

# Skywire<sup>®</sup> LTE CAT-M1 Socket Dial Application Note

NimbeLink Corp

Updated: March 2018



© NimbeLink Corp. 2018. All rights reserved.

NimbeLink Corp. provides this documentation in support of its products for the internal use of its current and prospective customers. The publication of this document does not create any other right or license in any party to use any content contained in or referred to in this document and any modification or redistribution of this document is not permitted.

While efforts are made to ensure accuracy, typographical and other errors may exist in this document. NimbeLink reserves the right to modify or discontinue its products and to modify this and any other product documentation at any time.

All NimbeLink products are sold subject to its published Terms and Conditions, subject to any separate terms agreed with its customers. No warranty of any type is extended by publication of this documentation, including, but not limited to, implied warranties of merchantability, fitness for a particular purpose and non-infringement.

Skywire and NimbeLink are registered trademarks of NimbeLink Corp. All other trademarks appearing in the document are the property of their respective owners.

# Table of Contents

<b>Table of Contents</b>	<b>2</b>
<b>Introduction</b>	<b>3</b>
Orderable Part Numbers	3
Prerequisites	3
<b>Socket Dial Procedure</b>	<b>4</b>
Overview	4
Firmware Check	4
Socket Setup	4
Initiate Socket Dial	4
Send Data via HTTP	5
Shutdown Socket Connection	5
<b>Working Example 1: Dweet.io</b>	<b>6</b>
Get IMEI of Modem	6
Socket Setup	6
Initiate the Socket Dial	6
Send Data via HTTP	7
Receive Data via HTTP	7
<b>Working Example 2: Exosite</b>	<b>8</b>
Overview	8
Get CIK of Modem or Device from Exosite	8
Initiate Socket Dial	9
Send Data via HTTP	9
Receive Data via HTTP	11
<b>Troubleshooting</b>	<b>13</b>
Serial Clients	13
ERROR When Issuing AT+SQNSD	13

# 1. Introduction

## 1.1 Orderable Part Numbers

Orderable Device	Description	Carrier	Network Type
NL-M1DK	Skywire M1 Development Kit	Any	Any
NL-SW-LTE-SVZM20-ES	LTE CAT M1, Engineering Sample	Verizon	LTE
NL-SW-LTE-SVZM20	LTE CAT M1	Verizon	LTE

## 1.2 Prerequisites



**This document assumes you have completed the initial setup of your modem and development kit using the Skywire® M1 Development Kit User Manual:**

**<http://nimbelink.com/skywire-m1dk/user-manual/>**

**If you have not completed those steps, refer to the link above and complete the modem setup before proceeding.**

## 2. Socket Dial Procedure

### 2.1 Overview

Using socket dials is useful for uploading information to or downloading information from a website or database via HTTP commands. Below is an example for connecting to a server, uploading data, downloading data, and disconnecting from the server.

### 2.2 Firmware Check

Socket dial functionality is available on firmware 29318 and newer. To check your firmware, type the command:

```
ATI1
```

followed by the Enter key, and the modem should respond with something similar to:

```
UE5.0.0.0-xxxxx
```

```
LR5.0.0.0-xxxxx
```

```
OK
```

Where xxxxx is the firmware version number. If your Skywire® returns a firmware version older than 29318, please consult the firmware update application note here:

<http://nimbelink.com/skywire-4g-lte-cat-m1/#documents>

### 2.3 Socket Setup

To begin, we need to setup the socket connection. Type the following command into your terminal program:

```
AT+SQNSCFG=3,3,300,90,600,50
```

followed by the Enter key, and the terminal should respond with:

```
OK
```

### 2.4 Initiate Socket Dial

In the terminal program, type the following command:

```
AT+SQNSD=3,0,80,"example.com"
```

where 1 is the socket we are using, 0 is the transmission protocol to use (0 - TCP, 1 - UDP), 80 is the port number you'd like to connect to, and example is the hostname or IP address you'd like to connect to, followed by the enter key. After a short delay, the terminal program should respond with:

```
CONNECT
```

Note: The TCP port (third parameter) that is typically used is port 80, which is the TCP Port for HTTP. Depending on your application, you may need a different port.

## 2.5 Send Data via HTTP

In order to send data to the webserver, you can use the HTTP POST command. The syntax of the POST command is as follows:

```
POST /test/demo_form.asp HTTP/1.1
```

POST is the HTTP command being issued, /test/demo\_form.asp is the endpoint of the server, and HTTP/1.1 is the HTTP version you will be using.

