Facebook BlackBerry SDK

Last Updated: 2010-09-17

Introduction

This SDK is a powerful and convenient Java library for BlackBerry developers who want to integrate facebook functionalities into their BlackBerry applications using the facebook Graph API (formerly called "facebook connect"). It is an open-source project under MIT license, and it is completely free to use and/or distribute.

Features

- Connect to facebook using the facebook Graph API.
- Compatible with facebook OAuth 2.0.
- Simple and intuitive to use.
- Sample BlackBerry application is provided.

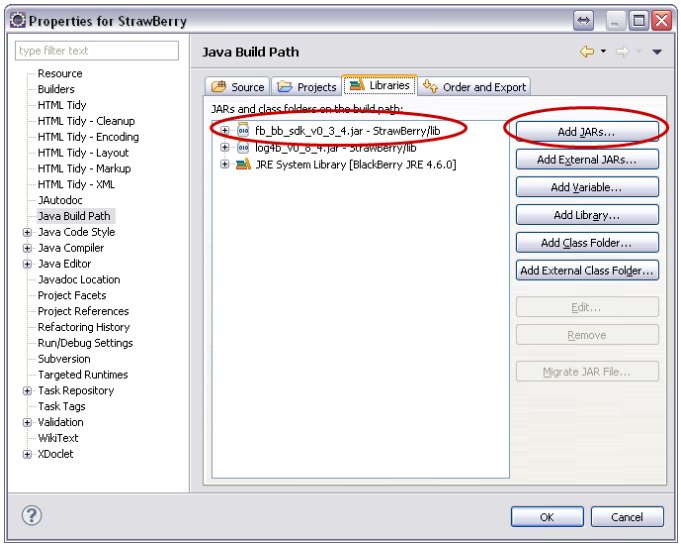
Getting Started

1. Setting up the environment

First of all, download the SDK binary files:

```
fb_bb_sdk_v0_3_4.jar
fb_bb_sdk_v0_3_4.cod
```

Put the JAR file somewhere in your BlackBerry project (e.g. /lib), and include it in the build path of your project like this:



Please note that when you are deploying your application, you have to include the SDK's COD file into the JAD file, together with your application's COD file(s). For example:

```
Manifest-Version: 1.0
RIM-COD-Module-Name: MyApp
RIM-COD-Module-Dependencies: net_rim_cldc,fb_bb_sdk_v0_3_4,net_rim_os
MIDlet-Jar-Size: 56428
MIDlet-1: ,MyApp.png,
RIM-COD-Creation-Time: 1284630427
MIDlet-Jar-URL: MyApp.jar
RIM-COD-URL: fb_bb_sdk_v0_3_4.cod
RIM-COD-Size: 42828
RIM-COD-URL-1: MyApp.cod
RIM-COD-SHA1-1: f9 d6 77 31 f6 1a 71 24 66 7e d8 cd d1 dd fd 0d be 00 8c 8f
RIM-COD-Size-1: 28260
MicroEdition-Configuration: CLDC-1.1
MIDlet-Version: 1.0.0
MIDlet-Name: MyApp
MIDlet-Vendor: <unknown>
MicroEdition-Profile: MIDP-2.0
RIM-MIDlet-Flags-1: 0
```

Secondly, make sure that you have a "facebook application" already set up on facebook (http://developers.facebook.com/setup/). Next, you have to create the FacebookContext object with all the facebook credentials like this:

```
private final String REST_URL = "http://api.facebook.com/restserver.php";
private final String GRAPH_URL = "https://graph.facebook.com";
private final String NEXT_URL = "http://www.facebook.com/connect/login_success.html";
private final String APPLICATION_KEY = "f21032d377681e02051e639830b4b678";
private final String APPLICATION_SECRET = "590906fcfea8e348589cf43f06192c2e";
private final String APPLICATION_ID = "317175255300";
ApplicationSettings appSettings = new ApplicationSettings(REST_URL, GRAPH_URL, NEXT_URL,
APPLICATION_KEY, APPLICATION_SECRET, APPLICATION_ID);
```

```
HttpConnectionFactory connFactory = new HttpConnectionFactory();

FacebookContext fbc = new FacebookContext(appSettings, connFactory);
```

2. Prompting the user to login (if not yet)

There is a handy class called LoginScreen, which can be used to handle this task nicely (See example below), but of course, feel free to implement your own custom login screen if you wish.

