

**pom.xml:**

```
<!-- Spring Boot Web -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<!-- Spring Boot Security -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>

<!-- JSON Web Token -->
<dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt</artifactId>
    <version>0.9.0</version>
</dependency>
```

**SpringlearnApplication.java:**

```
package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringlearnApplication {

    public static void main(String[] args) {

        SpringApplication.run(SpringlearnApplication.class, args);

    }

}
```

**AuthenticationController.java:**

```
package com.cognizant.springlearn.controller;

import java.util.Base64;
```

```

import java.util.Date;

import java.util.HashMap;

import java.util.Map;


import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestHeader;

import org.springframework.web.bind.annotation.RestController;


import io.jsonwebtoken.JwtBuilder;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;


@RestController

public class AuthenticationController {


    private static final Logger LOGGER = LoggerFactory.getLogger(AuthenticationController.class);


    @GetMapping("/authenticate")

    public Map<String, String> authenticate(@RequestHeader("Authorization") String authHeader) {

        LOGGER.info("Start");

        LOGGER.debug("Authorization Header: {}", authHeader);

        String user = getUser(authHeader);

        String token = generateJwt(user);

        Map<String, String> map = new HashMap<>();

        map.put("token", token);

        LOGGER.info("End");

        return map;

    }

    private String getUser(String authHeader) {

```

```

    LOGGER.info("Decoding Authorization Header");

    String encodedCredentials = authHeader.substring("Basic ".length());

    byte[] decodedBytes = Base64.getDecoder().decode(encodedCredentials);

    String decodedString = new String(decodedBytes);

    String user = decodedString.split(":")[0];

    LOGGER.debug("User: {}", user);

    return user;
}

private String generateJwt(String user) {

    LOGGER.info("Generating JWT token");

    JwtBuilder builder = Jwts.builder();

    builder.setSubject(user);

    builder.setIssuedAt(new Date());

    builder.setExpiration(new Date(System.currentTimeMillis() + 20 * 60 * 1000)); // 20 mins

    builder.signWith(SignatureAlgorithm.HS256, "secretkey");

    String token = builder.compact();

    LOGGER.debug("Token: {}", token);

    return token;
}
}

```

#### **SecurityConfig.java:**

```

package com.cognizant.springlearn.security;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import
org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuild
er;

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

@Configuration
@EnableWebSecurity

public class SecurityConfig extends WebSecurityConfigurerAdapter {

    private static final Logger LOGGER = LoggerFactory.getLogger(SecurityConfig.class);

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
        auth.inMemoryAuthentication()

            .withUser("admin").password(passwordEncoder().encode("pwd")).roles("ADMIN")

            .and()

            .withUser("user").password(passwordEncoder().encode("pwd")).roles("USER");
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.csrf().disable().httpBasic().and()

            .authorizeRequests()

            .antMatchers("/authenticate").hasAnyRole("USER", "ADMIN")

            .anyRequest().authenticated();
    }

    @Bean
    public PasswordEncoder passwordEncoder() {
        LOGGER.info("Password encoder created");
        return new BCryptPasswordEncoder();
    }
}

```

**OUTPUT:**

```
250714|00:24:25.688|restartedMain|INFO|DetailsManagerConfigurer|configure|Global AuthenticationManager configured with UserDetailsService bean with name inMemoryUserDetailsService
250714|00:24:25.655|restartedMain|INFO|OptionalLiveReloadServer|startServer|LiveReload server is running on port 35729
250714|00:24:25.155|restartedMain|INFO|s.b.w.e.t.TomcatWebServer|start|Tomcat started on port 8083 (http) with context path '/'
250714|00:24:25.181|restartedMain|INFO|.s.SpringLearnApplication|logStarted|Started SpringLearnApplication in 10.571 seconds (process running for 12.0)
250714|00:24:25.199|restartedMain|INFO|.s.SpringLearnApplication|main|START - main
250714|00:24:25.200|restartedMain|INFO|.s.SpringLearnApplication|displayDate|START - displayDate
250714|00:24:25.492|restartedMain|DEBUG|.s.SpringLearnApplication|displayDate|Parsed Date: Mon Dec 31 00:00:00 2018
250714|00:24:25.496|restartedMain|INFO|.s.SpringLearnApplication|displayDate|END - displayDate
250714|00:24:25.497|restartedMain|INFO|.s.SpringLearnApplication|displayCountry|START - displayCountry
250714|00:24:25.575|restartedMain|DEBUG|.c.c.spring_learn.Country|<init>|Inside Country Constructor
250714|00:24:25.577|restartedMain|DEBUG|.c.c.spring_learn.Country|setCode|Setting code
250714|00:24:25.578|restartedMain|DEBUG|.c.c.spring_learn.Country|setName|Setting name
250714|00:24:25.580|restartedMain|DEBUG|.c.c.spring_learn.Country|<init>|Inside Country Constructor
250714|00:24:25.580|restartedMain|DEBUG|.c.c.spring_learn.Country|setCode|Setting code
250714|00:24:25.580|restartedMain|DEBUG|.c.c.spring_learn.Country|setName|Setting name
250714|00:24:25.580|restartedMain|DEBUG|.c.c.spring_learn.Country|<init>|Inside Country Constructor
250714|00:24:25.581|restartedMain|DEBUG|.c.c.spring_learn.Country|setCode|Setting code
250714|00:24:25.582|restartedMain|DEBUG|.c.c.spring_learn.Country|setName|Setting name
250714|00:24:25.583|restartedMain|DEBUG|.c.c.spring_learn.Country|<init>|Inside Country Constructor
250714|00:24:25.585|restartedMain|DEBUG|.c.c.spring_learn.Country|setCode|Setting code
250714|00:24:25.585|restartedMain|DEBUG|.c.c.spring_learn.Country|setName|Setting name
250714|00:24:25.585|restartedMain|DEBUG|.c.c.spring_learn.Country|<init>|Inside Country Constructor
250714|00:24:25.586|restartedMain|DEBUG|.c.c.spring_learn.Country|setCode|Setting code
250714|00:24:25.587|restartedMain|DEBUG|.c.c.spring_learn.Country|setName|Setting name
250714|00:24:25.597|restartedMain|DEBUG|.s.SpringLearnApplication|displayCountry|Country [code=IN, name=India]
250714|00:24:25.597|restartedMain|INFO|.s.SpringLearnApplication|displayCountry|END - displayCountry
250714|00:24:25.597|restartedMain|INFO|.s.SpringLearnApplication|main|END - main
```