

## **Create authentication service that returns JWT**

### **AuthenticationController:**

```
package com.cognizant.spring_learn.controller;

import jakarta.servlet.http.HttpServletRequest;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;
import java.util.Base64;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;

@RestController

public class AuthenticationController {

    private static final String SECRET_KEY = "mysecretkey";

    @GetMapping("/authenticate")
    public Map<String, String> authenticate(HttpServletRequest request) {
        String authHeader = request.getHeader("Authorization");
        String base64Credentials = authHeader.substring("Basic".length()).trim();
        byte[] credDecoded = Base64.getDecoder().decode(base64Credentials);
        String credentials = new String(credDecoded);
        String[] values = credentials.split(":", 2);
        String username = values[0];
        String password = values[1];
        String token = Jwts.builder()
            .setSubject(username)
            .setIssuedAt(new Date())
```

```

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

        .signWith(SignatureAlgorithm.HS256, SECRET_KEY)

        .compact();

    Map<String, String> response = new HashMap<>();
    response.put("token", token);
    return response;
}
}

```

### **SecurityConfig:**

```

package com.cognizant.spring_learn.config;

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().disable()

            .authorizeRequests()

                .antMatchers("/authenticate").permitAll()

                .anyRequest().authenticated()

            .and()

                .httpBasic();

        return http.build();
    }
}

```

**Output:**

