Glympse Intents Tutorial

Please contact partners@glympse.com if you need assistance, find a problem, or would like to request a feature.

The Glympse Application SDK for Android package consists of the following components:

- Glympse Intents Library (lib/GlympseIntentsLib/libs/glympseintentslib.jar)
- Sample project
- Tutorials and programming guides
- Complete reference documentation

The `GlympseIntentsLib` project is provided as a convenience library when using Intents to interact with the Glympse application. It is highly recommended that you use this library rather than creating your own intents because it will be kept up to date with any future changes to the Glympte Intent specification. If you are using schema handlers to interact with the Glympse application instead of intents then it is not necessary to use this library.

Android Manifest and Build Requirements

- The latest version of the Android SDK Tools for building.
- The minimum runtime platform level supported is Android 2.0 (API level 5).
- There are no manifest requirements for using this SDK.
- This method of interacting with Glympse requires that the Glympse application is installed on the device.

Building the Sample Project

The following assumes you are using the latest Android SDK Tools with Eclipse.

1. Unzip the Android Glympse Application SDK archive to a folder on your hard drive.
2. In Eclipse, under the **[File]** menu, choose **[Import…]**.
3. In the dialog that appears, expand the **[General]** list item, select **[Existing Projects into Workspace]** and then press **[Next >]**.
4. Ensure that **[Select root directory:]** is selected and either type in the path to the Glympse Application SDK root folder or use the **[Browse...]** button to navigate to it.
5. Under **[Projects]**, you should see the Glympse Intents library and sample project. Make sure that all projects are selected for import (they should be by default, but you can press the **[Select All]** button to do so if necessary) and that **[Copy projects into workspace]** is unchecked (unless you really *do* want the projects copied from the Glympse Application SDK root folder into your Eclipse workspace folder). Press the **[Finish]** button.
6. Under the **[Project]** menu, choose **[Build All]**. Note that it may be necessary to do this several times as many of the projects have dependencies on other projects in the workspace and they are not likely to build in the required order.

Adding Glympse Intents Library to a New or Existing Project

The following assumes you are using the latest Android SDK Tools with Eclipse.

1. Follow steps 1-4 in Building the Sample Projects.
2. Under **[Projects]**, you should see a library and sample project. Make sure that `GlympseIntentsLib` project is selected. Ensure that **[Copy projects into workspace]** is unchecked (unless you really *do* want the projects copied from the Glympse Application SDK root folder into your Eclipse workspace folder). Press the **[Finish]** button.
3. In the Package Explorer, right click on *your* project's root item and choose **[Properties]** from the menu.
4. In the dialog that appears, choose the **[Android]** list item in the left panel. In the panel on the right, scroll down to the **[Library]** section and press the **[Add...]** button. In the dialog that appears, choose `GlympseIntentsLib` and press the **[OK]** button. Press **[OK]** to dismiss the properties dialog.

Tutorial: Sending a Glympse using GlympseIntentsLib

Sending a Glympse via the Glympse Intents library requires just a few steps. Make sure you have already Added the Glympse Intents Library to your project.

1. Import the required Glympse Intents packages:

```java
import com.glympse.android.intent.GlympseApp;
import com.glympse.android.intent.CreateGlympseParams;
import com.glympse.android.intent.GlympseCallbackParams;
import com.glympse.android.intent.Recipient;
import com.glympse.android.intent.Common;
```

2. Create a CreateGlympseParams object and populate it with pre-configured data as desired. Many fields are optional, but a typical scenario where the invite is delivered by the app invoking the Intent is as follows:

```java
// Allocate a CreateGlympseParams object.
CreateGlympseParams glympseCreateParams = new CreateGlympseParams();

// Specify that we want a single "app" recipient.
String subtype = "your_app_name";
String brand = "your_app_brand_name";
Recipient recipient = Recipient.createNew(
    Recipient.TYPE_APP, subtype, brand, null, null, true);

// Recipient picker will be presented in read-only mode.
glympseCreateParams.setFlags(
    Common.FLAG_RECIPIENTS_READ_ONLY |
    Common.FLAG_USE_ACTIVITY_RESULT);

// Add that recipient to the parameter object.
glympseCreateParams.setRecipient(recipient);

// Set listener to be notified when ticket is created.
glympseCreateParams.setStatusListener(this);

// Generate a "create a glympse" Intent and start the activity for it.
Intent intent = GlympseApp.getCreateGlympseIntent(this, createGlympseParams);
```

```
startActivityForResult(intent, REQUEST_CREATE_GLYMPSE);
```
Text

