

# Create authentication service that returns JWT

## Code:

### JwtAuthServiceApplication.java:

```
package com.example.jwtauth;

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

@SpringBootApplication

public class JwtAuthServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(JwtAuthServiceApplication.class, args);

    }

}
```

### AuthController.java:

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

import com.example.jwtauth.model.AuthRequest;
import com.example.jwtauth.model.AuthResponse;
import com.example.jwtauth.util.JwtUtil;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

@RestController

@RequestMapping("/api")

public class AuthController {

    @Autowired

    private JwtUtil jwtUtil;

    @PostMapping("/authenticate")

    public AuthResponse authenticate(@RequestBody AuthRequest request) {

        if ("user".equals(request.getUsername()) && "password".equals(request.getPassword())) {

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

            return new AuthResponse(token);

        } else {

            throw new RuntimeException("Invalid Credentials");

        }

    }

}
```

```
}  
}
```

#### **AuthRequest.java:**

```
package com.example.jwtauth.model;  
  
public class AuthRequest {  
    private String username;  
    private String password;  
    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;  
    }  
}
```

#### **AuthResponse.java:**

```
package com.example.jwtauth.model;  
  
public class AuthResponse {  
    private String token;  
    public AuthResponse(String token) {  
        this.token = token;  
    }  
    public String getToken() {  
        return token;  
    }  
}
```

**SecurityConfig.java:**

```
package com.example.jwtauth.security;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
public class SecurityConfig {

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http.csrf(csrf -> csrf.disable())
            .authorizeHttpRequests(auth -> auth.anyRequest().permitAll());
        return http.build();
    }
}
```

**JwtUtil.java:**

```
package com.example.jwtauth.util;

import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;
import io.jsonwebtoken.security.Keys;
import java.security.Key;
import java.util.Date;
import org.springframework.stereotype.Component;

@Component
public class JwtUtil {

    private final Key key = Keys.secretKeyFor(SignatureAlgorithm.HS256);

    public String generateToken(String username) {
        return Jwts.builder()
            .setSubject(username)
            .setIssuedAt(new Date())
            .setExpiration(new Date(System.currentTimeMillis() + 1000 * 60 * 30)) // 30 min
            .signWith(key)
            .compact();
    }
}
```

```
}  
}
```

**pom.xml:**

```
<project xmlns="http://maven.apache.org/POM/4.0.0"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
        http://maven.apache.org/xsd/maven-4.0.0.xsd">  
    <modelVersion>4.0.0</modelVersion>  
    <groupId>com.example</groupId>  
    <artifactId>jwt-auth-service</artifactId>  
    <version>1.0.0</version>  
    <packaging>jar</packaging>  
    <parent>  
        <groupId>org.springframework.boot</groupId>  
        <artifactId>spring-boot-starter-parent</artifactId>  
        <version>3.2.5</version>  
        <relativePath/>  
    </parent>  
    <properties>  
        <java.version>17</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-security</artifactId>  
        </dependency>  
        <dependency>  
            <groupId>io.jsonwebtoken</groupId>  
            <artifactId>jjwt-api</artifactId>  
            <version>0.11.5</version>
```

```

</dependency>
<dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-impl</artifactId>
    <version>0.11.5</version>
    <scope>runtime</scope>
</dependency>
<dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-jackson</artifactId>
    <version>0.11.5</version>
    <scope>runtime</scope>
</dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>

```

## Output:

The screenshot displays a REST client interface with two requests. The first request is a POST to `http://localhost:8080/authenticate` with a JSON body containing username and password. The response is a 200 OK status with a token. The second request is a GET to `http://localhost:8080/welcome`, which also returns a 200 OK status with an authorization header.

Request	Method	URL	Status	Time	Size
1	POST	http://localhost:8080/authenticate	200 OK	64 ms	198 B
2	GET	http://localhost:8080/welcome	200 OK	15 ms	16 B

**Request 1 Body:**

```

{
  "username": "user",
  "password": "password"
}

```

**Response 1 Body:**

```

{
  "token": "eyJwcm9uZGGOIiVZl1/NTEJv8.eyJebWU101Cs2VIwa1aAFOlJ1xI7O9MDds7CLJBGC0r3Ev3m0MP9ME0709x6-8ix54z9Qe30SJP1U"
}

```

**Response 2 Header:**

```

Authorization: "Bearer eyJwcm9uZGGOIiVZl1/NTEJv8.eyJebWU101Cs2VIwa1aAFOlJ1xI7O9MDds7CLJBGC0r3Ev3m0MP9ME0709x6-8ix54z9Qe30SJP1U"

```