

# Avnet M18Qx LTE IoT API Reference

Contents

[Avnet M18Qx LTE IoT API Reference 1](file:///C:\cygwin64\home\043674\AvnetWNCSDK\Avnet%20M18Qx%20LTE%20IoT%20API%20Guide.docx#_Toc494446319)

[Introduction 5](#_Toc494446320)

[1.1. Definitions and Abbreviations 5](#_Toc494446321)

[1.2. Purpose 5](#_Toc494446322)

[1.3. Sending a request 6](#_Toc494446323)

[1.4. Receiving a response 6](#_Toc494446324)

[1. Requests 7](#_Toc494446325)

[1.1. WWAN 7](#_Toc494446326)

[1.1.1. get\_wwan\_serving\_system\_status 7](#_Toc494446327)

[1.1.2. get\_wwan\_serving\_system\_provider 9](#_Toc494446328)

[1.1.3. get\_wwan\_serving\_system\_radio\_mode 10](#_Toc494446329)

[1.1.4. get\_wwan\_network\_radio\_mode 11](#_Toc494446330)

[1.1.5. set\_wwan\_network\_radio\_mode 12](#_Toc494446331)

[1.1.6. get\_wwan\_radio\_info 12](#_Toc494446332)

[1.1.7. get\_wwan\_band\_capability 13](#_Toc494446333)

[1.1.8. get\_wwan\_band\_info 14](#_Toc494446334)

[1.1.9. get\_wwan\_lte\_ca\_info 15](#_Toc494446335)

[1.1.10.get\_wwan\_serving\_system\_roaming 15](#_Toc494446336)

[1.1.11.get\_wwan\_allow\_data\_roaming 16](#_Toc494446337)

[1.1.12.set\_wwan\_allow\_data\_roaming 16](#_Toc494446338)

[1.1.13.get\_wwan\_network\_time 17](#_Toc494446339)

[1.1.14.get\_wwan\_ipv4\_network\_state 17](#_Toc494446340)

[1.1.15.get\_wwan\_ipv4\_network\_ip 18](#_Toc494446341)

[1.1.16.get\_wwan\_ipv4\_network\_connection\_time 19](#_Toc494446342)

[1.1.17.get\_wwan\_ipv4\_network\_status 19](#_Toc494446343)

[1.1.18.get\_wwan\_apn\_selection\_mode 21](#_Toc494446344)

[1.1.19.get\_wwan\_apn\_selection 21](#_Toc494446345)

[1.1.20.set\_wwan\_apn\_selection 23](#_Toc494446346)

[1.1.21.get\_wwan\_apn\_list 24](#_Toc494446347)

[1.1.22.get\_wwan\_apn\_settings 25](#_Toc494446348)

[1.1.23.set\_wwan\_preferred\_network 27](#_Toc494446349)

[1.1.24.get\_wwan\_preferred\_network\_mode 28](#_Toc494446350)

[1.2. Network 28](#_Toc494446351)

[1.2.1. get\_network\_connection\_status 28](#_Toc494446352)

[1.2.2. get\_network\_connection\_mode 30](#_Toc494446353)

[1.2.3. set\_network\_connection\_mode 31](#_Toc494446354)

[1.3. System 32](#_Toc494446355)

[1.3.1. get\_system\_firmware\_version 32](#_Toc494446356)

[1.3.2. get\_system\_model\_id 32](#_Toc494446357)

[1.3.3. get\_system\_imei 33](#_Toc494446358)

[1.3.4. get\_system\_imsi 33](#_Toc494446359)

[1.3.5. get\_system\_iccid 34](#_Toc494446360)

[1.3.6. get\_system\_msisdn 34](#_Toc494446361)

[1.3.7. get\_operating\_mode 35](#_Toc494446362)

[1.3.8. set\_operating\_mode 35](#_Toc494446363)

[1.3.9. get\_mal\_manager\_version 36](#_Toc494446364)

[1.3.10.set\_host\_info 36](#_Toc494446365)

[1.4. LOC 38](#_Toc494446366)

[1.4.1. get\_loc\_config 38](#_Toc494446367)

