Resilient Function - Twilio Send SMS

Table of Contents

This Resilient Function package can be used to send an SMS via Twilio from a workflow using the Functions feature of the Resilient Circuits integration framework.

Prerequisites:

```
Resilient version 33 or later
resilient_circuits version 32 or later
```

```
twilio version 6.21.0 or later
```

app.config settings

This package requires that it is installed on an Integrations server (preferably a RHEL platform) and that the resilient-circuits application is running. Install this package with `pip install`, or `python setup.py install`. To set the config values in the app.config file with a new resilient instance run `resilient-circuits config -c` or use `config -u` to append to an existing app.config file.

Config values example: The parameters twilio_account_sid and twilio_auth_token are as defined in your Twilio account. The parameter twilio_src_address should be set to the Twilio number you wish to use.

```
[fn_twilio_send_sms]
twilio_account_sid=
twilio_auth_token=
# This is the number that will originate the SMS and must be an active
SMS phone number on your Twilio Account
# The format should be as per the Twilio console properties for your
number, e.g. +1234567890
twilio_src_address=
```

Run with: `resilient-circuits run`.

twilio_send_sms

Function Inputs

Function Name	Type	Required	Example	Info
twilio_sms_destination	String	Yes	"+353861234567,+1234567"	A comma delimited (CSV) list of destination numbers in international format.
twilio_sms_message	String	No	"An incident has been created!"	The message you wish to send. If left empty a message will be send with the Incident title and priority.

Function Output

```
{
  'twilio_status': [
  {
    'phone_number': u'+1234',
    'error_message': "Unable to create record: The 'To' number  is not
a valid phone number.",
    'success': False
    },{
    'phone_number': u'+353834463164',
    'success': True
    }
  ],
  'inputs': {
    'twilio_sms_message': u'An incident test (2096) with Low priority
may require your attention',
    'twilio_sms_destination': u'1234,353834463164'
    }
  , 'success': True
}
```

Pre-Process Script

The following is an example of setup of each parameter using a simple workflow pre-processing script. The message can be customised to suit your own use case.

```
inputs.twilio_sms_destination = rule.properties.twilio_sms_destination
inputs.twilio_sms_message = 'An incident ' + incident.name + ' (' +
`incident.id` + ') with ' + incident.severity_code + ' priority may
require your attention'
```

Post-Process Script

The results returned to Resilient can be used to determine the status for each destination. Below is an example post process script which adds a note for each destination.

```
import java.util.Date as Date

for entry in results["twilio_status"]:
  if(entry.success == True):
    note_text = u"""<b>Twilio SMS Message:</b> {0}
            </br><b>sent to:</b>
{1}""".format(results.inputs.twilio_sms_message,
                                        entry.phone_number)

    incident.addNote(helper.createRichText(note_text))
  else:
    note_text = u"""<b>Unable to send Twilio SMS Message:</b> {0}
            </br><b> to:</b> {1}
({2})""".format(results.inputs.twilio_sms_message,
                                        entry.phone_number,
entry.error_message)

    incident.addNote(helper.createRichText(note_text))
```

```
    row = incident.addRow("twilio_sms_log")
    row['row_created'] = str(Date())
    row['status'] = entry.get("status") or "Failed"
    row['message_id'] = entry.get("messaging_service_sid")
    row['date_created'] = entry.get("date_created_ts")
    row['msg_body'] = entry.get("message_body") or
  entry.get("error_message")
    row['phone_number'] = entry.get("phone_number")
    row["direction"] = entry.get("direction")
```

Twilio: Get Responses

Function Inputs

Function Name	Type	Required	Example	Info
twilio_phone_number	String	Yes	"+353861234567"	A destination number to filter the responses returned
twilio_date_sent	String	No	"2020−01−25 14:59:51+00:00"	The date which responses are returned. If blank, all responses are returned
twilio_date_sent_ts	Number	No	15438583123	Alternative to twilio_after_date. Timestamp of a date which responses are returned. If blank, all responses are returned
twilio_wait_timeout	String	No	"120s"	Timeframe to wait if no responses are available. Timeframe can be seconds (ex. 30s), minutes (ex. 10m), hours (ex. 1h), days (ex. 1d). Default is 120s.

