

 eBook Gratuit

APPRENEZ quickblox

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#quickblox

Table des matières

À propos	1
Chapitre 1: Démarrer avec quickblox	2
Remarques.....	2
Exemples.....	2
Installation ou configuration.....	2
Importer Quickblox Chat Android SDK.....	2
Préparer le service de chat.....	2
Connexion au chat.....	3
Créer un nouveau dialogue de discussion (privé).....	4
Liste de demande de dialogue de l'utilisateur connecté.....	4
Crédits	7

À propos

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

Chapitre 1: Démarrer avec quickblox

Remarques

Cette section fournit une vue d'ensemble de ce qu'est quickblox et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans quickblox, et établir un lien avec les sujets connexes. La documentation de quickblox étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Exemples

Installation ou configuration

Instructions détaillées sur la configuration ou l'installation de quickblox.

Allez sur <https://admin.quickblox.com> et cliquez sur «S'inscrire» en haut ou suivez simplement le lien: <https://admin.quickblox.com/register> .

Importer Quickblox Chat Android SDK

Ajouter un dépôt

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

Ajouter dans Project Gradle pour la fonctionnalité Chat

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

Ajouter dans Project Gradle pour la fonctionnalité vidéo

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

Préparer le service de chat

Pour initialiser l'utilisation du service de chat:

```
QBChatService.setDebugEnabled (true); // activer la journalisation du chat
```

`QBChatService.setDefaultPacketReplyTimeout(10000);` // définit le délai de réponse en millisecondes pour le paquet de la connexion. Peut être utilisé pour des événements tels que la connexion, joindre à la boîte de dialogue pour augmenter le temps de réponse en attente du serveur si le réseau est lent.

Pour configurer le socket de chat, utilisez **`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);
```

Connexion au chat

Créer une session avec l'utilisateur et se connecter à 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) {

    }
});
```

Pour gérer différents états de connexion, utilisez **`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);

```

Créer un nouveau dialogue de discussion (privé)

```

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
    }
});

```

Liste de demande de dialogue de l'utilisateur connecté

Code pour recevoir des boîtes de dialogue de conversation à partir du serveur Quickblox de

l'utilisateur connecté (exemple avec 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,ChattingActivity.class).putExtra("dialog",result.get(position)));
                    }
                });
            }
            @Override
            public void onError(QBResponseException responseException) {

            }
        });
}
```

Code de l'adaptateur: -

```
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;
}
}
```

Lire Démarrer avec quickblox en ligne: <https://riptutorial.com/fr/quickblox/topic/5719/demarrer-avec-quickblox>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec quickblox	Community , Guru_quickblox_support , jagapathi