Create a CookieManager object to manage the cookies during the login process:

```
private CookieManager cookieManager = new CookieManager();
```

Create a LoginScreen object and register the ActionListener:

```
loginScreen = new LoginScreen(fbc, cookieManager);
loginScreen.addActionListener(this);
```

Display the LoginScreen:

```
loginScreen.login();
pushScreen(loginScreen);
```

3. Retrieving the facebook access token.

After the user logged in, the ActionListener will be called back with an "ACTION_LOGGED_IN" event. Now you have to handle this event by getting the session key and exchanging it for an "access_token" from facebook like this:

```
public void onAction(Action event) {
		if (event.getAction().equals(LoginScreen.ACTION_LOGGED_IN)) {
			try {
				popScreen(loginScreen);
			} catch (IllegalArgumentException e) {
			}

			try {
				fbc.getSession((String) event.getData());
				fbc.upgradeSession();

			} catch (Throwable t) {
				t.printStackTrace();
				Dialog.alert("Error: " + t.getMessage());
			}

		} else if (event.getAction().equals(LoginScreen.ACTION_ERROR)) {
			Dialog.alert("Error: " + event.getData());
		}
}
```

4. Requesting extended permissions (optional)

By default, your application can access all general information in a user's profile, including her name, profile picture, gender, and friend list. If your application needs to access other parts of the user's profile that may be private, your application can request extended permissions. There is another class called PermissionScreen to handle this task.

```
permissionScreen = new PermissionScreen(fbc, cookieManager);
permissionScreen.addActionListener(this);
permissionScreen.requestPermissions(new String[] { ExtendedPermission.OFFLINE_ACCESS,
ExtendedPermission.PUBLISH_STREAM });
pushScreen(permissionScreen);
```

After the user granted extended permissions to your application, the ActionListener will be called back again with an "ACTION_GRANTED" event. And now you are done with the whole login process and ready to display the home screen of your BlackBerry application.

```
public void onAction(Action event) {
		if (event.getAction().equals(PermissionScreen.ACTION_GRANTED)) {
			try {
				popScreen(permissionScreen);
			} catch (IllegalArgumentException e) {
			}

			try {
				if (homeScreen == null) {
					homeScreen = new HomeScreen(fbc);
				}
				pushScreen(homeScreen);

			} catch (Exception e) {
				e.printStackTrace();
				Dialog.alert("Error: " + e.getMessage());
			}
		} else if (event.getAction().equals(PermissionScreen.ACTION_ERROR)) {
			Dialog.alert("Error: " + event.getData());
		}
}
```

5. Interacting with facebook using the Graph API.

Examples:

To retrieve the currently logged in user:

```
User user = fbc.getLoggedInUser();
```

To update the user's status:

```
user.setStatus("Hello world!");
```

To retrieve the user's friends list:

```
User[] friends = user.getFriends();
```

To retrieve user's news feed:

```
Post[] posts = user.getStream();
```

or to limit the number of posts to retrieve to 10 items:

```
Post[] posts = user.getStream(10);
```

To post to user's wall:

```
user.publishStream("This is the message", "http://www.your-url.com/", "Link name", "Link
caption", "Link description");
```

To post to the user’s friends’ walls:

```
User[] friends = user.getFriends();

if (friends.length > 0) {
    friends[0].publishStream("This is the message", " http://www.your-url.com/", "Link
name", "Link caption", "Link description");
}
```

To retrieve post's comments:

```
Post[] comments = post.getComments();
```