Function Output

```
{
  'version': '1.0',
  'success': True,
  'reason': None,
  'content': [
   {
     'phone_number': '+19788354530',
     'messaging_service_sid': 'SMe9584a564764db7c4d24f612d6928b18',
     'date_created': '2020−01−25 18:05:47+00:00',
     'direction': 'inbound',
     'message_body': 'Acknowledged',
```

```
        'status': 'received',
        'error_message': None
      }
    ],
    'raw': '[{"phone_number": "+19788354530", "messaging_service_sid":
"SMe9584a564764db7c4d24f612d6928b18", "date_created": "2020-01-25
18:05:47+00:00", "direction": "inbound", "message_body":
"Acknowledged", "status": "received", "error_message": null}]',
    'inputs': {
      'twilio_wait_timeout': '60s',
      'twilio_after_date': '2020-01-25 14:59:51+00:00',
      'twilio_phone_number': '+19788354530'
    },
    'metrics': {
      'version': '1.0',
      'package': 'unknown',
      'package_version': 'unknown',
      'host': 'Marks-MacBook-Pro.local',
      'execution_time_ms': 941,
      'timestamp': '2020-01-29 13:01:09'
    }
  }
}
```

Pre-Process Script

The following is an example of setup of each parameter using a simple workflow pre-processing script. The message can be customised to suit your own use case.

From a datatable row:

```
inputs.twilio_phone_number = row['phone_number']
inputs.twilio_date_sent = row['date_created']
```

Post-Process Script

Below is an example post process script which adds a row to the datatable for each response.

```
import java.util.Date as Date

if results.success:
  for entry in results.content:
    row = incident.addRow("twilio_sms_log")
    row['row_created'] = str(Date())
    row['status'] = entry.get("status")
    row['message_id'] = entry.get("messaging_service_sid")
    row['date_created'] = entry.get("date_created_ts")
    row['msg_body'] = entry.get("message_body")
    row['phone_number'] = entry.get("phone_number")
    row["direction"] = entry.get("direction")
else:
  incident.addNote(u"Twilio Received Messages failed:
{}".format(results.reason))
```

Example: Twilio Receive Messages

Get all SMS messages based on destination phone number and date sent information.

Function Inputs

Function Name	Type	Required	Example	Info
twilio_phone_number	String	Yes	"+353861234567"	A destination number to filter the responses returned
twilio_date_sent	String	No	"2020-01-25 14:59:51+00:00"	The date which responses are returned. If blank, all responses are returned
twilio_date_sent_ts	Number	No	15438583123	Alternative to twilio_after_date. Timestamp of a date which responses are returned. If blank, all responses are returned
twilio_wait_timeout	String	No	"120s"	Timeframe to wait if no responses are available. Timeframe can be seconds (ex. 30s), minutes (ex. 10m), hours (ex. 1h), days (ex. 1d). Default is 120s.

Function Output

```
{
  'version': '1.0',
  'success': True,
  'reason': None,
  'content': [
    {
      'phone_number': '+19788354530',
      'messaging_service_sid': 'SMe9584a564764db7c4d24f612d6928b18',
      'date_created': '2020-01-25 18:05:47+00:00',
      'direction': 'inbound',
      'message_body': 'Acknowledged',
      'status': 'received',
      'error_message': None
    }
  ],
  'raw': '[{"phone_number": "+19788354530", "messaging_service_sid":
"SMe9584a564764db7c4d24f612d6928b18", "date_created": "2020-01-25
18:05:47+00:00", "direction": "inbound", "message_body":
"Acknowledged", "status": "received", "error_message": null}]',
```

```
  'inputs': {
    'twilio_wait_timeout': '60s',
    'twilio_after_date': '2020-01-25 14:59:51+00:00',
    'twilio_phone_number': '+19788354530'
  },
  'metrics': {
    'version': '1.0',
    'package': 'unknown',
    'package_version': 'unknown',
    'host': 'Marks-MacBook-Pro.local',
    'execution_time_ms': 941,
    'timestamp': '2020-01-29 13:01:09'
  }
}
```

Pre-Process Script

The following is an example of setup of each parameter using a simple workflow pre-processing script. The message can be customised to suit your own use case.

From a datatable row:

```
inputs.twilio_phone_number = rule.properties.twilio_sms_destination
inputs.twilio_date_sent_ts = rule.properties.twilio_after_date_ts
inputs.twilio_wait_timeout = rule.properties.twilio_wait_timeout
```

Post-Process Script

Below is an example post process script which adds a row to the datatable for each response.

```
import java.util.Date as Date

if results.success:
  for entry in results.content:
    row = incident.addRow("twilio_sms_log")
    row['row_created'] = str(Date())
    row['status'] = entry.get("status")
    row['message_id'] = entry.get("messaging_service_sid")
    row['date_created'] = entry.get("date_created_ts")
    row['msg_body'] = entry.get("message_body")
    row['phone_number'] = entry.get("phone_number")
    row["direction"] = entry.get("direction")
else:
  incident.addNote(u"Twilio Received Messages failed:
{}".format(results.reason))
```

Rules

Rule Name	Object Type	Workflow Triggered
Example: Send Twilio SMS	Incident	Example: Twilio Send SMS
Example: Twilio Receive Messages	Incident	Example: Twilio Receive Messages

Rule Name	Object Type	Workflow Triggered
Twilio: Get Responses	Datatable	Twilio: Get Responses