

Create authentication service that returns JWT

```
package com.example.jwtauth.model;
```

```
public class JwtRequest {  
    private String username;  
    private String password;  
  
    public JwtRequest() {}  
  
    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;  
    }  
}
```

```
package com.example.jwtauth.model;
```

```
public class JwtResponse {
```

```
private String token;

public JwtResponse(String token) {
    this.token = token;
}

public String getToken() {
    return token;
}

public void setToken(String token) {
    this.token = token;
}
}

package com.example.jwtauth.util;

import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;
import org.springframework.stereotype.Component;

import java.util.Date;

@Component
public class JwtUtil {
    private final String SECRET_KEY = "secret123";

    public String generateToken(String username) {
        return Jwts.builder()
```

```

        .setSubject(username)

        .setIssuedAt(new Date())

        .setExpiration(new Date(System.currentTimeMillis() + 1000 * 60 * 60)) // 1 hour

        .signWith(SignatureAlgorithm.HS256, SECRET_KEY)

        .compact();
    }
}

```

```
package com.example.jwtauth.service;
```

```

import org.springframework.security.core.userdetails.User;
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 java.util.ArrayList;
```

```
@Service
```

```
public class CustomUserDetailsService implements UserDetailsService {
```

```
    @Override
```

```
    public UserDetails loadUserByUsername(String username) throws
    UsernameNotFoundException {
```

```
        System.out.println("Attempting login for: " + username);
```

```
        if ("admin".equalsIgnoreCase(username)) {
```

```
            return new User(
```

```

        "admin",

        "$2a$10$Wz3lxqxciWBtdOMZgL8O8Es8Q5Qgzv6Fn71g2O6W6pxkq5qaZlG",
// Don't generate this in code!

        new ArrayList<>()

    );

    } else {

        throw new UsernameNotFoundException("User not found");

    }

}

}

```

```

package com.example.jwtauth.config;

```

```

import com.example.jwtauth.service.CustomUserDetailsService;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.authentication.AuthenticationManager;

import org.springframework.security.authentication.AuthenticationProvider;

import org.springframework.security.authentication.dao.DaoAuthenticationProvider;

import
org.springframework.security.config.annotation.authentication.configuration.Authentic
ationConfiguration;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import
org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.core.userdetails.UserDetailsService;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.security.crypto.password.PasswordEncoder;

import org.springframework.security.web.SecurityFilterChain;

```

@Configuration

@EnableWebSecurity

public class SecurityConfig {

 @Bean


 public UserDetailsService userDetailsService() {

 return new CustomUserDetailsService();

 }

 @Bean

 public PasswordEncoder passwordEncoder() {

 return new BCryptPasswordEncoder(); // No orange warning here 

 }

 @SuppressWarnings("deprecation")

 @Bean

 public AuthenticationProvider authenticationProvider() {

 DaoAuthenticationProvider provider = new DaoAuthenticationProvider();

 provider.setUserDetailsService(userDetailsService());

 provider.setPasswordEncoder(passwordEncoder());

 return provider;

 }

 @Bean

 public AuthenticationManager authenticationManager(AuthenticationConfiguration config) throws Exception {

 return config.getAuthenticationManager();

```
}
```

```
@Bean
```

```
public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
```

```
    return http
```

```
        .csrf(csrf -> csrf.disable())
```

```
        .authorizeHttpRequests(auth -> auth
```

```
            .requestMatchers("/authenticate").permitAll()
```

```
            .anyRequest().authenticated()
```

```
        )
```

```
        .build();
```

```
    }
```

```
}
```

```
package com.example.jwtauth.controller;
```

```
import com.example.jwtauth.model.JwtRequest;
```

```
import com.example.jwtauth.model.JwtResponse;
```

```
import com.example.jwtauth.util.JwtUtil;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.http.ResponseEntity;
```

```
import org.springframework.security.authentication.AuthenticationManager;
```

```
import org.springframework.security.authentication.BadCredentialsException;
```

```
import
```

```
org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
```

```
import org.springframework.web.bind.annotation.*;
```

```
@RestController
```

```
public class AuthController {
```

