



EBook Gratis

APRENDIZAJE quickblox

Free unaffiliated eBook created from
Stack Overflow contributors.

#quickblox

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con quickblox.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Importar quickblox Chat Android SDK.....	2
Preparar el servicio de chat.....	2
Inicia sesión para chatear.....	3
Crear nuevo diálogo de chat (privado).....	4
Solicitud de lista de diálogo de usuario registrado.....	4
Creditos.....	7

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [quickblox](#)

It is an unofficial and free quickblox ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official quickblox.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con quickblox

Observaciones

Esta sección proporciona una descripción general de qué es quickblox y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de quickblox y vincular a los temas relacionados. Dado que la Documentación para quickblox es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

Examples

Instalación o configuración

Instrucciones detalladas sobre cómo configurar o instalar QuickBox.

Vaya a <https://admin.quickblox.com> y haga clic en "Registrarse" en la parte superior o simplemente siga el enlace: <https://admin.quickblox.com/register> .

Importar quickblox Chat Android SDK

Añadir repositorio

```
repositories {
    maven {
        url "https://github.com/QuickBlox/quickblox-android-sdk-releases/raw/master/"
    }
}
```

Añadir en el Proyecto Gradle para funcionalidad de chat

```
dependencies {
    compile("com.quickblox:quickblox-android-sdk-chat:2.6.1")
}
```

Añadir en el Proyecto Gradle para la funcionalidad de video

```
dependencies {
    compile "com.quickblox:quickblox-android-sdk-videochat-webrtc:2.6.1"
}
```

Preparar el servicio de chat

Para inicializar el uso del servicio de chat:

```
QBChatService.setDebugEnabled (true); // habilitar el registro de chat
```

`QBChatService.setDefaultPacketReplyTimeout(10000);` // establece el tiempo de espera de respuesta en milisegundos para el paquete de conexión. Puede usarse para eventos como inicio de sesión, unirse al diálogo para aumentar el tiempo de respuesta de espera del servidor si la red es lenta.

Para configurar el socket de chat use **`QBChatService.ConfigurationBuilder`**;

```
QBChatService.ConfigurationBuilder chatServiceConfigurationBuilder = new
QBChatService.ConfigurationBuilder();
chatServiceConfigurationBuilder.setSocketTimeout(60); //Sets chat socket's read timeout in
seconds
chatServiceConfigurationBuilder.setKeepAlive(true); //Sets connection socket's keepAlive
option.
chatServiceConfigurationBuilder.setUseTls(true); //Sets the TLS security mode used when making
the connection. By default TLS is disabled.
QBChatService.setConfigurationBuilder(chatServiceConfigurationBuilder);
```

Inicia sesión para chatear

Crear sesión con usuario e iniciar sesión en QuickBlox Chat

```
// Initialise Chat service
QBChatService chatService = QBChatService.getInstance();

final QBUser user = new QBUser("garrysantos", "garrysantospass");
QBAuth.createSession(user, new QBEntityCallback<QBSession>() {
    @Override
    public void onSuccess(QBSession session, Bundle params) {
        // success, login to chat

        user.setId(session.getUserId());

        chatService.login(qbUser, new QBEntityCallback() {
            @Override
            public void onSuccess() {

            }

            @Override
            public void onError(QBResponseException errors) {

            }
        });
    }

    @Override
    public void onError(QBResponseException errors) {

    }
});
```

Para manejar diferentes estados de conexión use `ConnectionListener`:

```
ConnectionListener connectionListener = new ConnectionListener() {
    @Override
    public void connected(XMPPConnection connection) {
```

```

    }

    @Override
    public void authenticated(XMPPConnection connection) {

    }

    @Override
    public void connectionClosed() {

    }

    @Override
    public void connectionClosedOnError(Exception e) {
        // connection closed on error. It will be established soon
    }

    @Override
    public void reconnectingIn(int seconds) {

    }

    @Override
    public void reconnectionSuccessful() {

    }

    @Override
    public void reconnectionFailed(Exception e) {

    }
};

QBChatService.getInstance().addConnectionListener(connectionListener);

```

Crear nuevo diálogo de chat (privado)

```

QBChatDialog dialog = DialogUtils.buildPrivateDialog("USER_ID of other user");

QBRestChatService.createChatDialog(dialog).performAsync(new
QBEntityCallback<QBChatDialog>() {
    @Override
    public void onSuccess(QBChatDialog result, Bundle params) {
        //if dialog created successfully
        //result param return all details about that dialog
    }

    @Override
    public void onError(QBResponseException responseException) {
        //error creating dialog
    }
});

```

Solicitud de lista de diálogo de usuario registrado

Código para recibir los diálogos de chat del servidor Quickblox del usuario que inició sesión

(ejemplo con vista de lista)

```
private void receiveChatList() {
    QBRequestGetBuilder requestBuilder = new QBRequestGetBuilder();
    requestBuilder.setLimit(100);

    QBRestChatService.getChatDialogs(null, requestBuilder).performAsync(
        new QBEntityCallback<ArrayList<QBChatDialog>>() {
            @Override
            public void onSuccess(final ArrayList<QBChatDialog> result, Bundle params) {
                int totalEntries = params.getInt("total_entries");
                Log.wtf("chat", ""+result);
                TrumeMsgAdapter adapter=new TrumeMsgAdapter(this,result);
                chatlistView.setAdapter(adapter);
                chatlistView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
                    @Override
                    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
                        startActivity(new
Intent(this,ChatingActivity.class).putExtra("dialog",result.get(position)));
                    }
                });
            }
            @Override
            public void onError(QBResponseException responseException) {

            }
        });
}
```

Código del adaptador: -

```
public class TrumeMsgAdapter extends BaseAdapter {

    private ArrayList<QBChatDialog> chatlist;
    private Context context;

    public TrumeMsgAdapter(Context c,ArrayList<QBChatDialog> chatlist){
        this.chatlist=chatlist;
        this.context=c;
    }
    @Override
    public int getCount() {
        return chatlist.size();
    }

    @Override
    public Object getItem(int position) {
        return null;
    }

    @Override
    public long getItemId(int position) {
        return 0;
    }

    @Override
```

```
public View getView(int position, View convertView, ViewGroup parent) {
    View List;
    LayoutInflater inflater = (LayoutInflater) context
        .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    if (convertView == null) {
        List = inflater.inflate(R.layout.trume_msg_adapter, null);
        TextView username=(TextView) List.findViewById(R.id.UserName);
        TextView lastmessage=(TextView) List.findViewById(R.id.lastmessage);
        username.setText(chatlist.get(position).getName());
        lastmessage.setText(chatlist.get(position).getLastMessage());
    } else {
        List = convertView;
        TextView username=(TextView) List.findViewById(R.id.UserName);
        TextView lastmessage=(TextView) List.findViewById(R.id.lastmessage);
        username.setText(chatlist.get(position).getName());
        lastmessage.setText(chatlist.get(position).getLastMessage());
    }

    return List;
}
}
```

Lea Empezando con quickblox en línea:

<https://riptutorial.com/es/quickblox/topic/5719/empezando-con-quickblox>

Creditos

S. No	Capítulos	Contributors
1	Empezando con quickblox	Community , Guru_quickblox_support , jagapathi