
MessengerX.io

Release 0.1.0

Sep 16, 2020

1	Why MessengerX.io?	3
2	How it Works?	5
3	What are Chatbots?	7
4	Get Started	9
4.1	Signup via our Dev Portal	9
4.2	Click on Add New App	9
4.3	Fill out the details	9
5	Setup your Chatbot Server	11
5.1	Understanding Webhooks	11
5.2	Understanding Message Payloads	11
5.3	Encrypted Incoming Message Payload	11
5.4	Decrypting Incoming Message Payload	11
6	Sending On-Demand Responses	13
6.1	Sending a Text Message with Quick Replies	13
6.2	Sending a Text Message with Buttons and Quick Responses	13
6.3	Sending a Media Element with Quick Replies	14
6.4	Sending a Media Element with Buttons & Quick Replies	15
6.5	Sending a List of Media Elements with Buttons & Quick Replies	15
7	Personalization, Tagging & User Engagement	17
7.1	Tag a User	17
7.2	Un-tag a User	17
7.3	Get all the Active Tags for a Particular User in Context	18
8	Announcements (Subscription Messaging)	19
8.1	Sample CURL Command	19
9	Headless CMS	21
9.1	Insert new content	21
9.2	Search API	21
10	Open Source Samples	23
10.1	RASA (Python Sample)	23

11 Client Integration via SDKs	25
11.1 Requirements	25
11.2 Integrate your bot on your website	25
11.3 Integrate your bot in your Android App	26
11.4 Add Gradle Dependency	26
11.5 Modify Manifest (Add Token)	26
11.6 Add SingleBotActivity Reference to Manifest (Bot Developers)	26
11.7 Intialize MACHAAO SDK	26
11.8 Launch Your Bot / Chat App via our SDK [For Bot Developers / Partners]	27
12 Bonus for Developers	29
13 Resources / Tutorials	31
13.1 RASA + MessengerX.io Chat Bot Tutorial (YouTube Video)	31
13.2 Chatbot Showcase	31
14 Other Resources	33
15 Small Businesses / Enterprise	35
16 Join our Gitter Community	37

MessengerX.io is a developer marketplace for everyday chat apps also known as chatbots.

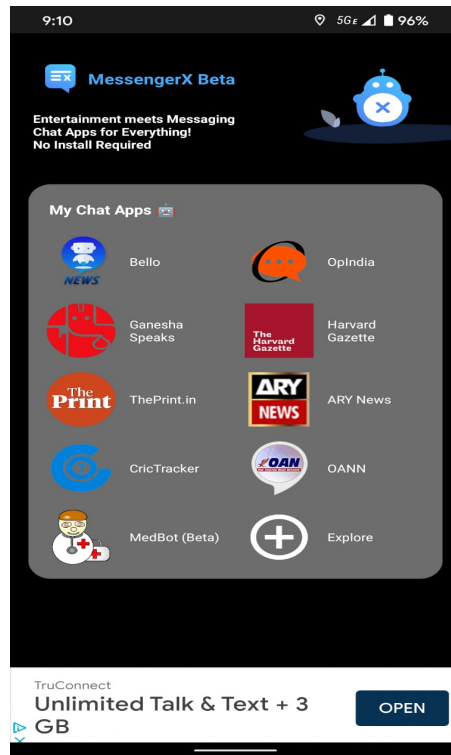


Fig. 1: MessengerX.io - Everyday Chat Apps for You

A joint venture between AppyHigh and MACHAAO with an aim to aid developers looking to build and monetize deeply personalized chat experiences.

CHAPTER 1

Why MessengerX.io?

With our unique learnings in the conversational space from processing over 2.5B+ messages from or to end users.

We would like to claim that we do know a thing or two about making good chatbots :)

We also see an evidential gap between chatbots and end users due to the operating nature of current messaging ecosystems.

We aim to bridge this gap between “great” chatbots and end users!

How it Works?

