



Telegram Channel Update Bot

Readme

Version 1.0
12 April 2020

Table of Contents

1. Introduction	3
1.1 Overview	3
1.2 Use cases	3
2. Getting Started	4
2.1 Quick Start	4
2.1.1 Configuration and Use	5
3. Requirements & Prerequisites	9
3.1 System Requirements	9
3.2 Prerequisites	9
4. Support & FAQs	10
4.1 Support.....	10
4.2 FAQs	10
Appendix A: References.....	11

1. Introduction

This document contains all essential information for the user to make full use of this A2019 *Telegram Channel Update* bot. It includes a description of the functions and capabilities and step-by-step procedures for setup & configuration of the *Telegram API bot*.

1.1 Overview

A2019 Telegram API bot can be used to send a message to a channel using Telegram Bot token and recipient Chat ID. It utilizes the Telegram 'sendMessage' API to send messages.

1.2 Use cases

The key use cases include:

- Having a Telegram Bot to send periodic updates to a Telegram channel (Market Indices, Stock Availability, Status updates etc.)
- Using this bot as a helper bot (sub-task) to send Telegram message(s) to subscribed Telegram Channels

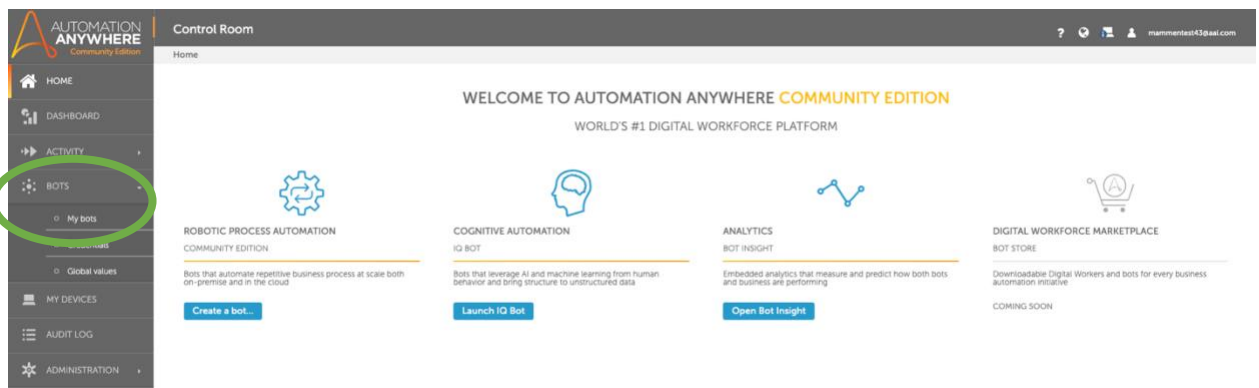
2. Getting Started

2.1 Quick Start

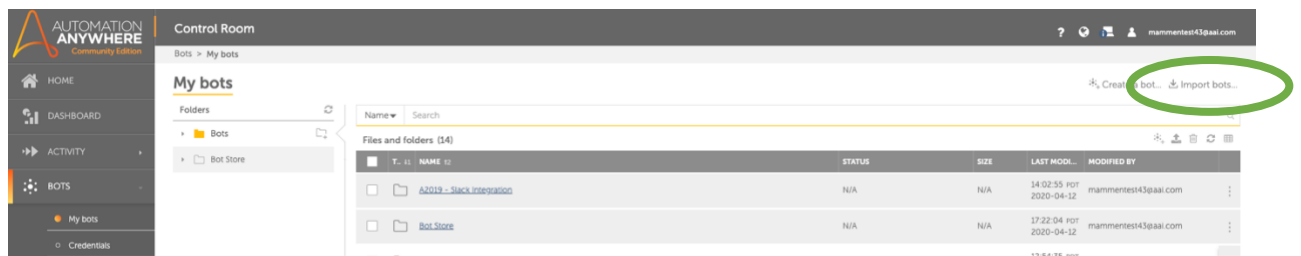
3.1.1 Import Bot

- Download the Bot from the [Community Edition Bots page](#) to your computer.
- Launch the Community Edition Control Room by logging into your Community Edition account. Note: If you do not have a Community Edition account, sign up [here for free](#).

