



# OAuth 2 & Login

## What is it?

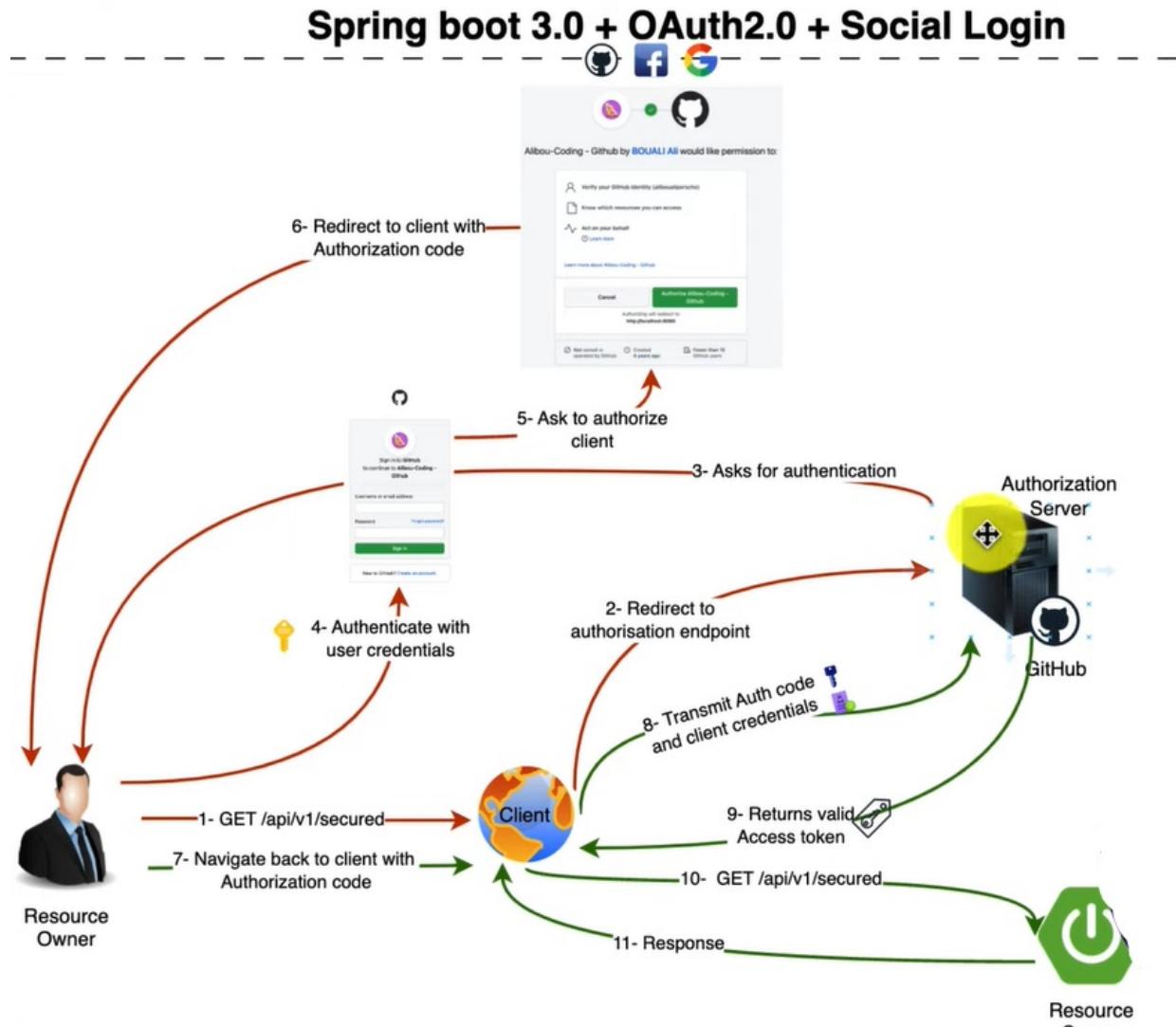
- First of all this is authorization API not a authentication.
- It provide feature the user can access the resource without any credentials.
- Instead it uses third part application to grant permission to access the resources from server

## How it works:

It basically uses 4 terms to authorize the user to access the resource.

- Resource owner (Mostly us).
- Client ( like browser, application, desktop application or any other).
- Authorization Server ( Third-Party app server will authenticate the user credentials related to this app ).
- Resource Server ( This is end point owner want to connect and get response(resource) from it). [It could be local machine or remote server].

This simple diagram serves as a comprehensive explanation, covering all the above points.



This is basic Security configuration code for authenticate all the requests that send by the client

```
@Configuration
@EnableWebSecurity
public class SecurityConfig {
```

```

@Bean//instead of the websecurity configurer
public SecurityFilterChain securityFilterChain(HttpSecurity http ) throws Exception{
    http
        .csrf().disable()
        .authorizeHttpRequests()
        .anyRequest()
        .authenticated()
        .and()
        .oauth2Login();

    return http.build();
}
}

```

## Key Difference between authorize and authenticate

1. Authentication focuses on verifying identity, while authorization determines what actions a user can perform.
2. Authentication is the initial step before authorization. Without authentication, authorization cannot take place.

## Obtain OAuth 2.0 application

### GitHub:

GitHub: Visit GitHub Developer Settings (<https://github.com/settings/developers>) → create a new OAuth application, and provide the required details → Receive your Client ID and Client Secret.

### Google:

Google: Go to the Google Cloud Console (<https://console.cloud.google.com>) → create a new project if needed, access "APIs & Services" → "Credentials", and generate an OAuth client ID → Retrieve your Client ID and Client Secret.

## Setting up

In application.yml file can easily configure this OAuth 2.0,

```
Spring:  
  security:  
    oauth2:  
      client:  
        registration:  
          github:  
            clientid: #from github developer setting  
            clientsecert: #from github developer setting  
          google:  
            client-id: #from google API services  
            client-secret: #from google API services
```

## Implementation

simple oAuth 2 login system using **GitHub** and **Google** source code available here.

<https://github.com/Karthikn-n/OAuth-2.0.git>

Here is the tutorial for this ,

[Click here.](#)