

How to login by Facebook

Simple guide on how to login by Facebook
using scibejava library



General Information

Version	1.0.0
Date	10.08.2016
Author	Phat H. Vu
Classification	Internal / Confidential / Public / Not classified
Status	On work / Draft / Released

Version control

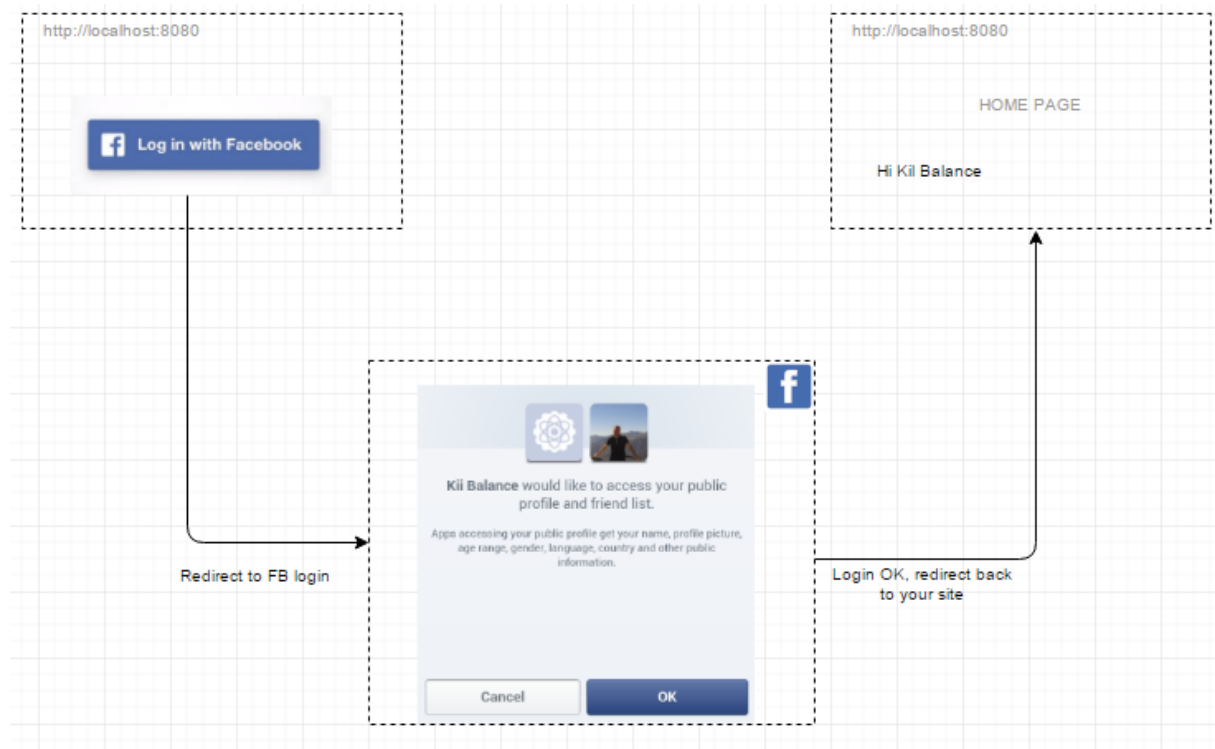
Date	Version	Author	Description
10.08.16	1.0.0	Phat H. Vu	Documentation initialization

Contents

1	Objectives	4
2	Create a Facebook app	4
3	Create a Spring MVC sample	7
4	Run the Spring MVC sample	10

1 Objectives

This guide will help you to demonstrate how to integrate Facebook login into a Spring Web MVC application. We use a framework named **scribejava** (<https://github.com/scribejava/scribejava>) to work with FB resources.



Overview flow when logging with Facebook

Basically, following are main steps:

- Create a Facebook app
 - Get App Id & App secret.
 - Specify call back URI. You can imagine once successfully authenticated from Facebook, which URI that FB will redirect.
- Use **scribejava** library to get access token, then use this access token to get Facebook profile.

2 Create a Facebook app

Access to: <https://developers.facebook.com/apps/>, click on button **Add a new App**. Fill required data:

Create a New App ID

Get started integrating Facebook into your app or website

Display Name

andy.vn

Contact Email

vuhongphat@gmail.com

By proceeding, you agree to the [Facebook Platform Policies](#)

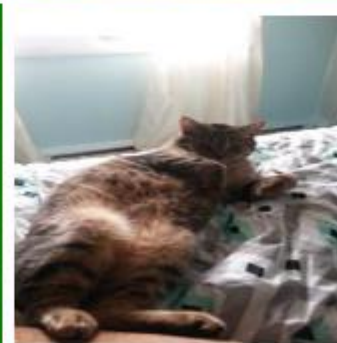
Cancel

Create App ID

Security Check

Please select all the photos which show a flower.