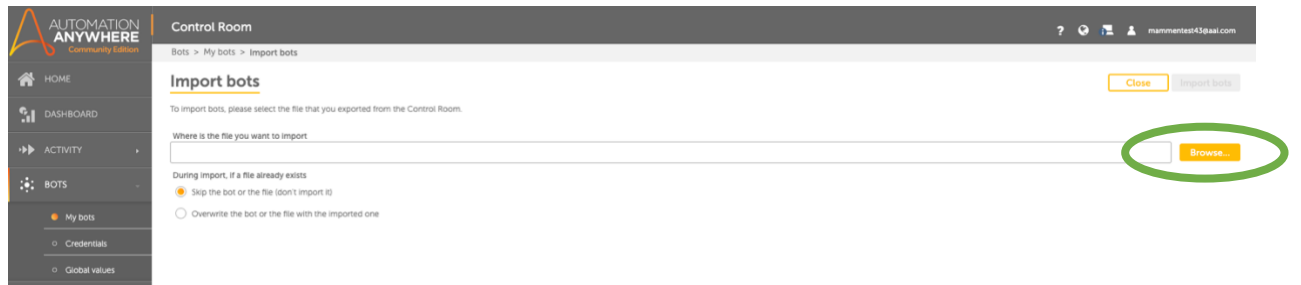
Navigate to **My Bots** by clicking **Bots** in the left panel and then select **My Bots**:



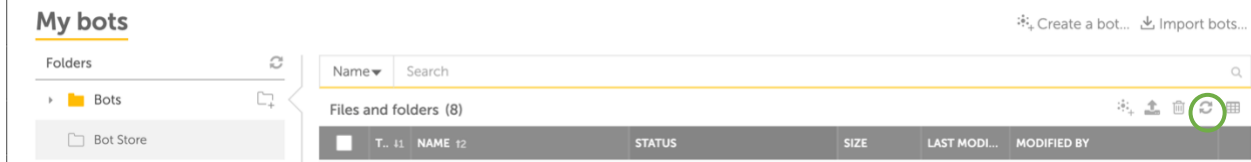
Click **Import Bots**:



Click **Browse** and select the Bot you would like to import:



Refresh the bot list by clicking on the “Refresh table” button to access the newly imported bot.



2.1.1 Configuration and Use

A2019 Telegram API Bot v1.0 can be deployed either in attended or unattended automation.

In attended mode, *input* the following variables while running the bot or the bot will request user to input these values during the execution of the bot, if the values are not passed initially –

S/N	Variable Name	Variable Type	Function
1	sChatID	String	Unique ChatID of the Telegram Channel or Telegram Bot
2	sBotToken	String	Unique Telegram Bot Token used for API authentication
3	sMessageText	String	Text Message that will be delivered via Telegram API call

Set input variables

Cancel Confirm

sChatID

sBotToken

sMessageText

Note: For creating Telegram Bot, retrieving Bot Token and ChatID, refer to Appendix B for Telegram documentation link at the end of this document and are not the functions of this bot.

In unattended mode, the input variables *sChatID* & *sBotToken* can be initialized with a default value and the bot can be modified to fetch the *sMessageText* from another bot/source.

Edit variable Cancel Apply

Name
sChatID

Description (optional)

Constant (read-only)

Use as input Use as output

Type
String

Default value (optional)
<Default Value>

Edit variable Cancel Apply

Name
sBotToken

Description (optional)

Constant (read-only)

Use as input Use as output

Type
String

Default value (optional)
<Default Value>

A2019 Telegram API Bot v1.0 execution via Control Room APIs –

Method: POST

API URL: [http\(s\)://<controlroomIP>/v3/automations/deploy](http(s)://<controlroomIP>/v3/automations/deploy)

Header: X-Authorization Token

Sample Body:

```
{
  "name": "Telegram API Bot v1.0.botrunner",
  "automationName": "Telegram API Bot v1.0.botrunner",
  "fileId": "<fileID>",
  "fileName": "Telegram API Bot v1.0.bot",
  "botInput": {
    "sChatID": {
      "type": "STRING",
      "string": "<sChatID>"
    },
    "sBotToken": {
      "type": "STRING",
      "string": "<sBotToken>"
    }
  },
  "status": "ACTIVE",
  "rdpEnabled": false,
  "runElevated": false,
  "setAsDefaultDevice": false,
  "poolIds": [],
  "workspaceName": "<folder>",
  "timeZone": "Asia/Singapore",
  "runAsUserIds": [
    "<BotRunnerUserID>"
  ],
  "scheduleType": "INSTANT"
}
```

