btn-lg btn-primary btn-block" type="submit">Submit</button>
</form:>
</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script src="${contextPath}/resources/js/bootstrap.min.js"></script>
</body>
</html>

```

login.jsp

```

<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>

<c:set var="contextPath" value="${pageContext.request.contextPath}" />

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Log in with your account</title>

    <link href="${contextPath}/resources/css/bootstrap.min.css" rel="stylesheet">
    <link href="${contextPath}/resources/css/common.css" rel="stylesheet">
</head>

<body>

    <div class="container">
        <form method="POST" action="${contextPath}/login" class="form-signin">
            <h2 class="form-heading">Log in</h2>

            <div class="form-group ${error != null ? 'has-error' : ''}">
                <span>${message}</span>
                <input name="username" type="text" class="form-control" placeholder="Username"
                    autofocus="true"/>
                <input name="password" type="password" class="form-control" placeholder="Password"/>
                <span>${error}</span>
                <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}"/>

                <button class="btn btn-lg btn-primary btn-block" type="submit">Log In</button>
                <h4 class="text-center"><a href="${contextPath}/registration">Create an account</a></h4>
            </div>
        </form>
    </div>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
    <script src="${contextPath}/resources/js/bootstrap.min.js"></script>
</body>
</html>

```

welcome.jsp

```

<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var="contextPath" value="${pageContext.request.contextPath}"/>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Create an account</title>
    <link href="${contextPath}/resources/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
    <div class="container">
        <c:if test="${pageContext.request.userPrincipal.name != null}">
            <form id="logoutForm" method="POST" action="${contextPath}/logout">
                <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}"/>
            </form>
            <h2>Welcome ${pageContext.request.userPrincipal.name} | <a onclick="document.forms['logoutForm'].submit()">Logout</a></h2>
        </c:if>
    </div>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
    <script src="${contextPath}/resources/js/bootstrap.min.js"></script>
</body>
</html>

```



Simplicity of a spreadsheet, flexibility of a database. Calendar, forms, & more.

Ad A beautiful spreadsheet interface coupled with the power of a database.

Airtable

[Learn More](#)

Define Properties, Configurations and Run

Properties

application.properties

```

spring.jpa.hibernate.ddl-auto=create
spring.jpa.show-sql=true

spring.mvc.view.prefix: /
spring.mvc.view.suffix: .jsp
spring.messages.basename=validation

```

validation.properties

```

NotEmpty=This field is required.
Size.userForm.username=Please use between 6 and 32 characters.

```

Duplicate.userForm.username=Someone already has that username.
 Size.userForm.password=Try one with at least 8 characters.
 Diff.userForm.passwordConfirm=These passwords don't match.

Web Security Configuration

WebSecurityConfig.java

```
package com.hellokoding.auth;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.authentication.AuthenticationManager;
import org.springframework.security.config.annotation.authentication.builders.AuthenticationBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

@Configuration
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    @Autowired
    private UserDetailsService userDetailsService;

    @Bean
    public BCryptPasswordEncoder bCryptPasswordEncoder() {
        return new BCryptPasswordEncoder();
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
            .authorizeRequests()
                .antMatchers("/resources/**", "/registration").permitAll()
                .anyRequest().authenticated()
                .and()
            .formLogin()
                .loginPage("/login")
                .permitAll()
                .and()
            .logout()
                .permitAll();
    }

    @Bean
    public AuthenticationManager customAuthenticationManager() throws Exception {
        return authenticationManager();
    }

    @Autowired
    public void configureGlobal(AuthenticationBuilder auth) throws Exception {
        auth.userDetailsService(userDetailsService).passwordEncoder(bCryptPasswordEncoder());
    }
}
```

Application Configuration

WebApplication.java

```
package com.hellokoding.auth;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