Note: For this step, you will not be able to see the text you are typing. To ensure that the data is accurate, it is recommended to copy and paste in the data. However, this is not necessary.

In the terminal program, enter the POST data you would like to send according to the above syntax. Make sure that the data is formatted to how your server is expecting it. Once you type the data, press the combination CTRL+M and CTRL+J **twice**. This will enter the two new line characters necessary to signal to the server that data transmission is complete. After a short delay, the terminal program should display:

```
HTTP/1.1 2xx OK
```

```
[text response from server]
```

```
OK
```

where 2xx is the successful response code of the HTTP server. 200 is the general HTTP/1.1 response of OK, meaning that the response was received properly. There are derivations of the 2xx response, however. If you receive an unexpected value, please consult the HTTP/1.1 Protocol Documentation at the following URL:

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

## 2.6 Shutdown Socket Connection

In the terminal program, type the following command:

```
+++
```

followed by the Enter key, and the terminal should respond with:

```
OK
```

This command (three plus signs) is the escape command to suspend the data connection in the socket connection.

To shutdown the socket connection, type the following command:

```
AT+SQNSH=3
```

followed by the Enter key, and the terminal should respond with:

```
OK
```

## 3. Working Example 1: Dweet.io

This section will provide a working example of sending and receiving data via an HTTP transfer using a Skywire NL-SW-LTE-SVZM20-ES modem using a 4G Verizon LTE CAT M1 SIM.

dweet.io is a lightweight messaging service specifically designed for IoT (Internet of Things) devices. In addition to being lightweight, dweet.io does not require an account to get up and running. At [www.dweet.io](http://www.dweet.io), they have an excellent “hello world” example we will be using.

### 3.1 Get IMEI of Modem

dweet.io requires a unique device name in order to send and receive data. For this, we will be using the IMEI of our modem, which is unique to our modem. To display the IMEI, type the following command into the terminal program:

```
AT+CGSN
```

followed by the Enter key, and the terminal will respond with something similar to:

```
353238060023699
```

```
OK
```

The number should be identical to the IMEI printed on the top of the modem label.

### 3.2 Socket Setup

Type the following command into your terminal program:

```
AT+SQNSCFG=3,3,300,90,600,50
```

followed by the Enter key, and the terminal should respond with:

```
OK
```

### 3.3 Initiate the Socket Dial

Using the syntax described in Step 2.4, type the following command into the terminal program:

```
AT+SQNSD=3,0,80,"dweet.io"
```

followed by the Enter key, and the terminal program should respond with

```
CONNECT
```

In this case, 3 is the socket we are using on the Skywire modem, 0 means we are using the TCP transmission protocol, 80 is the TCP port we are using (TCP port 80 is used for HTTP), and dweet.io is the hostname we are connecting to.

## 3.4 Send Data via HTTP

Using the syntax described in Step 2.5, enter the following command into the terminal program:

```
POST /dweet/for/353238060023699?hello=world HTTP/1.1
```

Note: 353238060023699 is the example IMEI that we used in Step 3.2.1. Replace this with your unique IMEI.

Press the combination CTRL+M and CTRL+J **twice**, and after a short delay the terminal program should respond with something similar to:

```
HTTP/1.1 200 OK
```

```
Access-Control-Allow-Origin: *
```

```
Content-Type: application/json
```

```
Content-Length: 203
```

```
Date: Tue, 15 Mar 2016 21:50:59 GMT
```

```
Connection: keep-alive
```

```
{"this": "succeeded", "by": "dweeting", "the": "dweet", "with": {"thing": "353238060023699", "created": "2016-03-15T21:50:59.822Z", "content": {"hello": "world"}}, "transaction": "debcf53b-1494-4a37-852e-c4d5278f79e6"}}
```

Notice that our IMEI appears next to "thing". By going to the website:

<https://dweet.io/get/latest/dweet/for/353238060023699>

and replacing 353238060023699 with your IMEI, you will see:

```
{this: "succeeded",by: "getting",the: "dweets",with: [{thing: "353238060023699",created: "2016-03-15T21:51:25.173Z",content: {hello: "world"}}]}
```

This text is the same as the text at the bottom of the terminal response, indicating a successful HTTP transaction.

## 3.5 Receive Data via HTTP

Using the syntax described in step 2.5, enter the following command into the terminal program:

```
GET /get/latest/dweet/for/353238060023699 HTTP/1.1
```

Note: 353238060023699 is the example IMEI that we used in Step 3.2.1. Replace this with your unique IMEI.