[1.4.2. get\_loc\_position\_info 38](#_Toc494446368)

[1.4.3. set\_loc\_config 39](#_Toc494446369)

[1.4.4. set\_loc\_mode 40](#_Toc494446370)

[2. Revision History 41](#_Toc494446371)

# Introduction

## Definitions and Abbreviations

|  |  |
| --- | --- |
| JSON | JavaScript Object Notation |
| MAL | Modem Abstraction Layer |

## Purpose

This document discusses the Modem Abstraction Layer (MAL) and how it is used by the applications to interact with the LTE modem. The MAL resides within the WNC M18Qx module and application uses JSON strings to communicate through a Linux socket with it. By sending and receiving JSON strings the MAL and Application, the following types of functionality are available:

SIM card access functions, e.g., PIN control

Network Access functions, e.g., GSM/WCDMA/LTE signal strength[[1]](#footnote-1), network registration and Circuit-Switched/Packet-Switched attach, serving network information, and network provider scan

Wireless Data functions, e.g., data-call setup and tear-down, packet transmission statistics, and data-session profile management

Short message functions, e.g., send short message, group message, receive short message, center number

Location functions, e.g., position information

The MAL also provides for aggregate services via management services, similar to those listed below:

Connection management

SMS management (\* SMS only support by customer request)

Module monitor management

MAL manager configuration management

By sending JSON string to the MAL, the application has the ability to obtain status and to control the MAL and thus LTE-Modem activity.

## Sending a request

A request is a JSON object with is a string containing both an **“action”** and optional **“args”**.

|  |
| --- |
| { “**action**”: "xxx", “**args**”: {“name1”: ”value1”} } |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| **action** | String | Mandatory | Name of API being called |
| **Args** | Object | Optional | API arguments for this call (name/value pairs) |

## Receiving a response

In response to a request, the MAL will send a JSON response. This response contains the name this response is in reply to (the requested API name) and a series of name/value pairs. Each response will always contain an **“errno”:integer** pair and a **“errmsg”:string** pair. Other response key/value pairs depend on the request that was made.

|  |
| --- |
| {  “**action**” : {  "**errno**" : 0,  "**errmsg**" : "",  “name1” : “value1”,  “object1”: { “name2” : “value2”}  }  } |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| **action** | String | Mandatory | Name of API that was called |
| **err** | Integer | Mandatory | Result error number |
| **errmsg** | String | Mandatory | A string representing the error |
| **name1/value1** | Object | Optional | API arguments for this call (name/value pairs) |
| **object1** | Object | Optional | An array object with name/value pairs |

# Requests

## WWAN

Wireless WAN APIs are used to control the LTE module.

### get\_wwan\_serving\_system\_status

This API is used to query the radio mode, service provider, and roaming status currently in use.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