@SpringBootApplication
public class WebApplication extends SpringBootServletInitializer {
    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
        return application.sources(WebApplication.class);
    }

    public static void main(String[] args) throws Exception {
        SpringApplication.run(WebApplication.class, args);
    }
}
```

Run with Maven

You can run the application using mvn clean spring-boot:run and visit to <http://localhost:8080>

Source code

<https://github.com/hellokoding/hellokoding-courses/tree/master/springboot-examples/springboot-registration-login>

See also

[Spring Boot Course In Practice with Hands-On Experienced Tutorials and Examples Series](#)

Security # Spring Boot

Love this post?

Tweet

Share

ⓘ ×

Custom Software Experts

Database Software Programs

Reliable and Affordable since 1994.

genesisdatabases.com

OPEN

Giau Ngo

Giau Ngo is a software engineer, creator of Hello Koding. Get in touch with him on Twitter and GitHub

<https://hellokoding.com/registration-and-login-example-with-spring-security-spring-boot-spring-data-jpa-hsql-jsp/>

18/26



Custom Software Experts - Database Software Programs

Ad genesisdatabases.com

Registration + Login Tutorial and Example with Spring MVC, Spring...

hellokoding.com

Airtable, a Powerful Database

Ad Airtable

Spring Boot CRUD Example with RESTful APIs, JPA, Hibernate,...

hellokoding.com

Instant No Code API Creation - DreamFactory API Solution

Ad dreamfactory.com

JPA and Hibernate Many To Many Relationship Mapping Example with...

hellokoding.com

Single Sign Out Example with JSON Web Token (JWT), Spring Boot and...

hellokoding.com

Single Sign On (SSO) Example with JSON Web Token (JWT) and...

hellokoding.com

Comments

142 Comments [hellokoding.com](#) [Login](#) ▾ [Recommend](#) 14 [Tweet](#) [Share](#)[Sort by Best](#) ▾

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name

**Sergey Vorozhtsov** • 2 years ago

can't run this, i did git clone and trying to run

```
2017-08-23 15:56:23.232 WARN 8864 --- [main] s.c.a.AnnotationConfigApplicationContext : Exception encountered during context initialization - cancelling refresh attempt: org.springframework.beans.factory.BeanDefinitionStoreException: Failed to parse configuration class [com.hellokoding.auth.WebApplication]; nested exception is java.lang.IllegalStateException: Failed to introspect annotated methods on class org.springframework.boot.context.web.SpringBootServletInitializer
```

```
2017-08-23 15:56:23.318 ERROR 8864 --- [main] o.s.boot.SpringApplication : Application startup failed
```

7 ^ | v • Reply • Share ›

**ANJEEET SHARMA** • 2 years ago

run this program throw Exception

Field bCryptPasswordEncoder in service.UserServiceImpl required a bean of type 'org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder' that could not be found.

Action:

Consider defining a bean of type 'org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder' in your configuration.

10 ^ | v • Reply • Share ›

**Ajay** • 2 years ago

Very nice article. You have done a wonderful job creating. Very much in depth coverage. Please tell me something though. I am stuck at this point: Where is the interface SecurityService?

2 ^ | v • Reply • Share ›

**Przem Or** → Ajay • 2 years ago

Look at GitHub there is a source code. There are no Service class on the page (author forgot about it)

1 ^ | v • Reply • Share ›

**Sergei** → Przem Or • a year ago

Please, can you describe what to do, I stucked as well on that place :(

p.s. I've checked the repo, there is SecurityService or am I wrong?

1 ^ | v • Reply • Share ›

**Sergey Vladimirovich** → Sergei • a year ago

you need to create class SecurityService...

^ | v • Reply • Share ›

**Faisal Julaidan** • 2 years ago

where is the code that save the user in the session ?

2 ^ | v • Reply • Share ›

**Faisal Julaidan** → Faisal Julaidan • 2 years ago

????

^ | v • Reply • Share ›

**Thieu Anh** • 2 years ago

I know i can find the JSP file code on other site but why don't you put them here . It's very important

2 ^ | v • Reply • Share ›

**Pawan Kumar** • 3 years ago

after registration, login is not working

Caused by: org.springframework.transaction.CannotCreateTransactionException: Could not open JPA EntityManager for transaction; nested exception is javax.persistence.PersistenceException: org.hibernate.TransactionException: JDBC begin transaction failed:

3 ^ | v • Reply • Share ›

**tulasi sirithoti** • 3 months ago



how to deploy this in tomcat server.

Im getting this error.

There was an unexpected error (type=Internal Server Error, status=500).

Error resolving template "login", template might not exist or might not be accessible by any of the configured Template Resolvers

[1](#) [^](#) [v](#) [• Reply](#) [• Share](#)



Gael • 2 years ago

How Giau, How are you doing? I'm facing this issue following the tutorial.

Field userRepository in com.xxx.auth.service.UserDetailsServiceImpl required a bean of type 'com.xxx.auth.repository.UserRepository' that could not be found.

Action:

Consider defining a bean of type 'com.xxx.auth.repository.UserRepository' in your configuration.