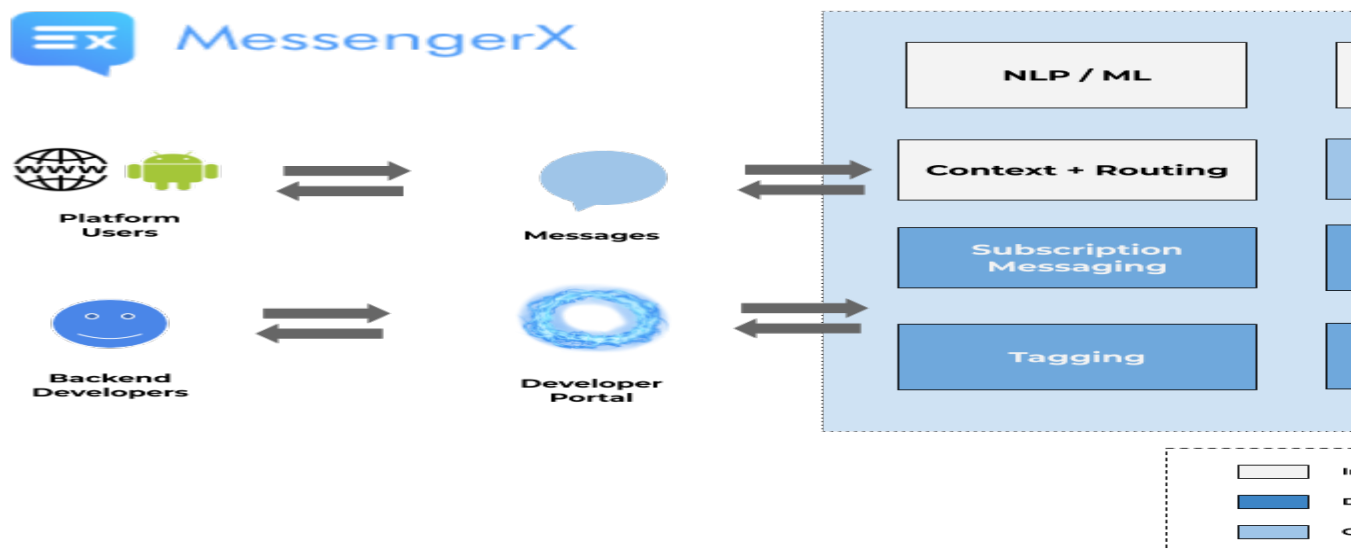


Fig. 1: How it Works? - MessengerX.io

When a user sends a message to your bot through Platform SDKs integrated inside publisher or partner app(s).

The following set of events sequence should take place when an incoming message is received destined for you bot.

- Our servers would route the incoming messages to Your Webhook or CHAT BOT URL, where your messaging app or chatbot is hosted.
- Using the Send Message API, the mini app or the chat bot can then respond to the person directly on the Partner App via our Platform APIs

The MessengerX Platform is FREE for developers looking to build highly engaging consumer based chatbots.

What are Chatbots?

Chat Apps also called as Chat Bots are functional conversational programs that one can chat with to carry on a series of complicated tasks in a quick fashion.

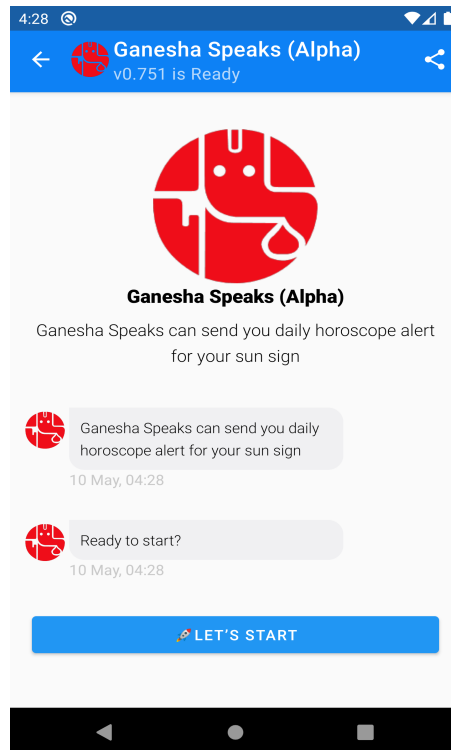


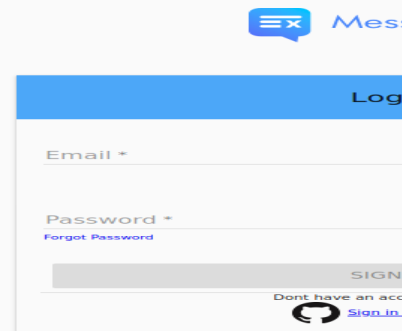
Fig. 1: Daily Horoscope Assistant (Powered by GaneshaSpeaks.com)

