API REPORT

`getSMSAccountPhoneNumbers()`

How to do it?

1. Ensure using the right profile by running *('twilio login $ACCOUNT_SID --auth-token=$AUTH -p local_twilio_creds -f'*) to log into the account which you want to list the numbers of. 2. Return list of bots associated with that number with *('twilio phone-numbers:list'*) and the data returned is the SID and Number.

```
SID Phone Number Friendly Name
PN0766dda7f73f63ca5879460503bd64c3 +447447735023 447447735023
```

How long to implement?

To just purely list all the numbers it will take maximum one day.

However, if we want to list the numbers and the bot associated with them, we would either have to:

1. Set the Friendly Name to the bot name. This would require the clinician to purchase the phone number via our interface so we can set the friendly name (*twilio phone numbers:update $NUMBER_SID –friendly-name=friendly_name*)when generating the assistant. (Possible: would require maximum 2 days)
2. Go through each phone number, look at the messaging webhook, extract the assistant UID and match an assistant to a number that way. (Twilio allows you to update webhooks via CLI but need to research deeper if we can list phone-number webhooks.)

Phone Number Webhook Format:

https://channels.autopilot.twilio.com/v1/$ACCOUNT_SID/$ASSISTANT_SID /twilio messaging?TargetTask=greeting

`getDeployedBots()`

How to do it?

1. Ensure using the right profile by running *('twilio login $ACCOUNT_SID --auth-token=$AUTH -p local_twilio_creds -f'*) to log into the account which you want to list the numbers of. 2. Return list of assistants on Twilio Autopilot by running (*'twilio autopilot:list'*)

```
SID Unique Name Friendly Name

UAdaa68cee37e85d4e644fb455e01105c8 MSK-CLI-2 Test

UA45894f3bb605740c29264f4863947560 HelloWorld Basic starter template

UA6ee4ab2532d9dcc958e0dc65ee7d0a65 MSK-T28
```

How long to implement?

If we want to list bots that at live or not, we would have to either:

1. When bots are suspended, we delete them from autopilot so it only lists currently running

bots. (2 days)

2. Not ideal but we have our own backend/database which keeps track of which bots are live and the phone-numbers they are associated with.

Twilio CLI does not give any info of when the bot was last updated. We would either have to store that data ourselves or download the schema.json for a bot (defines all the tasks for a bot) and read the "modelBuild" at the bottom of the .json file which tells you when the bot was built.

`_uploadBot()`

How to do it?

1. Run combined js version of diagram_to_tasks and tasks_to_twilio to first get the twilio environment initialised.
2. Then assign the autopilot webhook (*'https://channels.autopilot.twilio.com/v1/$ACCOUNT_SID/$ASSISTANT_SID /twilio messaging?TargetTask=greeting')* to the given phone number using (*twilio phone numbers:update –PHONE-SID*)
3. Set the Friendly Name of the phone-number to the name of the bot.

How long to implement?

Would require converting the python files to JS and adding the updating phone-number webhook feature. It would take maximum 1 week.

When it comes to uploadAndOverwiteBot, if we go down the route where we use "friendly-name" to associate an assistant with a bot rather than a database, then after changing the friendly-name of the new bot, the webhook of the number would have be changed to the bot that just got uploaded.

With uploadNewBot, similarly we can just see if friendly-name is already set and if not, then we add the new bot or return null.

`removeBot()`

How to do it?

1. Look for bot with list of assistants in Twilio Autopilot by running (*'twilio autopilot:list'*) 2. If specified bot UID/Name within list then remove via ('twilio autopilit:delete $ASSISTANT SID)
3. Clear webhook for sms-url for phone-number
4. Clear the serverless Twilio Function for associated bot.

How long to implement?

Once we free the phone-number, there is no way (need to investigate further) to see which phone numbers have empty sms-url's vs ones that are filled. This may mean we have our own database to keep track of numbers or we can set the "friendly-name" of the number to "free".

We would have to ensure that no one uploads a bot with the name “free”
Another issue is that Twilio keeps Autopilot bots for 30-days, giving you a chance to restore them if you make a mistake. This means that recently deleted bots cannot be reuploaded with the same name for 30 days.

`renameBot()`

How to do it?

1. Take current bot and export via *(‘twilio autopilot:export $ASSISTANT-SID’*) 2. Open the exported bot scheme and change the uniqueName to the new name of the bot. 3. Call uploadBot() to upload the bot with a new name.
4. Update any phone-number webhooks that are associated with the previous name.

How long to implement?

Twilio does not allow you to rename the bot in the exported schema. If you do so, it results in Twilio looking for a bot that does not exist, so you are forced to create the same bot but with the new name. This would take maximum 3 days to implement.

`suspendBot()`

How to do it?

1. Suspending the bot could mean we unlink the phone-number with the assistant by running (*‘twilio phone-number:update’*)

How long to implement?

This would require checking if the webhook has been wiped. Not entirely sure if that can be done with CLI but if it can this would at most take a day to implement or else we would require our own database to keep track. If able to be done with CLI would take maximum 2 days.

`resumeBot()`

How to do it?

1. Resuming the bot could mean we take the previously assigned phone-number and set the webhook to the twilio autopilot assistant we want to resume. (*‘twilio phone number:update’*)

How long to implement?

This would require updating the webhook which can be done with the Twilio CLI but there is no way of confirming if the phone-number webhook has been successfully changed and with a system that requires certainty it may require us to have our own database. If able to be done with CLI would take maximum 2 days.

`changeBotPhoneNumber()`

How to do it?

1. List the phone numbers (*'twilio phone-number:update'*)
2. If the number exists then, update the webhook to the relevant assistant as well as change the friendly-name to the new bot.

How long to implement?

This can be implemented within a day but again there is no certainty that the change was successful or not. Twilio may not return an error but another function could overwrite the phone-number webhook and we would lose track of which bot is associated with which number due to stale data.

Conclusion

Core Required API Calls:
getDeployedBots()

_uploadBot()

removeBot()

changeBotPhoneNumber()

The only bit of uncertainty is whether we can view the current webhook assigned to a bot. If this is possible then it will reduce the chance of us needing a backend.