

# Chat with SendBird

Somsak (Thumb)

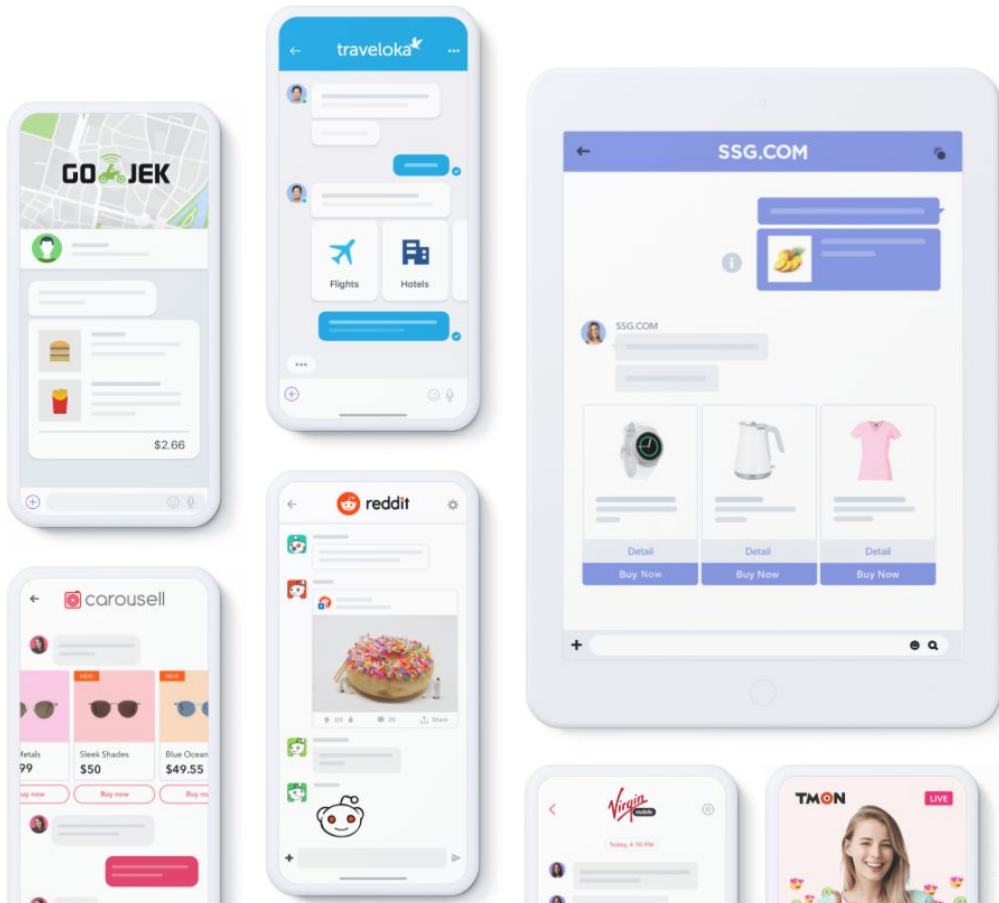
Growth Session #18 - September 13-14 2018

# Intro to SendBird



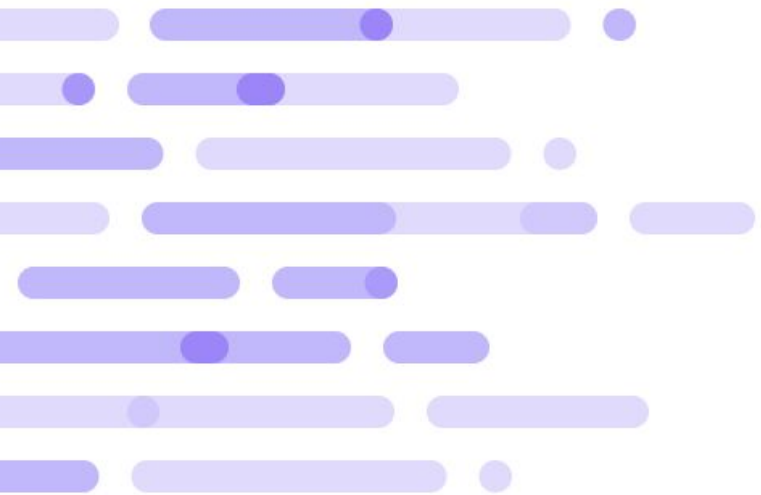
<https://sendbird.com/>

Chat platform with fully managed  
infrastructure



## Achievements and progress

- Integrates to one our current Ruby on Rails project.
- Research about SendBird JS SDK + Platform API.



+



- Creates new user to `SendBird`.
- Creates new 1-1 group channel for users.
- Sends message to specific channel.