CHAPTER 4

Get Started

4.1 Signup via our Dev Portal

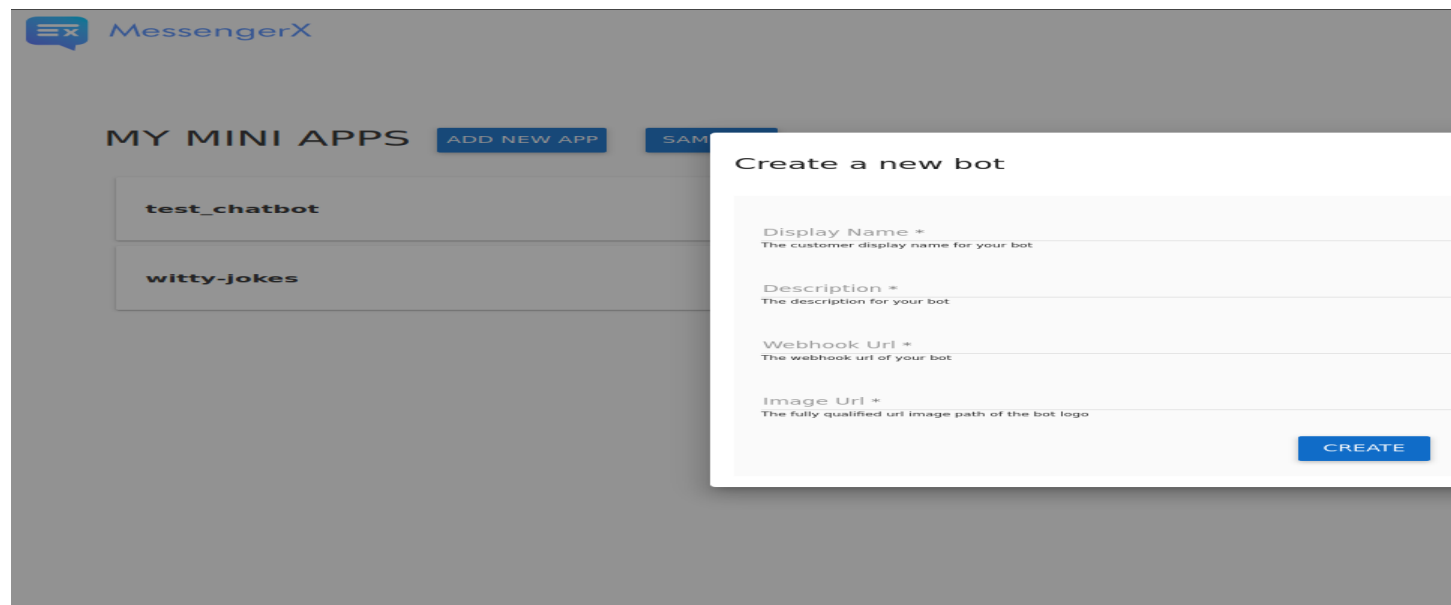
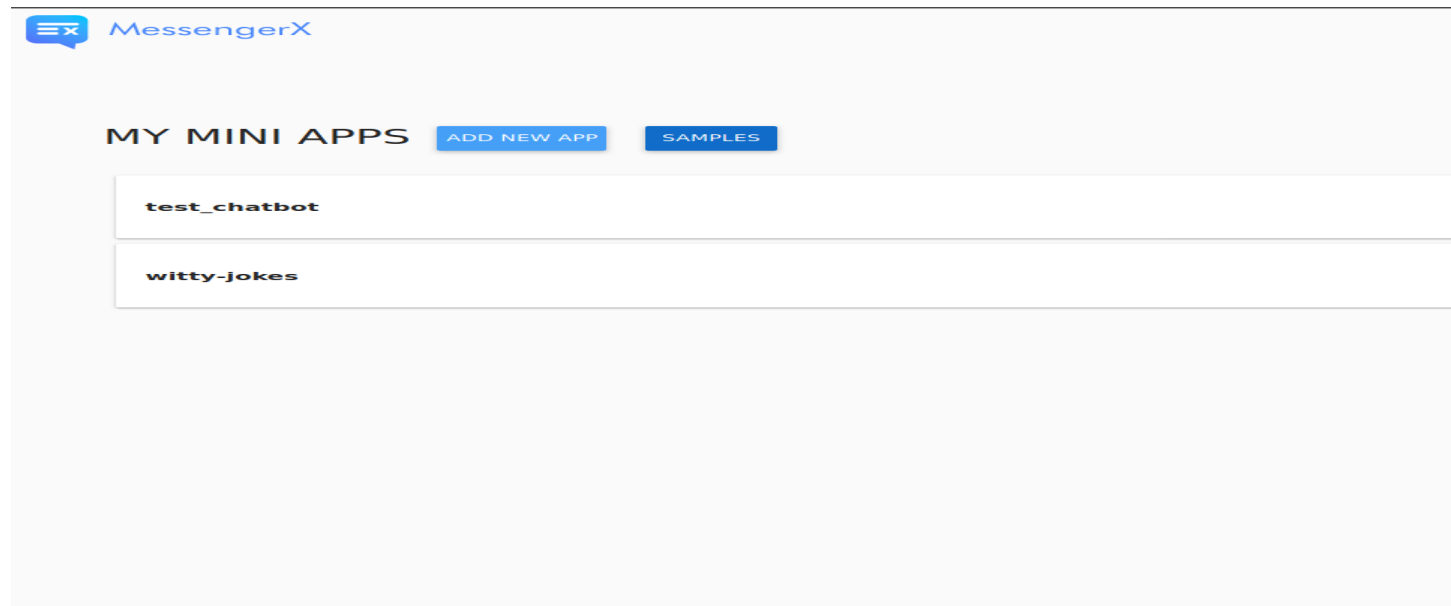
Signup and Get a FREE API Token @ <https://portal.messengerx.io>



