# **Mule Hackathon 2020**

## **WhatsApp Messaging Automation using Sikuli, MuleSoft, and Windows Batch**

#### Sreenivasa Komma & Venkatarami Konudula

* Purpose of this document is to capture
  + Information needed to better understand code and explanation of key logic
  + Screenshots to aid the explanation
  + Screenshots from the execution
  + Other pertinent information

|  |  |
| --- | --- |
| C:\ sikulix\**WindowsWakeup.bat** | start "Run Mule and Trigger API" cmd **/t:07** **/c** "echo hello **&** java -jar %SIKULIX\_HOME%\sikulixide.jar **-f** %SIKULIX\_HOME%\sikuli.log -r %SIKULIX\_HOME%\Start-Mule-And-Trigger-API.sikuli\Start-Mule-And-Trigger-API.py **--args** mule.bat %SIKULIX\_HOME%\sikuli\_images %SIKULIX\_HOME%\curlPostInput.json **<http.host for Mule APIs> <http.port for Mule APIs>**"  **/t:07** is to force command window background color to black and foreground color to white.  **/c** option is to carry out the command and then terminate  **java -jar %SIKULIX\_HOME%\sikulixide.jar –** Invoke SikuliX Desktop code  **-f** option is to write log file  **-r** option is to run the script, in this case, **Start-Mule-And-Trigger-API.py**  **--args** pass the arguments to the code  All the arguments enclosed in **<….>** need values to specific to your environment or configuration. |
| C:\sikulix\Start-Mule-And-Trigger-API.sikuli\**Start-Mule-And-Trigger-API.py** | Logic   1. setBundlePath(sikuliImagesFolder) - Point the images folder to C:\sikulix\sikuli\_images 2. Turn off CAPS\_LOCK if on 3. Open WhatsApp, check if it is dark theme, change it to light theme for the run. Will be reverted at the end of the test. **All the images for automation are based on light themed images.** That’s the reason for this step. 4. Start Mule standalone run time which should deploy the code in **%MULE\_HOME\apps** folder 5. Until code deployment is done, Mule health check API is called every few seconds 6. Once success is returned, it triggers WhatsApp automation API (POST call) 7. Following the update, restore the theme for WhatsApp if it was changed in Step (3) |
| Mule hosted APIs | * **Healthcheck API**: http://<http.host>:<http.port>/whatsapp/healthcheck (GET type)   Above Python script in a loop checks for the status code 200 from calling this API to make sure Mule runtime is up and code is deployed to serve API requests.   * **WhatsApp Automation API**: http://<http.host>:<http.port>/whatsapp (POST type)   **Input JSON body is**: C:\sikulix\curlPostInput.json. Supply value for **system\_default\_pictures\_folder**  {  "sikuli\_home\_folder" : "C:\\sikulix",  "sikuli\_images\_sub\_folder": "sikuli\_images",  "system\_default\_pictures\_folder": **"Your pictures root folder",**  "whatsapp\_desktop\_exe": "WhatsApp.exe",  "input\_csv\_file\_name" : "input\_data.csv",  "process\_done\_flag\_file": "whatsapp\_update\_done.txt"  } |
| Mule Flow & Code | **Healthcheck Flow**    **WhatsApp Automation Flow**    **Mule Flow XML file:**  https://github.com/srkomma/mule-hackathon-2020/blob/master/whatsapp-automation/src/main/mule/whatsapp-automation.xml  **Java Code used in Invoke Static component:**  https://github.com/srkomma/mule-hackathon-2020/blob/master/whatsapp-automation/src/main/java/com/dev/whatsapp/WhatsAppAutomationViaSikuliJava.java  **It mainly does two things:**  1) Send greeting based on data configured in the file **C:\sikulix\input\_data.csv**  2) It looks for a profile picture to update your WhatsApp profile with, in the folder:  **<system\_default\_pictures\_folder>** that you supplied in POST input **+** **“\”** + **4 digit year from current date**.  Example: C:\Users\srk\Pictures\2020  Acceptable picture formats are: {".png", ".gif", ".jpg", ".jpeg"}   * + Any special profile picture should be saved with prefix “DP”.   + Default profile picture should be saved with Prefix “SELF”.   Logic:   * + If a profile picture with DP prefix is not found, then default profile picture starting with SELF is set as profile pic. Also, a flag is set in C:\sikulix to indicate that a default profile pic is set. This is to make sure subsequent runs don’t set profile picture again until a special profile pic starting with DP is found.   + If a profile picture with DP prefix exists, it is set, moved to archive folder, and a flag is set in C:\sikulix folder to indicate that a special DP is set. Next run would look at this and revert the profile pic from special one to the default one. (Assumption: Usually any special profile pic that you set is valid for a day)   **Fille read by “Read CSV file” component:** C:\sikulix\input\_data.csv  CSV file layout:    Logic for Greetings:   * Code filters the data for current date (Day-Month portion) AND (SendMsg = Y OR SendImage = Y) * Combines Message and To field to form the greeting message * Sometimes you just want to send an image without an accompanying message. In that case, SendMsg = N and SendImage = Y would take care of that scenario. * Last column can have multiple people or groups delimited by “|” * Similar to profile pic, pictures for an occasion are looked for in the folder:   **<system\_default\_pictures\_folder>** that you supplied in POST input **+** **“\”** + **4 digit year from current date**.  Example: C:\Users\srk\Pictures\2020   * Picture for an Occasion should be saved with prefix <Occasion>. Example: Diwali2020.png * If Occasion if Birthday or Anniversary, code adds Emojis too. 😊 * Idea is to spend time once per year, may be during Christmas holidays, and configure the data for input\_data.csv and let the process take care of sending greetings.   Other logic in Mule flow:   * It creates a log file. Something like for different scenarios. * Sending Greetings & Setting profile pic   \*\*\*\*\* Run Date: 20201117  \*\*\*\*\* Image found for image type: SELF:SELF.jpg  \*\*\*\*\* Choosing this image for update: SELF.jpg  \*\*\*\*\* Whatsapp opened  \*\*\*\*\*  \*\*\*\*\* Sending Greetings|John!|Birthday|Vr2|Send Image:N  \*\*\*\*\*  \*\*\*\*\* Setting Profile Picture|C:\Users\srk\Pictures\2020|SELF.jpg  \*\*\*\*\* WhatsApp update done   * Setting profile pic with special one:   \*\*\*\*\* Run Date: 20201118  \*\*\*\*\* Image found for image type: DP:DP-Nov-18.png  \*\*\*\*\* Choosing this image for update: DP-Nov-18.png  \*\*\*\*\* Whatsapp opened  \*\*\*\*\* No data for greetings  \*\*\*\*\*  \*\*\*\*\* Setting Profile Picture|C:\Users\srk\Pictures\2020|DP-Nov-18.png  \*\*\*\*\* WhatsApp update done   * No data:   \*\*\*\*\* Run Date: 20201113  \*\*\*\*\* No Data To Process   * Emails the same log based on how you configured the EMAIL connector in the flow * Creates a DONE file **C:\sikulix\whatsapp\_update\_done.txt** |
| SleepAfterAllDone.bat | @ECHO OFF  IF EXIST %SIKULIX\_HOME%\whatsapp\_update\_done.txt (  DEL %SIKULIX\_HOME%\whatsapp\_update\_done.txt  powershell.exe -encodedCommand   )  This, you would set up as a task in Windows task scheduler to run every 5 minutes or so. It will put your Windows system to sleep once the file **whatsapp\_update\_done.txt** is found  Please note, this worked on my system which is Windows 10, 64 bit and there are a couple of pre-requisite tasks needed to get this working which are listed in GitHub README.md  Also, you may need Powershell exe rights. If you run into an issue, please google. |
| Windows task scheduler – Video on how to create tasks | It is .mp4 file. |
| Disclaimer | 1. Sikulix is pretty much an image drive workflow. If it finds image, it is all good. But again, images and their size, resolution, and other properties may vary based on the system. 2. If you run into an issue where image is not found, just capture that image on your system, and replace the image in C:\sikulix\sikuli\_images folder with yours. 3. Also, in some cases, if it does not click on the screen where it is supposed to, then you will have to calculate the offset which you can do using Sikulix Desktop IDE. I used offsets in Java method for some images. Video below shows how to calculate offset for a image if needed.      1. Sikuli code has wait period between clicks. Please be patient and try not move the cursor if you see it does not seem to do anything. I would say give it at least 30 seconds to a minute before you decide to switch windows and check what’s going on. Logs that can be checked:  * C:\sikulix\sikuli.log * %MULE\_HOME%\logs\mule\_ee.log |
| Sikulix links | <http://sikulix.com/>  <https://launchpad.net/sikuli/+download>  <http://sikulix.com/quickstart/>  <https://sikulix-2014.readthedocs.io/en/latest/toc.html> |