Press CTRL+M and CTRL+J **twice**, and after a short delay the terminal program should respond with something similar to:

```
HTTP/1.1 200 OK
```

Access-Control-Allow-Origin: \*  
Content-Type: application/json  
Content-Length: 152  
Date: Tue, 15 Mar 2016 21:58:13 GMT  
Connection: keep-alive

```
{"this": "succeeded", "by": "getting", "the": "dweet", "with": [{"thing": "353238060023699", "created": "2016-03-15T21:51:25.173Z", "content": {"hello": "world"}}]}
```

Notice that our IMEI appears next to "thing". This indicates that we successfully received the HTTP information.

## 4. Working Example 2: Exosite

### 4.1 Overview

This section will provide a working example of sending and receiving data via an HTTP transfer using a Skywire NL-SW-LTE-SVZM20-ES modem using a 4G Verizon LTE CAT M1 SIM.

Exosite is a management service that manages sensor and device data. Exosite allows you to set up a web dashboard that you can send data to via HTTP for M2M applications.

This example assumes that you have registered your device with Exosite. If you have not, please consult the Exosite documentation located at [www.exosite.com/support](http://www.exosite.com/support) and [docs.exosite.com](http://docs.exosite.com).

To post to Exosite, we have created a Dashboard specifically for NimbeLink customers to walk through the complete process of sending and receiving information from Exosite. The NimbeLink Exosite Example Dashboard is located at the following URL:

<https://nimbelink.exosite.com/views/2334257135/2984653781>

and contains three values that we can send and read: Key1, Key2, and Key3. Key1 is of type "String", Key2 is of type "integer", and Key3 is of type "float" (decimal). When we send information in Section 4.4, we must send information of the correct type.

In this example, we will be changing those three values to something we specify, in order to verify that communication is working as expected.

### 4.2 Get CIK of Modem or Device from Exosite

Exosite requires a unique device name in order to send and receive data. This unique name—called a CIK—is created by Exosite when you register your device with them. For NimbeLink's example page, we will be using the following CIK:



```
7cafee22ab8628b2838187a7774f5e4b3f05f877
```

Note: This CIK is unique to this example. When you register your device, you will have a different CIK.

### 4.3 Initiate Socket Dial

Using the syntax described in Section 2.3, type the following command into the terminal program:

```
AT+SQNSD=3,0,80,"m2.exosite.com"
```

followed by the enter key, and the terminal program should respond with

```
CONNECT
```

In this case, **3** is the socket we are using on the Skywire modem, **0** means we are using the TCP transmission protocol, **80** is the TCP port we are using (TCP port 80 is used for HTTP), and **m2.exosite.com** is the hostname we are connecting to.

### 4.4 Send Data via HTTP

Note: The following command must be entered correctly and in a specific order. As such, it is recommended to copy and paste the commands instead of typing them.

Note: Information entered into the terminal program at this point will not be visible on the screen.

Below is the entire command sequence we will be sending on the modem.

```
POST /onep:v1/stack/alias HTTP/1.1
Host: m2.exosite.com
X-Exosite-CIK: 7cafee22ab8628b2838187a7774f5e4b3f05f877
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Accept: application/xhtml+xml
Content-Length: 28
```

```
Key1=asdf&Key2=100&Key3=11.1
```

Below is the exact process needed to successfully send the data.

Enter the following command into the terminal program:

```
POST /onep:v1/stack/alias HTTP/1.1
```

followed by CTRL+M and CTRL+J on the keyboard.

Enter the following command into the terminal program:

```
Host: m2.exosite.com
```

followed by CTRL+M and CTRL+J on the keyboard.

Enter the following command into the terminal program:

```
X-Exosite-CIK: 7cafee22ab8628b2838187a7774f5e4b3f05f877
```

followed by CTRL+M and CTRL+J on the keyboard. Note that the example CIK is present in this line. This is where you would put your unique CIK in.