4.2 Click on Add New App

4.3 Fill out the details

You can place Webhook and Image Url as None till deployment of your Chat App to get started.



Setup your Chatbot Server

5.1 Understanding Webhooks

Webhook is a REST endpoint url which denotes your custom functions / callbacks.

This is the end point which will receive any incoming messages destined for your bot.

5.2 Understanding Message Payloads

A message payload is a JSON representation of an incoming message sent by the end user.

5.3 Encrypted Incoming Message Payload

The incoming messages that your webhook will receive are encrypted using HS512 algorithm with your token as the key.

Below is the JSON representation of the raw incoming message.

```
{"raw": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTUzIj09eyJyZWUiOiJmYXZlbnR1cm91bnQ..."} 
```

5.4 Decrypting Incoming Message Payload

Decrypt the incoming payload with the secret_key provided.

You can use the open source JWT.io libraries available for all major programming languages

Let's quickly go through the sample representation of the decrypted incoming message object payload

- messaging: list of incoming message actions

- `message_data`: details about the nature of data sent.
- `user`: basic user info details about the user associated with the message.
- `sender`: the unique device id of the user.
- `client`: the partner app client id.
- `version`: android / web sdk client version
- `silent`: if silent is true, don't reply back to the request.

```
{
  "sub":{
    "messaging":[
      {
        "message_data":{
          "text":"hi",
          "action_type":"get_started"
        },
        "user":{
          "userId":"<!-- USER_ID -->",
          "creation_time":1589518339556,
          "device_id":"311b145ed6a96d6",
          "email":"<c11b145ed6a96d6>@machaa.com",
          "timezone":"-7.0"
        },
        "sender":"dWnjre9rTr65ZeiOmrY1oU",
        "silent":false,
        "client":"messenger.---.debug",
        "version":"0.838"
      }
    ]
  }
}
```

Sending On-Demand Responses

Send a outgoing message in response to user's input can be done by the usage of our On Demand Messaging APIs as demonstrated below

6.1 Sending a Text Message with Quick Replies

Below is an example CURL request to send a message / response to a particular user id using our Core Messaging APIs.

```
curl --location --request POST 'https://ganglia-dev.machaa.com/v1/messages/send' \  
--header 'api_token: API_TOKEN' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "users":["<!-- UNIQUE_USER_ID -->"],  
  "message":{  
    "text": "I am a good bot",  
    "quick_replies": [{  
      "content_type": "text",  
      "title": "Hi",  
      "payload": "hi"  
    }]  
  }  
}'
```

