LEARNING

quickblox

#quickblox

Table of Contents

About

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

Chapter 1: Getting started with quickblox

Remarks

This section provides an overview of what quickblox is, and why a developer might want to use it.

It should also mention any large subjects within quickblox, and link out to the related topics. Since the Documentation for quickblox is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Detailed instructions on getting quickblox set up or installed.

Go to https://admin.quickblox.com and click on "Register" at the top or just follow the link: https://admin.quickblox.com/register.

Import quickblox Chat Android SDK

Add repository

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

Add in Project Gradle for Chat Functionality

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

Add in Project Gradle for Video Functionality

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

Prepare Chat service

To initialise chat service use:

```
QBChatService.setDebugEnabled(true); // enable chat logging
```

QBChatService.setDefaultPacketReplyTimeout(10000);//set reply timeout in milliseconds for connection's packet. Can be used for events like login, join to dialog to increase waiting response time from server if network is slow.

To configure chat socket 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);
```

Login to Chat

Create session with User & Sign In to 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) {

    }
});
```

To handle different connection states 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);
```

Create New Chat Dialog (private)

```
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
          }
      });
```

Request List Of Dialog Of Logged In User

Code to receive Chat dialogs from Quickblox server of Logged in user (Example with listview)

```
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) {

            }
        });
```

}

Adapter Code:-

```
    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;
}
}
```

Read Getting started with quickblox online: https://riptutorial.com/quickblox/topic/5719/getting-started-with-quickblox

Credits

S. No	Chapters	Contributors
1	Getting started with quickblox	Community, Guru_quickblox_support, jagapathi