

# Skywire™ Development Kit

## Socket Dial Example

NimbeLink Corp

Updated: April 2017



© NimbeLink Corp. 2017. 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.

XBee is a registered trademark of Digi International, Inc

NimbeLink is a registered trademark, and Skywire is a trademark, of NimbeLink Corp. All trademarks, service marks and similar designations referenced in this document are the property of their respective owners.

# Table of Contents

## [1. Introduction](#)

### [1.1. Prerequisites](#)

## [2. Socket Dial](#)

### [2.1. Overview](#)

### [2.2 Setup APN \(If Applicable\)](#)

### [2.2. Configure PDP Context \(If Applicable\)](#)

### [2.3. Activate Skywire™ Socket](#)

### [2.4. Initiate Socket Dial](#)

### [2.5. Send Data via HTTP](#)

### [2.6. Receive Data via HTTP](#)

### [2.7. Shutdown the Socket Connection](#)

## [3. Working Example 1: dweet.io](#)

### [3.1. Overview](#)

### [3.2. Get IMEI of Modem](#)

### [3.3. Initiate Socket Dial](#)

### [3.4. Send Data via HTTP](#)

### [3.5. Receive Data via HTTP](#)

## [4. Working Example 2: Exosite](#)

### [4.1. Overview](#)

### [4.2. Get CIK of Modem or Device from Exosite](#)

### [4.3. Initiate Socket Dial](#)

### [4.4. Send Data via HTTP](#)

### [4.5. Receive Data via HTTP](#)

# 1. Introduction

## 1.1. Prerequisites



This document assumes you have completed the initial setup of your modem and development kit. If you have not completed those steps, refer to the Skywire™ Development Kit User Manual and complete the modem setup before proceeding. If applicable, the modem must already be provisioned. See the Development Kit User Manual for details on how to accomplish these steps.

Note: If your Skywire™ modem is not listed here, please visit its respective page on [www.nimbelink.com](http://www.nimbelink.com) for the socket dial procedure.

## 2. Socket Dial

### 2.1. Overview

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

### 2.2 Setup APN (If Applicable)

For the following modems:

- NL-SW-GPRS
- NL-SW-HSPA
- NL-SW-HSPAP
- NL-SW-HSPAPG
- NL-SW-HSPAPE
- NL-SW-LTE-TEUG
- NL-SW-LTE-TNAG

type the following command into the terminal program:

```
AT+CGDCONT=1,"IP", "[your apn]"
```

where **[your apn]** is the APN for your carrier, followed by the Enter key.

The terminal should respond with:

```
OK
```

For example, if you are using an AT&T SIM and your APN is:

**isp.cingular**

you would type:

**AT+CGDCONT=1,"IP","isp.cingular"**

followed by the Enter key.

For the following modem:

- NL-SW-LTE-TSVG

type the following command into the terminal program:

**AT+CGDCONT=3,"IP","[your apn]"**

where **[your apn]** is the APN for your carrier, followed by the Enter key.

For example, if your Verizon APN is:

**vzwinternet**

you would type:

**AT+CGDCONT=3,"IP","vzwinternet"**

followed by the Enter key.

## 2.2. Configure PDP Context (If Applicable)

For the following modem:

- NL-SW-LTE-TSVG

type the following command into the terminal program:

**AT#SCFG=3,3,300,90,600,50**

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

**OK**

This setup is required to use PDP Context 3 with a socket dial.

## 2.3. Activate Skywire™ Socket

For the following modems:

- NL-SW-1xRTT-A
- NL-SW-1xRTT-S
- NL-SW-1xRTT-V
- NL-SW-EVDO-A
- NL-SW-EVDO-V
- NL-SW-GPRS

- NL-SW-HSPA
- NL-SW-HSPAP
- NL-SW-HSPAPG
- NL-SW-HSPAPE
- NL-SW-LTE-TEUG
- NL-SW-LTE-TNAG

type the following command into the terminal program:

**AT#SGACT = 1,1**

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

**#SGACT: "www.xxx.yyy.zzz"**

**OK**

where **www.xxx.yyy.zzz** is the IP address of your Skywire™ modem.

For the following modem:

- NL-SW-LTE-TSVG

type the following command into the terminal program:

**AT#SGACT=3,1**

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

**#SGACT: "www.xxx.yyy.zzz"**

**OK**

where **www.xxx.yyy.zzz** is the IP address of your Skywire™ modem.

## 2.4. Initiate Socket Dial

In the terminal program, type the following command:

**AT#SD=1,0,80,"example.com"**

where **1** is the socket you would like to connect with (1 – 6), **0** is the transmission protocol you'd like to use (0 – TCP, 1 – UDP), **80** is the port number you'd like to connect to (1 – 65535), and **example.com** is the host name 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**

and the cursor will be on the next line.