6.2 Sending a Text Message with Buttons and Quick Responses

Below is an example CURL request to send a message / response to a particular user id using our Core Messaging APIs.

```
curl --location --request POST 'https://ganglia-dev.machaaio.com/v1/messages/send' \
--header 'api_token: API_TOKEN' \
--header 'Content-Type: application/json' \
--data-raw '{
  "users":["<!-- UNIQUE_USER_ID -->"],
  "message":{
    "attachment":{
      "type":"template",
      "payload":{
        "template_type": "button",
        "text": "Hi, I am a good bot...!",
        "buttons": [{
          "title": "Hi",
          "type": "postback",
          "payload": "hi"
        }, {
          "title": "Source",
          "type": "web_url",
          "url": "https://provogue.s3.amazonaws.com/provogue-duffle1.jpg"
        }
      ]
    }
  },
  "quick_replies": [{
    "content_type": "text",
    "title": "Hi",
    "payload": "hi"
  }]
}
```

6.3 Sending a Media Element with Quick Replies

Below is an example CURL request to send a message attachment to a particular user id.

```
curl --location --request POST 'https://ganglia-dev.machaaio.com/v1/messages/send' \
--header 'api_token: API_TOKEN' \
--header 'Content-Type: application/json' \
--data-raw '{
  "users":["<!-- UNIQUE_USER_ID -->"],
  "message":{
    "attachment":{
      "type":"template",
      "payload":{
        "template_type":"generic",
        "elements":[
          {
            "title": "Test #786 - Duffle Bag + 200 Machaaio Credits",
            "subtitle":"Only Pay Shipping & Handling Charges. Combo Offer for_
↳Machaaio Users only.",
            "image_url":"https://provogue.s3.amazonaws.com/provogue-duffle1.jpg"
          }
        ]
      }
    },
    "quick_replies": [{
```

(continues on next page)

(continued from previous page)

```

        "content_type": "text",
        "title": "Hi",
        "payload": "hi"
    }}
}
}'

```

6.4 Sending a Media Element with Buttons & Quick Replies

Below is an example CURL request to send a message attachment to a particular user id with buttons.

```

curl --location --request POST 'https://ganglia-dev.machaa.com/v1/messages/send' \
--header 'api_token: API_TOKEN' \
--header 'Content-Type: application/json' \
--data-raw '{
  "users":["<!-- UNIQUE_USER_ID -->"],
  "message":{
    "attachment":{
      "type":"template",
      "payload":{
        "template_type":"generic",
        "elements":[
          {
            "title": "Test #786 - Duffle Bag + 200 Machaa Credits",
            "subtitle":"Only Pay Shipping & Handling Charges. Combo Offer for_
↪Machaa Users only.",
            "image_url":"https://provogue.s3.amazonaws.com/provogue-duffle1.jpg
↪",
            "buttons": [{
              "title": "Hi",
              "type": "postback",
              "payload": "hi"
            }, {
              "title": "Source",
              "type": "web_url",
              "url": "https://provogue.s3.amazonaws.com/provogue-duffle1.jpg"
            }
          ]
        }
      }
    },
    "quick_replies": [{
      "content_type": "text",
      "title": "Hi",
      "payload": "hi"
    }
  ]
}'