## Create a new `SendBird` user:

```
user = SendBird::User.new('new-user', 'New User')  
  
user.create
```



```
{  
  :has_ever_logged_in=>false,  
  :session_tokens=>[],  
  :user_id=>"new-user",  
  :access_token=>"bc15a1aaf41a5fc265e2ffe8445fb19e141907c4",  
  :discovery_keys=>[],  
  :is_online=>false,  
  :last_seen_at=>0,  
  :nickname=>"New User",  
  :profile_url=>"", :metadata=>{}}  
}
```

## Create a new 1-1 group channel for users:

```
channel = SendBird::Channel.new  
  
channel.create
```



```
{  
  :cover_url=>"https://sendbird.com/main/img/cover/cover_02.jpg",  
  :freeze=>false,  
  :members=>  
    [  
      { :state=>"joined", :user_id=>"somsak_arnon", :is_online=>true, :is_active=>true,  
        :last_seen_at=>0, :nickname=>"Somsak", :profile_url=>"", :metadata=>{} },  
      { :state=>"joined", :user_id=>"nimbl3", :is_online=>false, :is_active=>true, :last_seen_at=>0,  
        :nickname=>"Nimbl3", :profile_url=>"", :metadata=>{} }  
    ],  
  :last_message=>nil,  
  :unread_mention_count=>0,  
  :is_public=>false,  
  :is_super=>false,  
  :joined_member_count=>2,  
  :channel_url=>"sendbird_group_channel_73263689_d0b9c3329aa09ddb0edc89d1a54a4788ee9cbf4e"  
}
```

## Send a message to specific channel:

```
message = SendBird::Message.new('nimbl3',  
  'Hello Nimbl3 Growth!!!')  
  
message.send
```

---



```
{  
  :custom_type=>"",  
  :mentioned_users=>[],  
  :updated_at=>0,  
  :user=>{:nickname=>"Nimbl3", :user_id=>"nimbl3", :profile_url=>"", :metadata=>{}},  
  :file=>{},  
  :message=>"Hello Nimbl3 Growth!!!",  
  :data=>"",  
  :type=>"MSG",  
  :created_at=>1536919082658,  
  :mention_type=>"users",  
  :channel_url=>"sendbird_group_channel_73263689_8dc93720b6ae187df937743d5097a315623949b6",  
  :message_id=>1827478725  
}
```

- Create a `SendBird` connection for the current user.
- Bind the necessary event handler to the channel.



## On the front-end

Creates a `SendBird` connection for the current user:

```
sb.connect(userId, accessToken, function(user, error) {});
```

## Binds the necessary event handlers to the channel:

```
var sb = new SendBird({ appId: appId });
var ChannelHandler = new sb.ChannelHandler();

ChannelHandler.onMessageReceived = function (channel, message) { }; // Received a chat message.
ChannelHandler.onMessageUpdated = function (channel, message) { }; // Received an updated chat message.
ChannelHandler.onMessageDeleted = function (channel, msgId) { }; // When a message has been deleted.
ChannelHandler.onChannelChanged = function (channel) { }; // When a channel property has been changed. More...
ChannelHandler.onChannelDeleted = function (channelUrl, channelType) { }; // When a channel has been deleted.
ChannelHandler.onReadReceiptUpdated = function (groupChannel) { }; // When read receipt has been updated.
ChannelHandler.onTypingStatusUpdated = function (groupChannel) { }; // When typing status has been updated.
ChannelHandler.onUserJoined = function (groupChannel, user) { }; // When a new member joined the group channel.
ChannelHandler.onUserLeft = function (groupChannel, user) { }; // When a member left the group channel.
ChannelHandler.onUserEntered = function (openChannel, user) { }; // When a new user entered the open channel.
ChannelHandler.onUserExited = function (openChannel, user) { }; // When a new user left the open channel.
ChannelHandler.onUserMuted = function (openChannel, user) { }; // When a user is muted on the open channel.
ChannelHandler.onUserUnmuted = function (openChannel, user) { }; // When a user is unmuted on the open channel.
ChannelHandler.onUserBanned = function (openChannel, user) { }; // When a user is banned on the open channel.
ChannelHandler.onUserUnbanned = function (openChannel, user) { }; // When a user is unbanned on the open channel.
ChannelHandler.onChannelFrozen = function (openChannel) { }; // When the open channel is frozen.
ChannelHandler.onChannelUnfrozen = function (openChannel) { }; // When the open channel is unfrozen.
ChannelHandler.onChannelHidden = function (groupChannel) { }; // When the group channel is hidden.

ChannelHandler.onMetaDataCreated = function (channel, metaData) { }; // When metaData is created.
ChannelHandler.onMetaDataUpdated = function (channel, metaData) { }; // When metaData is updated.
ChannelHandler.onMetaDataDeleted = function (channel, metaDataKeys) { }; // When metaData is deleted.
ChannelHandler.onMetaCountersCreated = function (channel, metaCounter) { }; // When metaCounter is created.
ChannelHandler.onMetaCountersUpdated = function (channel, metaCounter) { }; // When metaCounter is updated.
ChannelHandler.onMetaCountersDeleted = function (channel, metaCounterKeys) { }; // When metaCounter is deleted.

// Add this channel handler to SendBird object.
sb.addChannelHandler(UNIQUE_HANDLER_ID, ChannelHandler);
```

## Next Steps

Package SendBird as a reusable module



Full management is  
NOT FREE :)

# Thanks!

Contact Nimbl3

[hello@nimbl3.com](mailto:hello@nimbl3.com)

399 Sukhumvit Road, Interchange 21  
Klongtoey nua, Wattana  
Bangkok 10110

28C Stanley St,  
Singapore 068737

20th Floor, Central Tower  
28 Queen's Road  
Central, Hong Kong

[nimbl3.com](http://nimbl3.com)