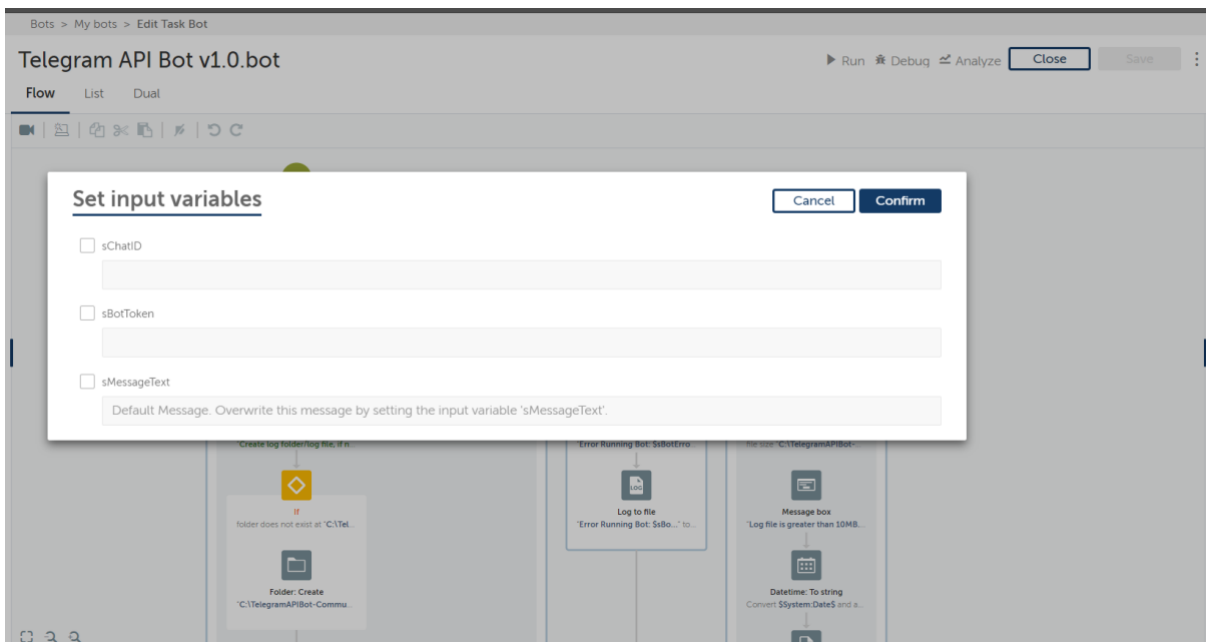
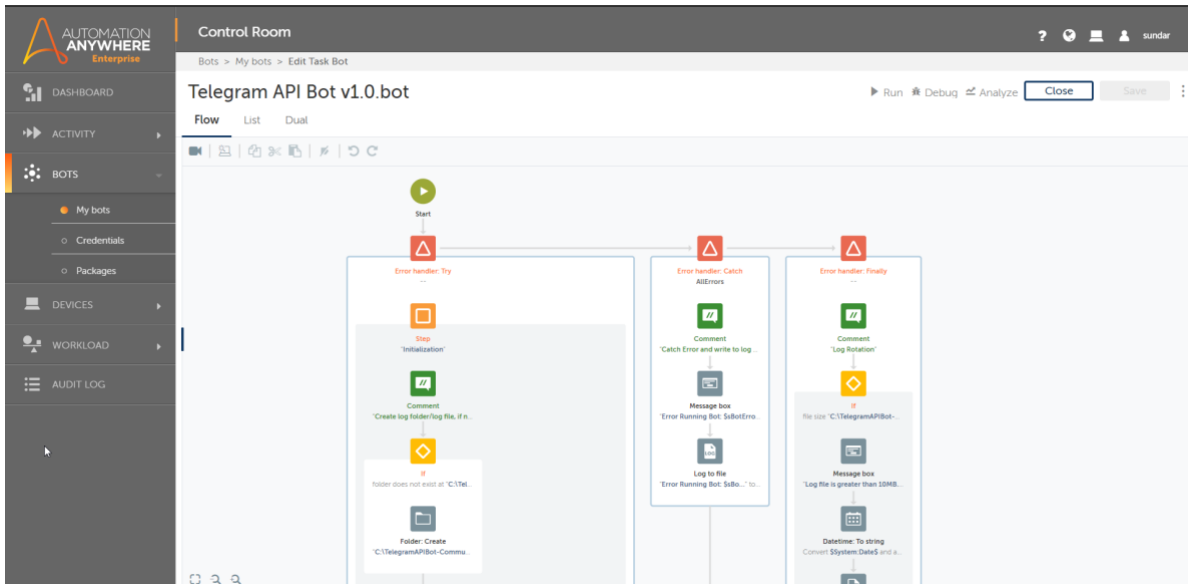
Sample Response:

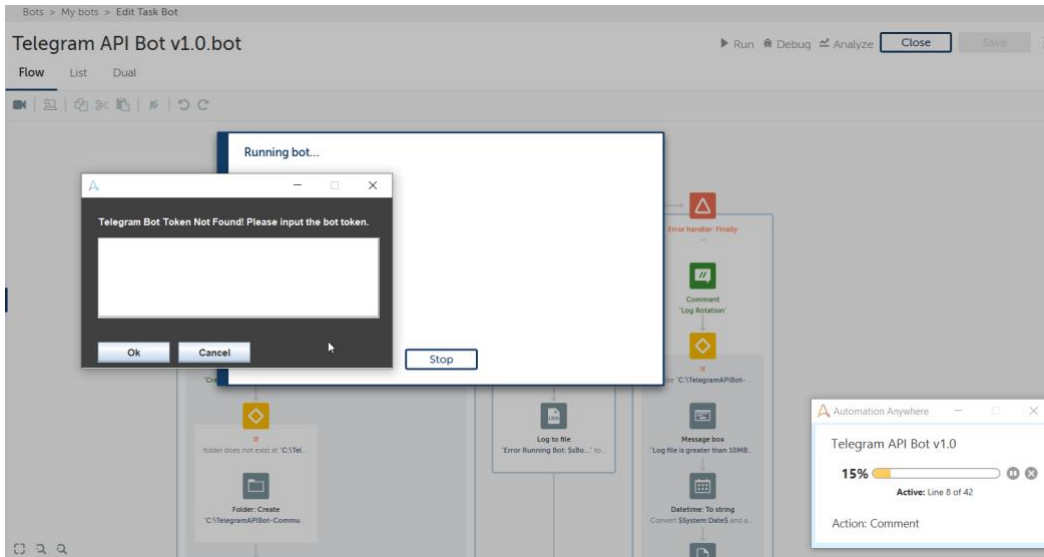
```
{
  "deploymentId": "6bbb5ba-d5dd-4f5f-9186-acf11a0ab341"
}
```

Log File

- A2019 Telegram API Bot will log each successful/unsuccessful Telegram API message delivery under “C:\TelegramAPIBot-CommunityEdition\TelegramAPIBot-CommunityEdition_log.txt” file with timestamp (log folder and log file will be automatically created, if not present).
- Log files will be rotated automatically after reaching 10MB in size.

Sample Screenshots:





3. Requirements & Prerequisites

3.1 System Requirements

[Enterprise A2019 \(Cloud deployed\) and Community Edition device requirements.](#)

Review the machine hardware specifications, operating system versions, and browser types supported by Automation Anywhere Enterprise for creating and running bots and command packages as an Enterprise A2019 (Cloud deployed) or Community Edition user on your local machine.

3.2 Prerequisites

- Automation Anywhere Enterprise A2019 Build 10 & above (On-Prem/Cloud/CommunityEdition)
- Internet Connection with access to Telegram API servers

4. Support & FAQs

4.1 Support

Free bots are not officially supported. You can get access to Community Support through the following channels:

- You can get access to Community Support, connecting with other Automation Anywhere customers and developers on [APeople](#) – the [Bot Building Forum](#), the [Bot Store Support Forum](#), or the [Developers Everywhere Group](#).
- Automation Anywhere also provides a [Product Documentation portal](#) which can be accessed for more information about our products and guidance on [Enterprise A2019](#).

4.2 FAQs

For questions relating to Enterprise A2019: See the [Enterprise A2019 FAQs](#).

Appendix A: References

No.	Topic	Reference Link
1	Overview of Enterprise A2019	Click here
2	Guidance: Building basic A2019 bots	Click here
3	Guidance: Building A2019 action packages	Click here
4	APeople Community Forum	Click here
5	Automation Anywhere University	Click here
6	Telegram Bot Documentation	Click here