```

6.5 Sending a List of Media Elements with Buttons & Quick Replies

Below is an example CURL request to send a message attachment to a particular user id with buttons.

```

curl --location --request POST 'https://ganglia-dev.machaaio.com/v1/messages/send' \
--header 'api_token: API_TOKEN' \
--header 'Content-Type: application/json' \
--data-raw '{
  "users":["<!-- UNIQUE_USER_ID -->"],
  "message":{
    "attachment":{
      "type":"template",
      "payload":{
        "template_type":"generic",
        "elements":[
          {
            "title": "Test #786 - Duffle Bag + 200 Machaaio Credits",
            "subtitle":"Only Pay Shipping & Handling Charges. Combo Offer for_
↪Machaaio Users only.",
            "image_url":"https://provogue.s3.amazonaws.com/provogue-duffle1.jpg
↪",
            "buttons": [{
              "title": "Hi",
              "type": "postback",
              "payload": "hi"
            }, {
              "title": "Source",
              "type": "web_url",
              "url": "https://provogue.s3.amazonaws.com/provogue-duffle1.jpg"
            }
          ],
          {
            "title": "Test #787 - Duffle Bag + 200 Machaaio Credits",
            "subtitle":"Only Pay Shipping & Handling Charges. Combo Offer for_
↪Machaaio Users only.",
            "image_url":"https://provogue.s3.amazonaws.com/provogue-duffle1.jpg
↪",
            "buttons": [{
              "title": "Hi",
              "type": "postback",
              "payload": "hi"
            }, {
              "title": "Source",
              "type": "web_url",
              "url": "https://provogue.s3.amazonaws.com/provogue-duffle1.jpg"
            }
          ]
        ]
      }
    },
    "quick_replies": [{
      "content_type": "text",
      "title": "Hi",
      "payload": "hi"
    }
  ]
}'

```

Personalization, Tagging & User Engagement

The personalization and engagement api is the core base to build sophisticated re-engaging bots.

The process starts with tagging a user, Tagging a user allows you to open up multiple re-targeting or re-engagement use cases such as sending daily news, personalized responses, etc.

This opens up your chat bot to support variety of deeply personalized use cases without a need for a huge dev ops team.

7.1 Tag a User

Annotate or Tag a user with values for deeper personalization.

```
curl --location --request POST 'https://ganglia-dev.machaa.com/v1/users/tag/<USER_ID>' \
  --header 'api_token: <API_TOKEN>' \
  --header 'Content-Type: application/json' \
  --data-raw '{
  "tag": "preferred_languages",
  "status": 1,
  "values": ["en", "fr"],
  "displayName": "Languages"
}'
```

7.2 Un-tag a User

Un-tag a user for deeper personalization.

```
curl --location --request POST 'https://ganglia-dev.machaa.com/v1/users/tag/<USER_ID>' \
  --header 'api_token: <API_TOKEN>' \
```

(continues on next page)

(continued from previous page)

```
--header 'Content-Type: application/json' \  
--data-raw '{  
  "tag": "preferred_languages",  
  "status": 0  
}'
```

7.3 Get all the Active Tags for a Particular User in Context

Get all the tags for deeper personalization use cases.

```
curl --location --request GET 'https://ganglia-dev.machaa.com/v1/users/tags/<USER_ID>  
↪' \  
--header 'api_token: <API_TOKEN>' \  
--header 'Content-Type: application/json'
```

Announcements (Subscription Messaging)

Sending announcements in order to re-engage your bot user (rate limited to a max of 1 per hour per user)

8.1 Sample CURL Command

Below is an example CURL request to send an announcement to a particular tag or list of tags using our Announcement APIs.

```
curl --location --request POST 'https://ganglia-dev.machaa.com/v1/messages/announce' \
↪ \
--header 'api_token: API_TOKEN' \
--header 'Content-Type: application/json' \
--data-raw '{
  "tags":["india", "pakistan", "usa"],
  "message":{
    "text": "I am a good bot",
    "quick_replies": [{
      "content_type": "text",
      "title": "Hi",
      "payload": "hi"
    }]
  }
}'
```


Tagging a user allows you to open up multiple re-targeting or re-engagement use cases such as sending daily news, personalized responses, etc.

9.1 Insert new content

Auto-Annotate and insert content for your chat app.

```
curl --location --request POST 'https://ganglia-dev.machaa.com/v1/content' \  
--header 'api_token: <API_TOKEN>' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "url": "https://www.youtube.com/watch?v=5UmM-tclggg",  
  "tags": ["india", "pakistan", "bangladesh"]  
'
```

9.2 Search API

Allows you to perform search content for your chat app by query based in a paginated way.

```
curl --location --request GET 'https://ganglia-dev.machaa.com/v1/content/search?  
→q=ipl&limit=10&skip=0' \  
--header 'api_token: <API_TOKEN>' \  
--header 'Content-Type: application/json'
```


CHAPTER 10

Open Source Samples

Below are some samples which will help get you started <https://github.com/machao/machao-samples>

