Spring Boot Security

Consider we have a normal spring boot application with just one REST API and added a **spring boot starter-security** dependency and trying to hit the API.

POM File

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

</dependencies>

Rest Controller

package com.skg.springbootjwt.controller;

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

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

@RestController

public class HomeController {

@GetMapping("/")

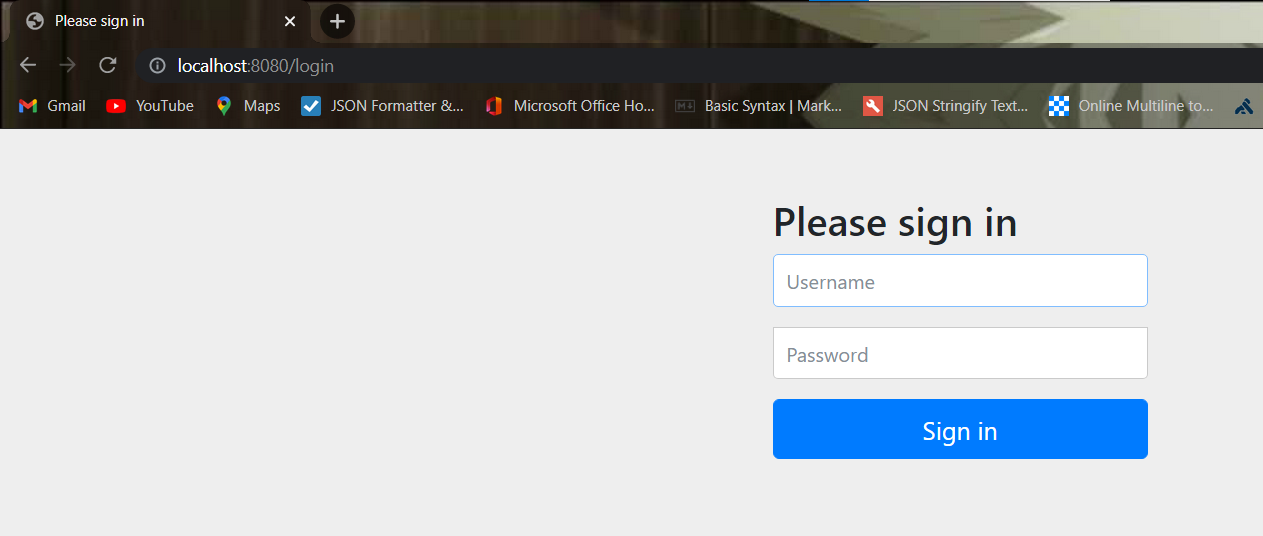
public String home() {

return "Hello, JWT !!";

}

}

Start the application and hit the API.



For example, if we try to access any API it redirects us to the default login page, the default user name is the **user** and the **password** will be printed on the console logs.

How does spring security can do this, how do all the requests get intercepted and take control over the whole application? Well, the answer is using something called **filters**.

Filters are one of the core concepts of servlet technology. **How does it work?**

Consider a web app consisting of servlets, whose work is to help the user to map it to the particular servlet mapping and does the functionality, and provides the response for that particular request.



Spring security adds the filter stand right in the middle and they intercept every request and this gives the opportunity to do something with every request.

Spring Security default behavior after adding Spring Boot Starter Security to the classpath

1. Adds mandatory authentication for URLs
2. Adds login form
3. Handles login error
4. Creates a user and sets a default password

How to say spring to use user-provided user name and password as a default value. It’s by mentioning the properties in the **application.properties**

***spring.security.user.name=foo***

***spring. security.user.password=fod***