



Bot Starter Template

Readme

Version 1.0
15 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 Import Bot.....	4
2.1.2 Setup	5
2.1.3 Configuration and Use	5
3. Requirements & Prerequisites	6
3.1 System Requirements	6
3.2 Prerequisites.....	6
4. Support & FAQs.....	7
Appendix A: References.....	8

1. Introduction

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

1.1 Overview

A2019 Bot Starter bot can be used as the starting point of your bot building journey. This template will give you insight into the best bot building practices, along with additional logging capabilities for the automation you're about to create.

1.2 Use cases

The key use cases include:

- Providing a template for users to create a resilient bot
- Providing logging and error handling in a consistent manner

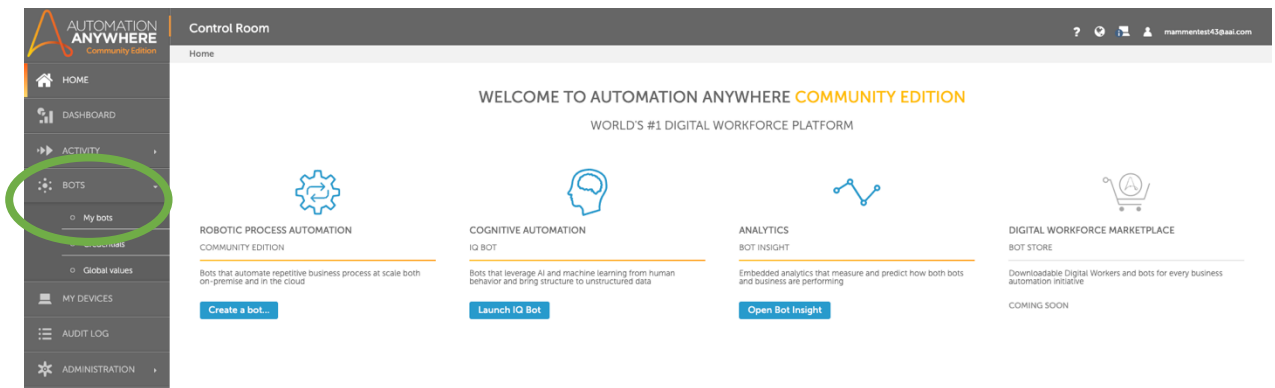
2. Getting Started

2.1 Quick Start

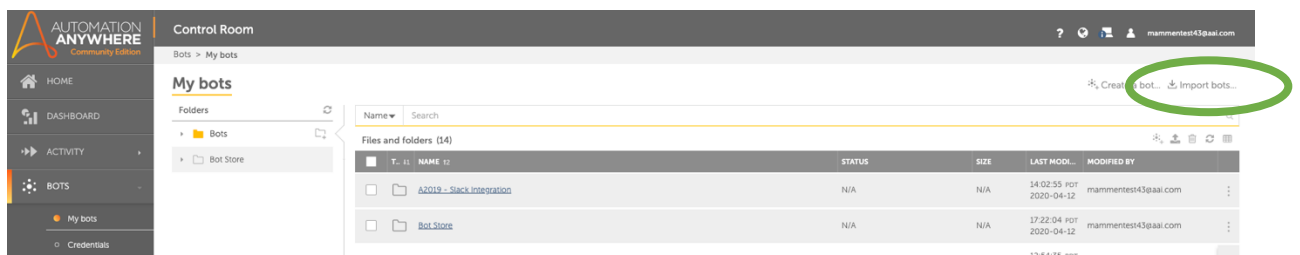
2.1.1 Import Bot

- Download the Telegram Web 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](#).

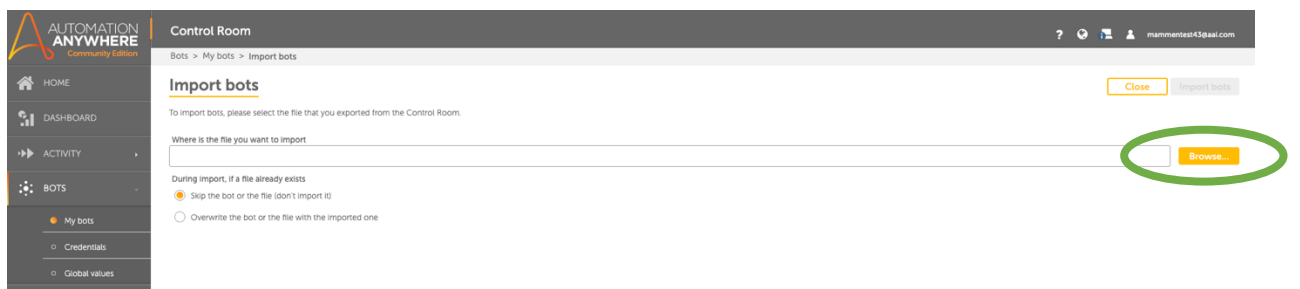
Navigate to **My Bots** by clicking **Bots** in the left panel and then select **My Bots**:



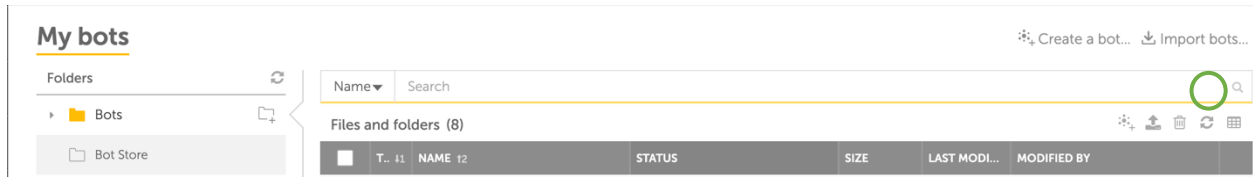
Click **Import Bots**:



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.



2.1.2 Setup

Download the A2019 *Bot Starter* bot file and import into your A2019 Control Room Repository.

2.1.3 Configuration and Use

User to fill in the following details:

- Line 4 – Add the Bot Name to the field “Select the source string variable(s)/value (optional)”
- Line 5 – Add the Vendor Name to the field “Select the source string variable(s)/value (optional)”

Even though the fields are titled “optional”, the values are required for Line 4 and Line 5.

Start building your bot from line 51.

Note:

The whole purpose of the bot is to provide logging and error handling in a consistent way so that developers can just focus on building their actual logic without having to worry about how to write the error handling.

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/Community Edition)
- Google Chrome Browser
- Internet Connection
- Any other requirement that needs to be met to successfully complete your automation use-case.

4. Support & FAQs

4.1 Support

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

- Community Support, connecting with other Automation Anywhere customers and developers on [APeople](#)
- Automation Anywhere's [Product Documentation portal](#), which can be accessed for more information about our products and guidance on [Enterprise A2019](#).

4.2 FAQs

Q: Can I make changes to this bot?

A: Absolutely. Free bots are for you to make use of, customize, and/or include in your builds as you see fit.

Q: Can I upgrade the packages for this bot?

A: Yes – but know that the bot was created/tested on a specific package version. Any modifications to that may result in unexpected outcomes – so make sure to test the bot/subtask after making any package version modifications.

Q: How can I change the output format of the bot?

A: When calling a subtask, the return value is always of type dictionary, and the value(s) returned from the subtask can be access through their specific variable names as a key within the dictionary. Once that value has been returned, the resulting number could be converted to a string should you need it to be.

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