| Name | Type | Requirement | Description |
| --- | --- | --- | --- |
| radio\_mode | Integer | Mandatory | Radio interface currently in use:  0 - No service  3 - 3G  4 - 4G |
| signal\_strength | Integer | Mandatory | Signal strength |
| Level | Integer | Optional | Signal level (0–5) |
| lte\_rsrp | Integer | Optional | [Level 5] rsrp ≥ –95  [Level 4] –95 > rsrp ≥ –105  [Level 3] –105 > rsrp ≥ –116  [Level 2] –116 > rsrp ≥ –125  [Level 1] –125 > rsrp > –135  [Level 0] rsrp ≤ –135  **Or**  0 grid: RSRP < –114  1 grid: –114 ≤ RSRP < –104  2 grid: –104 ≤ RSRP < –98  3 grid: –98 ≤ RSRP < –92  4 grid: –92 ≤ RSRP < –84  5 grid: –84 ≤ RSRP |
| wcdma\_rscp | Integer | Optional | 5 grid: >= -84 RSCP 4 grid: >=-90 RSCP <-84 3 grid: >=-96 RSCP <-90 2 grid: >=-102 RSCP <-96 1 grid: >=-108 RSCP <-102 0 grid: <-108 |
| State | Integer | Mandatory | Registration state:  0 - Not registered  1 - Registered with a network  2 - Searching  3 - Registration denied by the visible network  4 - Registration state is unknown. |
| cs\_state | Integer | Mandatory | Circuit-switched domain attach state of the module:  0 - Unknown  1 - Attached  2 – Detached |
| ps\_state | Integer | Mandatory | Packet-switched domain attach state of the module:  0 - Unknown  1 - Attached  2 – Detached |
| Mcc | Integer | Mandatory | Mobile country codes |
| Mnc | Integer | Mandatory | Mobile network codes |
| lac | Integer | Mandatory | Location Area Code |
| cell\_id | Integer | Mandatory | Cell ID |
| tac | Integer | Optional | Tracking Area Code |
| pci | Integer | Optional | Physical cell ID |
| roaming | Integer | Mandatory | Roaming status:  0 - Home  1 - Roaming  2 – Unknown |
| provider | String | Mandatory | Network provider name  Unknown - If the network is not registered |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_serving\_system\_status" }  { "get\_wwan\_serving\_system\_status": { "errno": 0, "errmsg": "", "radio\_mode": 4, "signal\_strength": -32, "level": 5, "state": 1, "cs\_state": 1, "ps\_state": 1, "mcc": 466, "mnc": 92, "lac": 0, "cell\_id": 81023777, "tac": 13700, "pci": 482, "roaming": 0, "provider": "Chunghwa Telecom" } } |

### get\_wwan\_serving\_system\_provider

This API is used to query the service provider currently in use.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

| Name | Type | Requirement | Description |
| --- | --- | --- | --- |
| provider | String | Mandatory | Network provider name  Unknown - If the network is not registered |
| short\_name | String | Optional | Network provider short name  Unknown - If the network is not registered |
| source | Integer | Optional | Network provider name source:  0 - unknown  1 - PNN  2 - CHPS  3 - NITZ  4 - ME  5 - MCC-MNC  6 - SPN |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_serving\_system\_provider" }  { "get\_wwan\_serving\_system\_provider": { "errno": 0, "errmsg": "", "provider": "Chunghwa Telecom", "short name": "Chunghwa", "source": 4 } } |

### get\_wwan\_serving\_system\_radio\_mode

This API is used to query the radio mode currently in use.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

| Name | Type | Requirement | Description |
| --- | --- | --- | --- |
| radio\_mode | Integer | Mandatory | Radio interface currently in use:  0 - No service  3 - 3G  4 - 4G |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_serving\_system\_radio\_mode" } { "get\_wwan\_serving\_system\_radio\_mode": { "errno": 0, "errmsg": "", "radio\_mode": 3 } } |

### get\_wwan\_network\_radio\_mode

This API is used to get the network selection radio mode.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| radio\_mode | Integer | Mandatory | 3 - 3G only  4 - 4G only  34 - 3G and 4G |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_network\_radio\_mode" }  { "get\_wwan\_network\_radio\_mode": { "errno": 0, "errmsg": "", "radio\_mode": 234 } } |

### set\_wwan\_network\_radio\_mode

This API is used to specify the network selection radio mode.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| radio\_mode | Integer | Mandatory | 3 - 3G only  4 - 4G only  34 - 3G and 4G |

Example:

|  |
| --- |
| { "action" : "set\_wwan\_network\_radio\_mode" }  { "set\_wwan\_network\_radio\_mode ": { "errno": 0, "errmsg": "" } } |

### get\_wwan\_radio\_info

This API is used to query the detailed GSM/UMTS/LTE radio information.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

