OAuth2 Authentication with Spring and Github

Last Updated : 23 Jul, 2025

OAuth2 is a secure way to allow users to log in to your application using third-party providers like GitHub. Instead of creating a custom login system, we can use GitHub's authentication to verify users and grant access.

Working of OAuth2

Imagine a user wants to log in to your app using GitHub. When they click "Login with GitHub," they are redirected to GitHub's login page. After they sign in, GitHub sends an authorization code to your app, which exchanges it for an access token. This token allows your app to access basic user details securely.

Setting Up OAuth2 with Spring Boot

1. Create a Spring Boot Project

Start a new Spring Boot project with these dependencies:

- Spring Web
- Spring Security
- OAuth2 Client

2. Register an OAuth App on GitHub

- Go to GitHub Developer Settings and create a new OAuth App.
- Fill in the required details:
  - **Homepage URL**: http://localhost:8080

  - **Authorization Callback URL**: http://localhost:8080/login/oauth2/code/github
- Save the app and copy the Client ID and Client Secret.

3. Configure application.yml

Add your credentials in **application.yml**:

```
spring:
  security:
    oauth2:
      client:
        registration:
          github:
            client-id: your-client-id
            client-secret: your-client-secret
            scope: user:email
```

4. Security Configuration

Create **SecurityConfig.java** to enable OAuth2 login:

```java
@Configuration
@EnableWebSecurity
public class SecurityConfig {
    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws
Exception {
        http
            .authorizeHttpRequests(auth -> auth
                .requestMatchers("/").permitAll()
                .anyRequest().authenticated()
            )
            .oauth2Login();
        return http.build();
    }
}
```

5. Main Application Class

```java
@SpringBootApplication
@RestController
public class DemoApplication {

    @GetMapping("/")
    public String home(OAuth2AuthenticationToken token) {
        if (token == null) {
            return "Please log in with GitHub.";
        }
        return "Hello, " + token.getPrincipal().getAttribute("name");
    }

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

6. Run and Test

- Start your Spring Boot application.
- Open http://localhost:8080.
- Click "Login with GitHub," sign in, and see your GitHub username displayed.

This article simplifies OAuth2 login with GitHub in Spring Boot. It makes authentication easier and more secure without needing a custom login system.

Comment More info Advertise with us