

Using the GTalk Service

Before you can access the GTalk Service, you need to import the `gtalkservice` library into your application with a `uses-library` tag inside the application node of the project manifest, as shown below:

```
<uses-library android:name="com.google.android.gtalkservice"/>
```

You also need to add the GTalk `uses-permission` tag, as shown in this XML snippet:

```
<uses-permission android:name="android.permission.GTALK"/>
```

Android Instant Messaging functionality is exposed through various interfaces as described below:

- ❑ `IGTalkService` Is used to create, access, and manage GTalk connections.
- ❑ `IGTalkConnection` A GTalk Connection represents a persistent socket connection between the device and the server it's connecting to. The GTalk Service creates a default connection upon start-up that you can access by calling `getDefaultConnection` on the GTalk Service object.
- ❑ `IImSession` Most instant messaging functionality is handled through the `IImSession` interface. It's used to retrieve the IM roster, set the user presence, obtain the presence of contacts, and manage chat sessions. Each GTalk Connection creates a default session, available through the `getDefaultSession` method.
- ❑ `IChatSession` All instant messaging chats are handled through the `IChatSession` interface. New Chat Sessions are created by initiating new chats, or joining existing ones, from an IM Session object. Using the Chat Session interface, you can send new chat messages, invite new participants to a group chat, and return a list of people involved in a chat.
- ❑ `IChatListener` Implement `IChatListener` to listen for messages in an IM Session or Chat Session. The `IChatListener` interface handlers listen for incoming messages, new chat participants, and people leaving a chat.
- ❑ `IGroupChatInvitationListener` Implement `IGroupChatInvitationListener` to listen for invitations to join group chats. The `onInvitationReceived` handler is passed a `GroupChatInvitation` that includes the username of the inviter, the room address, a "reason" (usually the room description), and the password you need in order to join the group chat.
- ❑ `IRosterListener` You can monitor your IM contacts roster, and the presence of the people on it, by implementing the `IRosterListener` interface. The Roster Listener includes event handlers that are fired when there are changes in a contact's presence as well as upon the addition and removal of contacts from the roster.

Binding to the GTalk Service

To use the GTalk Service, it must be bound to your application component using `bindService`.

The `bindService` method accepts two input parameters, an `Intent`, which specifies a component to bind to, and a `ServiceConnection` implementation. The following skeleton code demonstrates the pattern used to bind to the GTalk service:

```
IGTalkService gTalkService;
private void bindGTalk() {
    Intent i = new Intent();
    i.setComponent(GTalkServiceConstants.GTALK_SERVICE_COMPONENT);
    bindService(i, gTalkConnection, 0);
}
```

```
private ServiceConnection gTalkConnection = new ServiceConnection() { // When the service connects, get the default
    GTalk Session
    public void onServiceConnected(ComponentName className, IBinder service) {
        gTalkService = IGTalkService.Stub.asInterface(service);
    } // If the service disconnects
    public void onServiceDisconnected(ComponentName className) {
        gTalkService = null; } };
```

A bound GTalk Service represents a connection between your application and the GTalk Service APIs.

Before you can use the Service to use Android's Instant Messaging functionality, you need to initiate a new `GTalkConnection`, as shown in the following section.