| Name | Type | Requirement | Description |
| --- | --- | --- | --- |
| radio\_mode | Integer | Mandatory | Radio interface currently in use:  0 - No service  3 - 3G  4 - 4G |
| gw\_signal\_strength | Integer | Optional | (This value is returned only when the radio mode is UMTS.) |
| gw\_bitErrorRate | Integer | Optional | (This value is returned only when the radio mode is UMTS.) |
| gw\_ecio | Integer | Optional | (This value is returned only when the radio mode is UMTS.) |
| wcdma\_rscp | Integer | Optional | (This value is returned only when the radio mode is UMTS.) |
| lte\_signal\_strength | Integer | Optional | (This value is returned only when the radio mode is LTE.) |
| lte\_rsrp | Integer | Optional | (This value is returned only when the radio mode is LTE.) |
| lte\_rsrq | Integer | Optional | (This value is returned only when the radio mode is LTE.) |
| lte\_rssnr | Integer | Optional | (This value is returned only when the radio mode is LTE.) |
| lte\_cqi | Integer | Optional | (This value is returned only when the radio mode is LTE.) |
| gwl\_csq | Integer | Optional | (This value is returned only when the radio mode is UMTS/LTE.) |
| rx0\_pwr | Float | Optional | (This value is returned only when the radio is turn on.) |
| rx1\_pwr | Float | Optional | (This value is returned only when the radio is turn on.) |
| tx\_pwr | Float | Optional | (This value is returned only when the radio is turn on.) |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_radio\_info" }  { "get\_wwan\_radio\_info": { "errno": 0, "errmsg": "", "radio\_mode": 4, "lte\_signal\_strength": -36, "lte\_rsrp": -65, "lte\_rsrq": -10, "lte\_rssnr": 29, "lte\_cqi": -1, "gwl\_csq": 31, "rx0\_pwr": -40, "rx1\_pwr": -93 } } |

### get\_wwan\_band\_capability

This API is used to query the band capability of the module.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| lte\_band | Array | Mandatory | LTE bands supported by the module |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_band\_capability" }  { "get\_wwan\_band\_capability": { "errno": 0, "errmsg": "", "lte\_band": [ 5, 7, 28 ] } } |

### get\_wwan\_band\_info

This API is used to query the current RF band information.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Parameter | Type | Requirement | Description |
| band\_info | | | | |
|  | radio\_if | String | Mandatory | Radio access technology. |
|  | band | String | Mandatory | Active band |
|  | channel | Integer | Mandatory | Active channel |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_band\_info" }  { "get\_wwan\_band\_info": { "errno": 0, "errmsg": "", "band\_info": [{ "radio\_if": "LTE", "band": "E-UTRA Operating Band 3", "channel": 1725 } } |

### get\_wwan\_lte\_ca\_info

This API is used to query carrier aggregation information of the LTE network.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

| Name | Type | Requirement | Description |
| --- | --- | --- | --- |
| State | Integer | Mandatory | Physical carrier aggregation of the Scell state  0 - Deconfigured  1 - Configured and deactivated  2 - Configured and activated |
| pcell\_pci | Integer | Optional | Physical cell ID of the Pcell |
| pcell\_freq | Integer | Optional | Absolute cell’s frequency |
| pcell\_bw | String | Optional | Downlink bandwidth |
| pcell\_band | Integer | Optional | Band |
| scell\_pci | Integer | Optional | Physical cell ID of the Scell |
| scell\_freq | Integer | Optional | Absolute cell’s frequency |
| scell\_bw | String | Optional | Downlink bandwidth |
| scell\_band | Integer | Optional | Band |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_lte\_ca\_info" }  { "get\_wwan\_lte\_ca\_info": { "errno": 0, "errmsg": "", "state": 1, "pcell\_pci": 189, "pcell\_freq": 1725, "pcell\_bw": "15MHz", "pcell\_band": 122, "scell\_pci": 189, "scell\_freq": 3650, "scell\_bw": "5MHz", "scell\_band": 127 } } |

### get\_wwan\_serving\_system\_roaming

This API is used to query the roaming status currently in use.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| Roaming | Integer | Mandatory | Roaming status:  0 - Home  1 - Roaming  2 - Unknown |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_serving\_system\_roaming" }  { "get\_wwan\_serving\_system\_roaming": { "errno": 0, "errmsg": "", "roaming": 0 } } |

### get\_wwan\_allow\_data\_roaming

This API is used to get data on the roaming setting.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| Enable | Integer | Mandatory | 0 - Do not allow data roaming.  1 - Allow data roaming. |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_allow\_data\_roaming" }  { "get\_wwan\_allow\_data\_roaming": { "errno": 0, "errmsg": "", "enable": 0 } } |

### set\_wwan\_allow\_data\_roaming

This API is used to specify the data roaming setting.