```
2017-11-06 20:09:50.942 ERROR 86309 --- [ main] o.s.test.context.TestContextManager : Caught exception while allowing TestExecutionListener
[org.springframework.test.context.web.ServletTestExecutionListener@6156496] to prepare test instance
[com.xxx.BasecampWebApplicationTests@5875de6a]
```

Can you please help me to fix this ?

[1](#) [^](#) [v](#) [• Reply](#) [• Share](#)



Ayushman Srivastava ➔ **Gael** • 5 months ago

You have to check if all the controllers,entities and services are annotated with @Controller @Service and @Entity .

[^](#) [v](#) [• Reply](#) [• Share](#)



Thieu Anh • 2 years ago

In my project :

The codes:

```
" Object userDetailsService = SecurityContextHolder.getContext().getAuthentication().getDetails();"
```

I need to change to "SecurityContextHolder.getContext().getAuthentication().getPrincipal();"

For someone has same problem!

[1](#) [^](#) [v](#) [• Reply](#) [• Share](#)



Nicolas Costet ➔ **Thieu Anh** • 2 years ago

Thanks a lot, I've spent 1 hour to understand why it didn't work

[^](#) [v](#) [• Reply](#) [• Share](#)



fazkaRus • 2 years ago

hi, thanks for guide, i'm newbie for java, so can you explain plz this error:

Failed to

```
execute goal org.apache.maven.plugins:maven-deploy-plugin:2.8.2:deploy
(default-deploy) on project SpringAppLogin: Deployment failed: repository
element was not specified in the POM inside distributionManagement element or
in -DaltDeploymentRepository=id::layout::url parameter -> [Help 1]
```

[1](#) [^](#) [v](#) [• Reply](#) [• Share](#)



Giau Ngo Mod ➔ **fazkaRus** • 2 years ago

hi fazka, can attach ur pom.xml?

[^](#) [v](#) [• Reply](#) [• Share](#)



NecmiK • 16 days ago

Where can e find common.css file?

[^](#) [v](#) [• Reply](#) [• Share](#)



Fujie Zhang • a month ago

Nice work, Giau! I'd like to use this as a template for production. So, I switched from H2 to MySQL and Spring embedded tomcat to a standalone one. I kept everything else about the same. Everything worked fine except one thing. Once logged in and then out, if I attempt log in again, instead of getting the welcome page, I get some either css or javascript (the ones embedded in the jsp templates) displayed in the raw form(text). I spend hours trying to figure this out to no avail. Any help will be greatly appreciated!

[^](#) [v](#) [• Reply](#) [• Share](#)



Alexandra Veres • 2 months ago

I get this at runtime: java.lang.IllegalStateException: Failed to introspect Class [org.springframework.security.config.annotation.web.configuration.WebSecurityConfiguration] from ClassLoader
Can anyone help?

[^](#) [v](#) [• Reply](#) [• Share](#)

**iammiquelmx** • 2 months ago

Hey i have a dude, how configure the roles for differents users?

^ | v • Reply • Share >

**Dinesh Dhiman** • 2 months ago

How can i see the db entries in this tutorial.

^ | v • Reply • Share >

**Rob Redido** • 2 months ago

@Giau Ngo In your UserServiceImpl.java, why did you set the roles of the new User to all existing roles? Or if anyone else knows the answer, it will be appreciated.

<https://github.com/hellokod...>

^ | v • Reply • Share >

**Amir Mohammad HI** • 3 months ago

How can I handle exceptions like "account is locked" in Thymeleaf?

^ | v • Reply • Share >

**Adalid Villanueva** • 4 months ago • edited

Hello, thanks for the example. On question: How can I create a default user, for example, and admin already registered when I run the web app?

^ | v • Reply • Share >

**Edson Passos** • 4 months ago[...] getAuthentication().getDetails();
getPrincipal() should be used instead of getDetails().

^ | v • Reply • Share >

**Konstantina** • 5 months ago

Hi,

When I load any URL for the controller methods, e.g. <http://localhost:8080/registration>, I see the string 'registration' on screen, instead of the Registration view (registration.jsp).

I am using IntelliJ Community Edition 2017.3.5 and I had a notification that .jsp files are not supported. I went to Settings - File Types - HTML under Recognized File Types, and I added a new registered pattern, *.jsp, overriding the already existing registered pattern for JSP Files (syntax Highlighting Only). The message that .jsp was not supported disappeared but I still don't see the Registration page properly.

Any ideas?

^ | v • Reply • Share >

**Himanshu Goyal** ➔ Konstantina • 4 months ago

Add the dependency for JSP support.

^ | v • Reply • Share >

**Koduru Hari Krishna** • 6 months ago

where we are creating the role like admin and password

^ | v • Reply • Share >

**Koduru Hari Krishna** • 6 months ago

How we are creating the roles like admin and password

^ | v • Reply • Share >

**zeba shabnam** • 6 months ago

Error starting ApplicationContext. To display the conditions report re-run your application with 'debug' enabled.

2019-03-08 13:36:53.254 ERROR 5660 --- [main] o.s.b.d.LoggingFailureAnalysisReporter :

APPLICATION FAILED TO START

Description:

The bean 'userRepository', defined in null, could not be registered. A bean with that name has already been defined in null and overriding is disabled.

Action:

Consider renaming one of the beans or enabling overriding by setting spring.main.allow-bean-definition-overriding=true

^ | v • Reply • Share >

**samet kurumahmutoğlu** • 7 months ago

Hi when i run as executable jar file like "java -jar abc.jar" i get whitelabel error page.Any suggestion?

[^](#) [v](#) • Reply • Share > fc van den berg • 9 months ago

I am trying to understand the part where the data is saved to the user_role table. Can anyone please explain to me? thanks

[^](#) [v](#) • Reply • Share > Nilesh Patil • 9 months ago

Hi How to handle spring security together for formlogin and reservices in same war

Login form