Enter the following command into the terminal program:

```
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

followed by CTRL+M and CTRL+J on the keyboard.

Enter the following command into the keyboard:

```
Accept: application/xhtml+xml
```

followed by CTRL+M and CTRL+J on the keyboard.

Enter the following command into the terminal program:

```
Content-Length: 28
```

followed by CTRL+M and CTRL+J **twice** on the keyboard. Typing CTRL+M and CTRL+J twice will put the necessary blank line in that the HTTP server is expecting.

Note: The value after **Content-Length:** must be equal to the number of characters that you are sending on with the following command. In this case, we have 28 characters.

Finally, enter the following command into the terminal program:

```
Key1=asdf&Key2=100&Key3=11.1
```

where `asdf` is a string of characters, `100` is an integer number, and `11.1` is a decimal number. There is no need to type anything additional to get the data sent. You may want to change these values while keeping the same type to see your specific changes on the Exosite page.

The terminal program should respond with the following:

```
HTTP/1.1 204 No Content
```

```
Date: [today's date and time]
```

```
Content-Length: 0
```

```
Server: nginx
```

If you visit the NimbeLink Exosite Example Dashboard at the following URL:


<https://nimbelink.exosite.com/views/2334257135/2984653781>

you will see that your values have been updated:

Data source List			
Name▲	Value	Units	Last Reported Time
Key1	asdf	String	17:04:17 Mar 15, 2016
Key2	100	integer	17:04:17 Mar 15, 2016
Key3	11.1	float	17:04:17 Mar 15, 2016

**ON-OFF test**

SwitchState



Click 'ON' to write a 0 to your data source.  
Click 'OFF' to write a 1 to your data source.

## 4.5 Receive Data via HTTP

Note: The following command must be entered correctly and in a specific order. As such, it is recommended to copy and paste the commands instead of typing them.

Note: Information entered into the terminal program at this point will not be visible on the screen.

Below is the entire command sequence we will be sending on the modem.

```
GET /onep:v1/stack/alias?Key1&Key2&Key3 HTTP/1.1
Accept: application/x-www-form-urlencoded; charset=utf-8
Accept-Encoding: gzip, deflate
Host: m2.exosite.com
Connection: Close
X-Exosite-CIK: 7cafee22ab8628b2838187a7774f5e4b3f05f877
```

Below is the exact process needed to successfully receive data from the Exosite Dashboard.

Enter the following command into the terminal program:

```
GET /onep:v1/stack/alias?Key1&Key2&Key3 HTTP/1.1
```

followed by CTRL+M and CTRL+J on the keyboard.

Enter the following command into the terminal program:

```
Accept: application/x-www-form-urlencoded; charset=utf-8
```

followed by CTRL+M and CTRL+J on the keyboard.

Enter the following command into the terminal program:

```
Accept-Encoding: gzip, deflate
```

followed by CTRL+M and CTRL+J on the keyboard.

Enter the following command into the terminal program:

```
Host: m2.exosite.com
```

followed by CTRL+M and CTRL+J on the keyboard.

Enter the following command into the keyboard:

```
Connection: Close
```

followed by CTRL+M and CTRL+J on the keyboard.

Finally, enter the following command into the terminal program:

```
X-Exosite-CIK: 7cafee22ab8628b2838187a7774f5e4b3f05f877
```

followed by CTRL+M and CTRL+J **three times** on the keyboard. Typing CTRL+M and CTRL+J three times will put the necessary blank lines in that the HTTP server is expecting. Note that the example CIK is present in this line. This is where you would put your unique CIK in.

The terminal program should respond with the following:

```
HTTP/1.1 200 OK
```

```
Date: Tue, 15 Mar 2016 22:08:05 GMT
```

```
Content-Length: 28
```

```
Connection: close
```

```
Server: nginx
```

```
Key3=11.1&Key2=100&Key1=asdf
```

# 5. Troubleshooting

## 5.1 Serial Clients

If you are manually inputting the above sequences into a serial program such as PuTTY or TeraTerm, some serial clients may interpret the "Carriage Return" (CTRL-M) and "Line Feed" (CTRL-J) characters differently. For instance, if you are having issues sending information using a HTTP POST, try sending four "Line Feed" characters instead of the sequence of "Carriage Return" and "Line Feed".

So, in Section 2.3.5, instead of sending:

```
"CTRL-M" "CTRL-J" "CTRL-M" "CTRL-J"
```

send:

```
"CTRL-J" "CTRL-J" "CTRL-J" "CTRL-J"
```

## 5.2 ERROR When Issuing AT+SQNSD

If you receive an ERROR when trying to open the socket with AT+SQNSD, reset the Skywire:

```
AT^RESET
```

and try the socket dial command without the:

```
AT+SQNSCFG
```

command. The AT+SQNSCFG command is saved to non-volatile memory (NVM), so it does not need to be issued each time.