Note: For the socket number, use the socket you set up during Section 2 of the Skywire™ Development Kit User Manual or Section 2.2 above. In general, NL-SW-LTE-TSVG uses socket 3, while all other modems use socket 1.

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 use a different port.

## 2.5. Send Data via HTTP

In order to send data to the website, 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, **/text/demo\_form.asp** is the endpoint on 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 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]**

**NO CARRIER**

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. Receive Data via HTTP

In order to receive data from a website, you must use the HTTP GET command. The syntax of the GET command is similar to the POST command, and is as follows:

**GET /test/demo\_form.asp HTTP/1.1**

where **GET** is the HTTP command being issued, **/text/demo\_form.asp** is the location of the data, 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, the command still works if typed correctly.

In the terminal program, enter the GET data you would like to request 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 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 xxx OK  
[text response from server]  
NO CARRIER
```

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>

## 2.7. Shutdown the Socket Connection

In the terminal program, type the following command:

```
+++
```

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

```
NO CARRIER
```

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

To close the socket, type the following command into the terminal program:

```
AT#SH=x
```

where **x** is the socket opened in the previous steps, followed by the enter key, and the terminal should respond with:

```
OK
```

The data connection and socket connection has now been terminated.

## 3. Working Example 1: dweet.io

### 3.1. Overview

This section will provide a working example of sending and receiving data via an HTTP transfer using a Skywire™ NL-SW-LTE-TSVG modem using a 4G Verizon LTE 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.

### 3.2. 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+GSN**

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

**XXXXXXXXXXXXXXXXXX**

**OK**

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

### 3.3. Initiate Socket Dial

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

**AT#SD=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 2.4, enter the following command into the terminal program:

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

where the string of x's is your IMEI from setup 3.2, followed by the enter key. Press 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: 150  
Date: [today's date]  
Connection: keep-alive
```

```
{"this":"succeeded","by":"dweeting","the":"dweet","with":{"thing":"xxxxxxxxxxxxxxxxx"},"created":"[today's date]","content":{"hello":"world"}}
```

where `xxxxxxxxxxxxxxxxx` is your IMEI. By going to the website <https://dweet.io/get/latest/dweet/for/xxxxxxxxxxxxxxxxx>, replacing the x's at the end with your IMEI, you will see the following text:

```
{"this":"succeeded","by":"getting","the":"dweets","with":[{"thing":"xxxxxxxxxxxxxxxxx"}],"created":"[today's date]","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/xxxxxxxxxxxxxxxxx
```

where the string of x's is your IMEI from setup 3.2, followed by the enter key. Press 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: [today's date]**  
**Connection: keep-alive**

```
{"this":"succeeded","by":"getting","the":"dweets","with":{"thing":"  
xxxxxxxxxxxxxxxxxx ","created":"[today's  
date]","content":{"hello":"world"}}}
```

where xxxxxxxxxxxxxxxx is your IMEI.

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-TSVG modem using a 4G Verizon LTE 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#SD=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+J on the keyboard.

Enter the following command into the terminal program:

**Host: m2.exosite.com**

followed by CTRL+J on the keyboard.

Enter the following command into the terminal program:

**X-Exosite-CIK: 7cafee22ab8628b2838187a7774f5e4b3f05f877**

followed by 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+J on the keyboard.

Enter the following command into the keyboard:

**Accept: application/xhtml+xml**

followed by CTRL+J on the keyboard.

Enter the following command into the terminal program:

**Content-Length: 28**

followed by CTRL+J **twice** on the keyboard. Typing 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=abcd&Key2=123&Key3=45.6**

where **abcd** is a string of characters, **123** is an integer number, and **45.6** 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**

**NO CARRIER**

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	abcd	String	11:41:54 Jun 17, 2015
Key2	123	integer	11:41:54 Jun 17, 2015
Key3	45.6	float	11:41:54 Jun 17, 2015

## 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&Key3HTTP/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+J on the keyboard.

Enter the following command into the terminal program:

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

followed by CTRL+J on the keyboard.

Enter the following command into the terminal program:

**Accept-Encoding: gzip, deflate**

followed by CTRL+J on the keyboard.

Enter the following command into the terminal program:

**Host: m2.exosite.com**

followed by CTRL+J on the keyboard.

Enter the following command into the keyboard:

**Connection: Close**

followed by CTRL+J on the keyboard.

Finally, enter the following command into the terminal program:

**X-Exosite-CIK: 7cafee22ab8628b2838187a7774f5e4b3f05f877**

followed by CTRL+J **three times** on the keyboard. Typing 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 in your unique CIK.

The terminal program should respond with the following:

**HTTP/1.1 200 OK**

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

**Content-Length: 28**

**Connection: close**

**Server: nginx**

**Key3=45.6&Key2=123&Key1=abcd**

**NO CARRIER**