```
http.authorizeRequests()
    .antMatchers("/", "/login").permitAll()
    .antMatchers(HttpMethod.GET, "/resources/**", "/static/**", "/css/**", "/js/**", "/vendor/**", "/images/**", "/vendor**").permitAll()
    .antMatchers("/admin/**").hasAuthority("ADMIN")
    .anyRequest()
    .authenticated()
    .and().csrf().disable().formLogin()
        .loginPage("/login")
        .loginProcessingUrl("/login")
        .failureUrl("/login?error=true")
        .defaultSuccessUrl("/admin/dashboard")
        .usernameParameter("userName").passwordParameter("password")
        .and().rememberMe().rememberMeParameter("remember-me").tokenRepository(persistentTokenRepository()).tokenValiditySeconds(3600)
        .and().logout()
        .logoutRequestMatcher(new AntPathRequestMatcher("/logout"))
```

[see more](#)[^](#) [v](#) • Reply • Share >

Nilesh Patil → Nilesh Patil • 9 months ago

Please help with inputs

[^](#) [v](#) • Reply • Share >

muhammed • a year ago

this is a structure ready for spring security.not suitable for professional applications.

[^](#) [v](#) • Reply • Share >

AkshatArora • a year ago • edited

Running the program returns

SecurityServiceImpl required a bean of type 'org.springframework.security.authentication.AuthenticationManager' that could not be found.
Action:

Consider defining a bean of type 'org.springframework.security.authentication.AuthenticationManager' in your configuration.

What could I be doing wrong.

It worked after I removed Autowiring and created a new instance, but I believe that would be incorrect

[^](#) [v](#) • Reply • Share >

Nathaniel Marzan → AkshatArora • 7 months ago

encountered the same but with version 2.1 of Spring Boot

[^](#) [v](#) • Reply • Share >

Juan Antonio González → Nathaniel Marzan • 13 days ago

You have to implement WebSecurityConfig

[^](#) [v](#) • Reply • Share >

mounika • a year ago

Actually i didnt understand What is the use of this Role.java?

[^](#) [v](#) • Reply • Share >

mounika • a year ago

i am getting this error

"Field authenticationManager in com.itcinfo.encode.service.SecurityServiceImpl required a bean of type 'org.springframework.security.authentication.AuthenticationManager' that could not be found."

[^](#) [v](#) • Reply • Share > mounika → mounika • a year ago



could someone hep me out!!

thanks in advance

[^](#) [v](#) • Reply • Share [›](#)



Juan Antonio González → mounika • 13 days ago

You have to implement WebSecurityConfig...

[^](#) [v](#) • Reply • Share [›](#)



Anu • a year ago

Hi Giau, Which server is been used here?

[^](#) [v](#) • Reply • Share [›](#)



Giau Ngo Mod → Anu • a year ago

Hi Anu, It's Tomecat server.

16 [^](#) [v](#) • Reply • Share [›](#)



Anu → Giau Ngo • a year ago

Hi Giau, Is tomcat is inbuilt in Spring boot?

[^](#) [v](#) • Reply • Share [›](#)



Xiaofeng Li → Anu • 8 months ago