**REQUEST ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| Enable | Integer | Mandatory | 0 - Do not allow data roaming.  1 - Allow data roaming. |

**RESPONSE ARGUMENTS:** None.

### get\_wwan\_network\_time

This API is used to query the network time.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| network\_time | String | Mandatory | YYYY/MM/DD hh:mm:ss[TZ,DST\_offset]  If no [TZ,DST\_offset], this indicates the network time was not obtained from the system (not the operator). |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_network\_time" }  { "get\_wwan\_network\_time": { "errno": 0, "errmsg": "", "network\_time": "2014\/10\/20 06:35:27+00,00" } } |

### get\_wwan\_ipv4\_network\_state

This API is used to query the IPv4 network connection state.

**REQUEST ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| device\_id | Integer | Optional | PDN No. (The default is “0”.)  0 - PDN0  1 - PDN1  2 - PDN2 |

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| State | Integer | Mandatory | Connection status:  0 - Disconnected  1 - Disconnecting  2 - Connecting  3 - Connected |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_ipv4\_network\_state" }  { "get\_wwan\_ipv4\_network\_state": { "errno": 0, "errmsg": "", "state": 3 } } |

### get\_wwan\_ipv4\_network\_ip

This API is used to query the IPv4 network IP currently in use.

**REQUEST ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| device\_id | Integer | Optional | PDN No. (The default is “0”.)  0 - PDN0  1 - PDN1  2 - PDN2 |

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| Ip | String | Mandatory | IP address |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_ipv4\_network\_ip" }  { "get\_wwan\_ipv4\_network\_ip": { "errno": 0, "errmsg": "", "ip": "100.80.143.133" } } |

### get\_wwan\_ipv4\_network\_connection\_time

This API is used to query the IPv4 network connection time.

**REQUEST ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| device\_id | Integer | Optional | PDN No. (The default is “0”.)  0 - PDN0  1 - PDN1  2 - PDN2 |

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| connection\_time | String | Mandatory | Day:Hours:Minutes:Seconds |

Example:

|  |
| --- |
| { "action" : "get\_wwan\_ipv4\_network\_connection\_time" }  { "get\_wwan\_ipv4\_network\_connection\_time": { "errno": 0, "errmsg": "", "connection\_time": "00:16:07:05" } } |

### get\_wwan\_ipv4\_network\_status

This API is used to query the IPv4 network status currently in use.

**REQUEST ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| device\_id | Integer | Optional | PDN No. (The default is “0”.)  0 - PDN0  1 - PDN1  2 - PDN2 |

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| state | Integer | Mandatory | Connection status:  0 - Disconnected  1 - Disconnecting  2 - Connecting  3 - Connected |
| connection\_time | String | Optional | Day:Hours:Minutes:Seconds  (This value is returned only when the connection state is connected.) |
| ip | String | Optional | (This value is returned only when the connection state is connected.) |
| subnet | String | Optional | (This value is returned only when the connection state is connected.) |
| dns | String | Optional | (This value is returned only when the connection state is connected.) |
| gw | String | Optional | (This value is returned only when the connection state is connected.) |

Example:

|  |
| --- |
| { "action": "get\_wwan\_ipv4\_network\_status" }  { "get\_wwan\_ipv4\_network\_status": { "errno": 0, "errmsg": "", "state": 3, "connection\_time": "00:01:23:45", "ip": "100.82.136.144", "subnet": "255.255.255.252", "gw": "100.82.136.145", "dns": "61.31.233.1" } } |

### get\_wwan\_apn\_selection\_mode

This API is used to query the APN selection mode currently in use.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| mode | Integer | Mandatory | Selection mode:  0 - Auto  1 - Manual  2 - Selection |

Example:

|  |
| --- |
| { "action": "get\_wwan\_apn\_selection\_mode" }  { "get\_wwan\_apn\_selection\_mode": { "errno": 0, "errmsg": "", "mode": 2 } } |

### get\_wwan\_apn\_selection

This API is used to query the related information of the APN selection currently in use.