3. When the invite is created, your app will receive a callback. In this example we passed `this`, which means this class needs to implement `GlympseApp.StatusListener`. See `GlympseIntentsSample` project for an example.

```java
/**
* Implementation of GlympseApp.StatusListener interface
*/
public void glympseDoneSending(GlympseCallbackParams params)
{
    // Get the list of recipients for the Glympse. There should be just one.
    Recipient[] recipients = params.getRecipients();

    // Check to see if we have at least one recipient and it has a URL.
    if ( (null != recipients) &&
        (recipients.length > 0) )
    {
        // Get invite as a url
        String url = recipients[0].getUrl();
        // Get duration of ticket in milliseconds
        long duration = params.getDuration();
        // Get the message specified by the user
        String message = params.getMessage();
        // See Glympse_Intents_Reference for all fields
    }
}

public void glympseFailedToCreate(GlympseCallbackParams params)
{
    // Receiving this means that Glympse wasn't able to create the invite
}
```
Java

4. In case if `Common.FLAG_USE_ACTIVITY_RESULT` was specified and `startActivityForResult()` was used to present Send a Glympse wizard, Glympse application will also notify caller on creation status using activity result intent:

```java
@Override protected void onActivityResult(
    int requestCode, int resultCode, Intent intent)
{
    // Check if we are returning from creating a Glympse. This is only
    // needed when not using a StatusListener to obtain the result.
    if ((REQUEST_CREATE_GLYMPSE == requestCode) &&
        (RESULT_OK == resultCode) && (null != intent))
    {
        glympseDoneSending(GlympseApp.getCreateResult(intent));
    }

    super.onActivityResult(requestCode, resultCode, intent);
```
Java

```
}
```

Tutorial: Viewing a Glympse using GlympseIntentsLib

Viewing a Glympse via the Glympse Intents library is also very simple. Make sure you have already Added the Glympse Intents Library to your project.

1. Import the required Glympse Intents packages:

```java
import com.glympse.android.intent.GlympseApp;
import com.glympse.android.intent.UriParser;
import com.glympse.android.intent.ViewGlympseParams;
```

2. `GlympseIntentsLib` has convenience methods for parsing Glympse invite codes and group names out of Strings. To launch an activity to view a Glympse, populate a `ViewGlympseParams` object with the parsed results and call GlympseApp.viewGlympse().

```java
// Grab the received text containing a Glympse invite URL. An example would be:
String buffer = "Sample text message that contains one or more Glympse URLs, such
as http://glympse.com/BOT-002";

// Parse the string and check to see if it has any Glympse URLs that
// contain glympse codes or group names.
UriParser parseBufferResult = GlympseApp.parseBuffer(buffer);
if ( parseBufferResult.hasGlympseOrGroup() )
{
    // Allocate a ViewGlympseParams object and tell it what we want to view.
    ViewGlympseParams glympseViewParams = new ViewGlympseParams();
    glympseViewParams.addAllGlympsesAndGroups(parseBufferResult);

    // Generate a "view a glympse" Intent and start the activity for it.
    GlympseApp.viewGlympse(this, true, glympseViewParams);
}
```

More Information

See the **Glympse_Intents_Reference** document for more details about using Intents to interact with Glympse.
See the **Glympse_URI_Schema** document for more details about using Glympse schema handlers.
If your question cannot be answered by these documents, please contact partners@glympse.com for assistance.