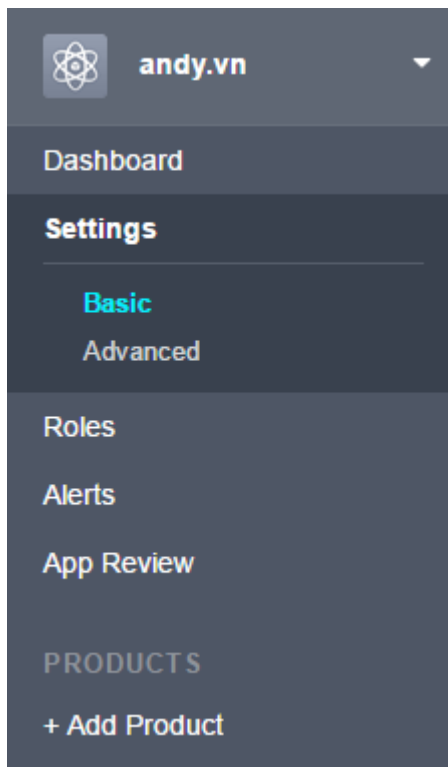
You can also do this security check with text or audio instead of photos.



If you think you're seeing this by mistake, please let us know.

Submit

Cancel



Once successfully creating an Facebook app, collect App ID and App Secret:

App ID	App Secret
657595544391983 Show

Must specify Valid OAuth redirect URIs. The URI is a callback when the login successfully. For example, you can specify <http://localhost:8080>. In production environment, you can replace it by a domain name:

Client OAuth Settings

☒ Yes

Client OAuth Login
Enables the standard OAuth client token flow. Secure your application and prevent abuse by locking down which token redirect URIs are allowed with the options below. Disable globally if not used. [?]

☒ Yes

Web OAuth Login
Enables web based OAuth client login for building custom login flows. [?]

☐ No

Force Web OAuth Reauthentication
When on, prompts people to enter their Facebook password in order to log in on the web. [?]

☐ No

Embedded Browser OAuth Login
Enables browser control redirect uri for OAuth client login. [?]

Valid OAuth redirect URIs

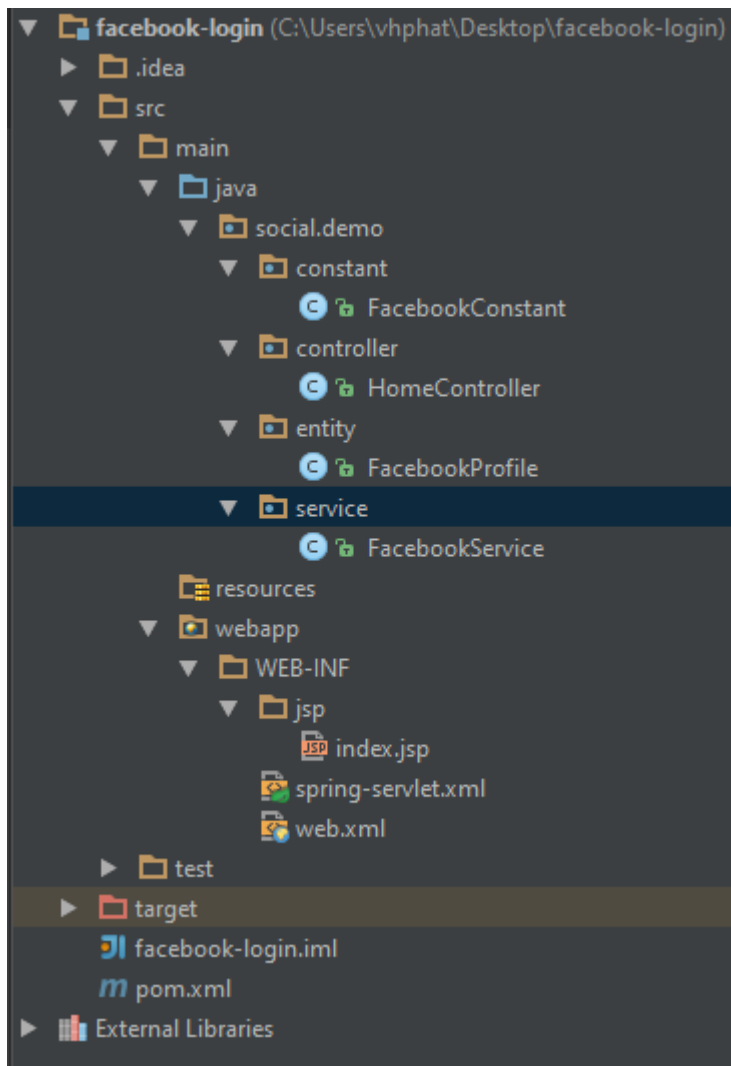
http://localhost:8080/ x

☐ No

Login from Devices
Enables the OAuth client login flow for devices like a smart TV [?]