10.1 RASA (Python Sample)

<https://github.com/machao/rasa-sample-nlu-bot>

11.1 Requirements

JDK 8 Android Studio

11.2 Integrate your bot on your website

Step_1: Below is a sample script you need to paste into your website to install webchat for your chatbot.

```
<script
  id="webchat"
  src="https://dev.messengerx.io/bot/js/widget.js"
  type="text/javascript"
  themecolor="#2196f3"
  chathost="https://dev.messengerx.io"
  botname="<!-- your bot name -->"
  machaaokay="<!-- your api token -->"
  avatarurl="<!-- your bot logo url -->"
></script>
```

Step_2: You will need to update variables above as shown below:

themecolor : Put the desired color in hex or rgb format which will be applied to the chat header background, buttons and message bubble background.

botname : The name of the bot that will appear on the chat header

avatarurl : The url of the image that is shown on bot launcher icon.

chathost : Url where the static assets for the webchat are hosted.

machaaokay : The API token for your bot provided by Machaa

11.3 Integrate your bot in your Android App

Add following to your app gradle file.

```
maven {
    url "https://machaa0-android-builds.s3.amazonaws.com/sdk/android/snapshots"
}
```

11.4 Add Gradle Dependency

```
debugImplementation('com.machaa0.android:machaa0-sdk:0.874-SNAPSHOT') {
    transitive = true
}

releaseImplementation('com.machaa0.android:machaa0-sdk:0.874') {
    transitive = true
} // [contact support@messengerx.io for access]
```

11.5 Modify Manifest (Add Token)

```
<meta-data android:name="com.machaa0.android.sdk.token"
android:value="<!-- Replace with your bot / api token assigned by MACHAAO Inc.
↳ (connect@machaa0.com) -->" />
```

11.6 Add SingleBotActivity Reference to Manifest (Bot Developers)

```
<activity android:name="com.machaa0.android.sdk.activities.SingleBotActivity"
    android:logo="@mipmap/ic_launcher"
    android:windowSoftInputMode="stateHidden|adjustResize"
    android:screenOrientation="portrait"
    android:launchMode="singleTop"
    tools:node="merge"
    android:theme="@style/AppTheme.Orange">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <action android:name="android.intent.action.VIEW" />
    </intent-filter>
</activity>
```

11.7 Intialize MACHAAO SDK

```
Machaa0.initialize(this);
```

11.8 Launch Your Bot / Chat App via our SDK [For Bot Developers / Partners]

```
Intent intent = new Intent(this, SingleBotActivity.class);
intent.putExtra("botToken", botToken);
startActivity(intent);
```

Sample Android Chat App @ https://github.com/machaa0/machaa0-samples/tree/master/basic_sample_bot/client/android/sample_bot_client

CHAPTER 12

Bonus for Developers

In addition to the massive savings on marketing and infrastructure costs, the platform also offers multiple other Rest APIs for developers looking to build deeply personalized chatbots:

- Rich Messaging Support via On Demand Messaging API
- Deep Personalization via Tagging API
- Auto ML based Engagement via Announcement API
- Data Capture API (Subjected to Approval)
- Transactional Wallet API (Subjected to Approval)
- FREE Hosting for your chat bot (Subjected to Approval)
- Guaranteed Message Processing (Subjected to Approval)
- Admin Dashboard (Premium)

13.1 RASA + MessengerX.io Chat Bot Tutorial (YouTube Video)

This is a beginner's guide on how to build a python & RASA chatbot from scratch.

The video helps you build your first sample bot using RASA's open-source NLU template and lets you train your bot by writing logic in python.

13.2 Chatbot Showcase

MedBot - Your Pocket Medical Guide

Built using Python + RASA using MessengerX API

Check out <https://www.messengerx.io/medbot> for a web demo

CHAPTER 14

Other Resources

Partner Deck: <https://docsend.com/view/jxdbref>

CHAPTER 15

Small Businesses / Enterprise

- Do you have an existing facebook messenger chatbot?
- Make your existing chatbot / platform work inside your client android app or website within hours (iOS coming soon).
- Conversational Bot Designer (Premium)

CHAPTER 16

Join our Gitter Community

<https://gitter.im/messengerx-io/community>