no in the pom.xml, there is a buildin tomcat plugin, that is only good for development.

[^](#) [v](#) • Reply • Share [›](#)



DIMENSIONAL WANDERER • a year ago

When i implement this into my code, it gives me a null pointer exception due to a null instance of userDetailsService, i have a custom implementation for this class, but apparently spring automatically uses DaoAuthProvider and it does not have the userdetails instance so it gives a nullpointer when calling .retrieveUser(), how can i configure spring so that it takes my userDetailsService and injects it into the DaoAuthProvider?

[^](#) [v](#) • Reply • Share [›](#)



qamar • a year ago

its working awesome. great tutorial , appriciate your effort. Allah bless you

[^](#) [v](#) • Reply • Share [›](#)



Andrew Bauer • a year ago

I simply imported this into Eclipse Oxygen.3a Release (4.7.3a) and after build the Application startup fails with the error:

Error creating bean with name 'org.springframework.boot.autoconfigure.MessageSourceAutoConfiguration': Initialization of bean failed; nested exception is java.lang.NoClassDefFoundError: javax/xml/bind/ValidationException

Could you kindly update the example so that it works? ... or is it incompatible with Eclipse?

[^](#) [v](#) • Reply • Share [›](#)

[Load more comments](#)

ALSO ON HELLOKODING.COM

Database Migration/Evolution with Flyway and JPA/Hibernate

1 comment • 8 months ago



Hoang Nam — Hello,

Avatar Could you let me know how to generate sql script ? I often let Hibernate generate table schemes for me by changing entities (domain), so how do I

Handling Properties file in Spring Boot application

1 comment • 2 years ago



Valentine Golubev — Hello Giau! what if 'application.properties' fie is

Avatar outside the compiled package?

RESTful APIs Example with Spring Boot, Integration Test with ...

2 comments • a year ago

lex Cept — Where to put dislike?

Avatar

Handling Form Submission Example with Java, Spring Boot ...

2 comments • 2 years ago

asdf — Is there any example to create a custom Comment handler in spring

Avatarboot?

[✉ Subscribe](#) [>Add Disqus to your site](#) [Add Disqus Add](#) [🔒 Disqus' Privacy Policy](#) [Privacy Policy](#) [Privacy Policy](#)

Wanna get latest updates?



Custom Software Experts

Database Software Program

Reliable and Affordable since
genesisdatabases.com

OPEN

Which would you click?



*prices found on 1/21

wikibuy.



Which would you click?



*prices found on 1/21

wikibuy.

Airtable, a Powerful Database

Airtable Is The Real-time Collaborative Web-based Database Used By 80,000+ Companies.

Airtable

Ad closed by Google

[Report this ad](#)

[Why this ad? ⓘ](#)

Hello Koding - Practical Coding Courses, Tutorials and Examples

[Start here <<](#)

Data Structure

Algorithm

Java

Spring Boot

JPA and Hibernate

RESTful API

 GitHub

 Facebook

 Twitter