# Security Configuration

The provided SecurityConfig class demonstrates how to configure security for a Spring Boot application using a custom implementation of the SecurityFilterChain. This approach allows for fine-grained control over HTTP security configurations.

## SecurityConfig.java

```java
package com.spring.security.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.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity
public class SecurityConfig {

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {

        return http.build();
    }
}
```

**Note:** After creating an object of SecurityFilterChain on your own. And writing my own logic. I found that they do not ask for authentication.