**REQUEST ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| mode | Integer | Mandatory | Selection mode:  0 - Auto  1 - Manual  2 - Selection |
| device\_id | Integer | Optional | PDN No. (The default is “0”.)  0 - PDN0  1 - PDN1  2 - PDN2 |

**RESPONSE ARGUMENTS:**

| Name | Type | Requirement | Description |
| --- | --- | --- | --- |
| index | Integer | Optional | Profile index |
| apn | String | Optional | Access point name |
| pdp\_type | Integer | Optional | Packet Data Protocol (PDP) type:  4 - IPv4  6 - IPv6  46 - IPv4 and IPv6 |
| user\_name | String | Optional | Username used during data network authentication |
| password | String | Optional | Password used during data network authentication |
| auth\_type | Integer | Optional | Authentication algorithm:  0 - None  1 - PAP  2 - CHAP  3 - Auto |
| dial\_num | String | Optional | Dial number |

Example:

|  |
| --- |
| { "action": "get\_wwan\_apn\_selection", "args": { "mode": 0 } }  { "get\_wwan\_apn\_selection" : { "errno": 0, "errmsg": "", "apn": "internet", "pdp\_type": 4, "user\_name": "", "password": "", "auth\_type": 0 } }  { "action": "get\_wwan\_apn\_selection", "args": { "mode": 1 } }  { "get\_wwan\_apn\_selection" : { "errno": 0, "errmsg": "", "apn": "internet", "pdp\_type": 4, "user\_name": "", "password": "", "auth\_type": 0, "dial\_num": "" } }  { "action": "get\_wwan\_apn\_selection", "args": { "mode": 2 } }  { "get\_wwan\_apn\_selection" : { "errno": 0, "errmsg": "", "index": 0 } } |

### set\_wwan\_apn\_selection

This API is used to specify the related information of APN selection.

**REQUEST ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| mode | Integer | Mandatory | Selection mode:  0 - Auto  1 - Manual  2 - Selection |
| device\_id | Integer | Optional | PDN No. (The default is “0”.)  0 - PDN0  1 - PDN1  2 - PDN2 |
| index | Integer | Optional | Profile index |
| apn | String | Optional | Access point name |
| pdp\_type | Integer | Optional | Packet Data Protocol (PDP) type:  4 - IPv4  6 - IPv6  46 - IPv4 and IPv6 |
| profile\_name | String | Optional | Profile name |
| user\_name | String | Optional | Username used during data network authentication |
| password | String | Optional | Password used during data network authentication |
| auth\_type | Integer | Optional | Authentication algorithm:  0 - None  1 - PAP  2 - CHAP  3 - Auto |
| dial\_num | String | Optional | Dial number |

**RESPONSE ARGUMENTS :** None.

Example:

### get\_wwan\_apn\_list

This API is used to query the APN list.

**REQUEST ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| index | Integer | Optional | Indicates the profile index  If it is not indicated, all profiles will be returned. |

**RESPONSE ARGUMENTS:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Parameter | Type | Requirement | Description |
| number | | | | |
|  |  | Integer | Mandatory | Indicates the number of the profiles |
| profiles | | | | |
|  | index | Integer | Mandatory | Profile index |
|  | apn | String | Mandatory | Access point name |
|  | pdp\_type | Integer | Mandatory | Packet Data Protocol (PDP) type:  4 - IPv4  6 - IPv6  46 - IPv4 and IPv6  If not indicated, the default is “4”. |
|  | profile\_name | String | Mandatory | Profile name |
|  | user\_name | String | Mandatory | Username used during data network authentication |
|  | password | String | Mandatory | Password used during data network authentication |
|  | auth\_type | Integer | Mandatory | Authentication algorithm:  0 - None  1 - PAP  2 - CHAP  3 - Auto |
|  | dial\_num | String | Mandatory | Dial number |