@Autowired

private AuthenticationManager authenticationManager;

@Autowired

private JwtUtil jwtUtil;

@PostMapping("/authenticate")

public ResponseEntity<?> createToken(@RequestBody JwtRequest request) {

try {

authenticationManager.authenticate(

new UsernamePasswordAuthenticationToken(

request.getUsername(), request.getPassword())

);

} catch (BadCredentialsException e) {

return ResponseEntity.status(401).body("Invalid Credentials");

}

final String token = jwtUtil.generateToken(request.getUsername());

return ResponseEntity.ok(new JwtResponse(token));

}

}

OUTPUT:

eclipse-workspace - jwtauth/src/main/java/com/example/jwtauth/JwtAuthApplication.java - Eclipse IDE

File Edit Source Refactor Source Navigate Search Project Run Window Help

Project Explorer

- 01_Intro
- 03_HCQL
- 08_SpringMVCDemo
- 11_SpringBootLoginDemo
- ControlStructureProject
- countryconfig [boot]
- countryservice [boot]
- course-registration-service
- course-service
- demo
- dependency-injection-demo
- e-commerce (in spring boot api)
- ECommerceSearchApp
- eureka-server
- exp-6
- Experiments
- FactoryPatternDemo
- FinancialForecast
- Hackathon [boot]
- handloom-app
- handmadeshop-master
- HCQL
- helloworld [boot]
- jsdscaptcha (in jsdcrud)
- junltdemo
- jwtauth [boot]
 - src/main/java
 - com.example.jwtauth
 - com.example.jwtauth.config
 - com.example.jwtauth.controller
 - com.example.jwtauth.model
 - com.example.jwtauth.service
 - com.example.jwtauth.util

JwtRequest.java JwtResponse.java JwtUtil.java CustomUserDetails... SecurityConfig.java AuthController.java application.prop... JwtAuthApplication.java

```
1 package com.example.jwtauth;
2
3 import org.springframework.boot.SpringApplication;
4
5
6 @SpringBootApplication
7 public class JwtAuthApplication {
8     public static void main(String[] args) {
9         SpringApplication.run(JwtAuthApplication.class, args);
10    }
11 }
12
```

Problems Servers Terminal Data Source Explorer Properties Console Outline JUnit

JwtAuthApplication (2) [Java Application] C:\Users\BORRA Panitha\p2\pool\plugins\org.eclipse.justopenjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (13 Jul 2025, 7:29:05 pm) [pid: 27668]

```
2025-07-13T19:29:11.593+05:30 INFO 27668 --- [jwtauth] [main] o.a.c.c.C.[Tomcat].[...]
2025-07-13T19:29:11.597+05:30 INFO 27668 --- [jwtauth] [main] w.s.c.ServletWebServer
2025-07-13T19:29:11.923+05:30 INFO 27668 --- [jwtauth] [main] eAuthenticationProvider
2025-07-13T19:29:11.924+05:30 WARN 27668 --- [jwtauth] [main] r$InitializeUserData
2025-07-13T19:29:12.976+05:30 INFO 27668 --- [jwtauth] [main] o.s.b.w.embedded.tomcat
2025-07-13T19:29:13.004+05:30 INFO 27668 --- [jwtauth] [main] com.example.jwtauth..
2025-07-13T19:29:26.026+05:30 INFO 27668 --- [jwtauth] [nio-8090-exec-1] o.a.c.c.C.[Tomcat].[...]
2025-07-13T19:29:26.027+05:30 INFO 27668 --- [jwtauth] [nio-8090-exec-1] o.s.web.servlet.Dispatcher
2025-07-13T19:29:26.029+05:30 INFO 27668 --- [jwtauth] [nio-8090-exec-1] o.s.web.servlet.Dispatcher
Attempting login for: admin
```