3 Create a Spring MVC sample

Project structure:



pom.xml:

```
<properties>
  <jdkVersion>1.8</jdkVersion>
  <jackson.version>2.5.0</jackson.version>
</properties>

<dependencies>
  <dependency>
    <groupId>com.github.scribejava</groupId>
    <artifactId>scribejava-apis</artifactId>
    <version>3.0.0</version>
  </dependency>

  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>4.3.0.RELEASE</version>
  </dependency>

  <!-- Jackson & JSON dependencies -->
  <dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-annotations</artifactId>
    <version>${jackson.version}</version>
  </dependency>
</dependencies>
```

```

        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-core</artifactId>
        <version>${jackson.version}</version>
    </dependency>
    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-databind</artifactId>
        <version>${jackson.version}</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/jstl/jstl -->
    <dependency>
        <groupId>jstl</groupId>
        <artifactId>jstl</artifactId>
        <version>1.2</version>
    </dependency>
</dependencies>

```

HomeController.java:

```

@Controller
@RequestMapping(value = "/")
public class HomeController {
    @Autowired
    FacebookService facebookService;

    @RequestMapping(method = RequestMethod.GET)
    public String showBooks(Model model) {
        model.addAttribute("fb_auth_url", facebookService.getAuthUrl());
        return "index";
    }

    @RequestMapping(value = "login", method = RequestMethod.GET)
    public String facebookLogin(@RequestParam(value = "code") String code, Model
model) throws IOException {
        String name = facebookService.getProfileName(code);
        model.addAttribute("profile_name", name);

        return "index";
    }
}

```

FacebookConstant.java:

```

public class FacebookConstant {
    public static final String CLIENT_ID = "657595544391983";
    public static final String CLIENT_SECRET = "fa4737e14b08fe2b7b9982cef39afa96";
    public static final String PROTECTED_RESOURCE_URL =
"https://graph.facebook.com/v2.6/me";
    public static final String CALLBACK_URL = "http://localhost:8080/login";
}

```

FacebookProfile.java:

```

public class FacebookProfile {
    private String id;
    private String name;
    // getters & setters
}

```

FacebookService.java:

```

@Service
public class FacebookService {
    OAuth20Service service;

    @PostConstruct

```



```

public void init() {
    service = new ServiceBuilder()
        .apiKey(FacebookConstant.CLIENT_ID)
        .apiSecret(FacebookConstant.CLIENT_SECRET)
        .callback(FacebookConstant.CALLBACK_URL)
        .build(FacebookApi.instance());
}

public String getAuthUrl() {
    return service.getAuthorizationUrl();
}

public String getProfileName(String authCode) throws IOException {
    OAuth2AccessToken accessToken = service.getAccessToken(authCode);
    OAuthRequest request = new OAuthRequest(Verb.GET,
FacebookConstant.PROTECTED_RESOURCE_URL, service);
    service.signRequest(accessToken, request);
    Response response = request.send();

    ObjectMapper mapper = new ObjectMapper();
    FacebookProfile facebookProfile = mapper.readValue(response.getBody(),
FacebookProfile.class);

    return facebookProfile.getName();
}
}

```

web.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">

    <display-name>Demonstration how to Login by Facebook</display-name>

    <servlet>
        <servlet-name>spring</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>spring</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

</web-app>

```

spring-servlet.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.2.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.2.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd">

    <context:component-scan base-package="social.demo"/>

    <!-- Resolves views selected for rendering by @Controllers to .jsp resources
in the /WEB-INF/jsp directory -->

```

```

        <bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/jsp/" />
        <property name="suffix" value=".jsp" />
    </bean>

    <mvc:default-servlet-handler />

    <!-- Enables the Spring MVC @Controller programming model -->
    <mvc:annotation-driven />
</beans>

```

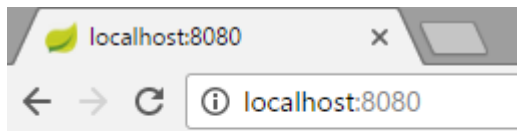
index.jsp :

```

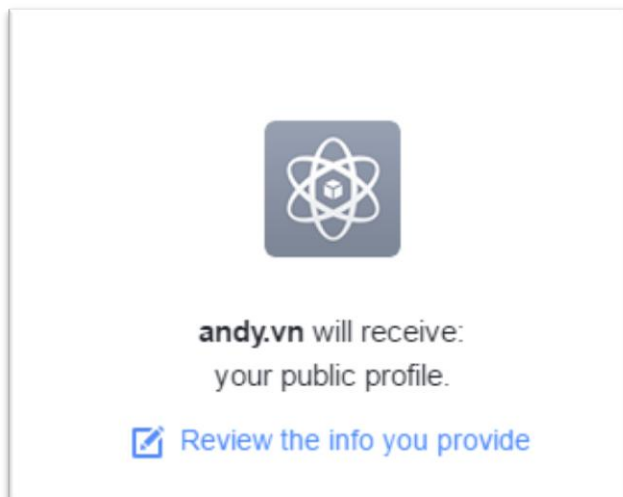
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <title></title>
</head>
<body>
Home page <br>
<c:choose>
    <c:when test="${profile_name != null}">
        Hi ${profile_name}
    </c:when>
    <c:otherwise>
        <a href="${fb_auth_url}" target="_blank">Login by Facebook</a>
    </c:otherwise>
</c:choose>
</body>
</html>

```

4 Run the Spring MVC sample



Home page
[Login by Facebook](#)





andy.vn will receive:
your public profile.

☒ [Review the info you provide](#)

Continue as Vũ

Cancel

This does not let the app post to Facebook