Example:

|  |
| --- |
| { "action": "get\_wwan\_apn\_list" }  { "get\_wwan\_apn\_list" :  { "errno": 0, "errmsg": "", "number": 5,  "profiles": [  { "index": 0, "pdp\_type": 4, "profile\_name": "profile1", "apn": "internet", "user\_name": "", "password": "", "auth\_type": 0, "dial\_num": "" },  { "index": 1, "pdp\_type": 4, "profile\_name": "APN1", "apn": "internet1", "user\_name": "", "password": "", "auth\_type": 0, "dial\_num": "" },  { "index": 2, "pdp\_type": 4, "profile\_name": "APN2", "apn": "internet3", "user\_name": "", "password": "", "auth\_type": 0, "dial\_num": "" },  { "index": 3, "pdp\_type": 4, "profile\_name": "APN3", "apn": "internet3", "user\_name": "", "password": "", "auth\_type": 0, "dial\_num": "" },  { "index": 4, "pdp\_type": 4, "profile\_name": "APN4", "apn": "internet4", "user\_name": "", "password": "", "auth\_type": 0, "dial\_num": "" } ] } } |

### get\_wwan\_apn\_settings

This API is used to query the related information of APN settings.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

| Name | Parameter | Type | Requirement | Description |
| --- | --- | --- | --- | --- |
| Mode | | | | |
|  |  | Integer | Mandatory | Selection mode:  0 - Auto  1 - Manual  2 - Selection |
| auto | | | | |
|  | apn | String | Mandatory | Access point name |
|  | pdp\_type | Integer | Mandatory | Packet Data Protocol (PDP) type:  4 - IPv4  6 - IPv6  46 - IPv4 and IPv6 |
|  | user\_name | String | Mandatory | Username used during data network authentication |
|  | password | String | Mandatory | Password used during data network authentication |
|  | auth\_type | Integer | Mandatory | Authentication algorithm:  0 - None  1 - PAP  2 - CHAP  3 - Auto |
| manual | | | | |
|  | apn | String | Mandatory | Access point name |
|  | pdp\_type | Integer | Mandatory | Packet Data Protocol (PDP) type:  4 - IPv4  6 - IPv6  46 - IPv4 and IPv6 |
|  | user\_name | String | Mandatory | Username used during data network authentication |
|  | password | String | Mandatory | Password used during data network authentication |
|  | auth\_type | Integer | Mandatory | Authentication algorithm:  0 - None  1 - PAP  2 - CHAP  3 - Auto |
|  | dial\_num | String | Mandatory | Dial number |
| selection | | | | |
|  | index | Integer | Mandatory | Profile index |

Example:

|  |
| --- |
| { "action": "get\_wwan\_apn\_settings" }  { "get\_wwan\_apn\_settings": { "errno": 0, "errmsg": "", "mode": 2,  "auto": { "apn": "internet", "pdp\_type": 4, "user\_name": "", "password": "", "auth\_type": 0 },  "manual": { "apn": "internet", "pdp\_type": 4, "user\_name": "", "password": "", "auth\_type": 0, "dial\_num": "" },  "selection": { "index": 0 } } } |

### set\_wwan\_preferred\_network

This API is used to specify network selection preference.

**REQUEST ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| plmn\_mode | Integer | Mandatory | 0 - Automatic network selection (Device registers according to its provisioning; mcc, mnc, radio\_mode fields are ignored.)  1 - Manual network selection (Device registers to a specified network; mcc and mnc fields must also contain valid values.) |
| mcc\_mnc | String | Optional | PLMN |
| radio\_mode | Integer | Optional | 3 - 3G  4 - 4G |

**RESPONSE ARGUMENTS:** None.

Example:

|  |
| --- |
| { "action": "set\_wwan\_preferred\_network", "args": { "plmn\_mode" : 1, "mcc\_mnc": "46002" } }  { "set\_wwan\_preferred\_network" : { "errno": 0, "errmsg": "" } } |

### get\_wwan\_preferred\_network\_mode

This API is used to get the network selection mode.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| plmn\_mode | Integer | Mandatory | 0 - Automatic network selection  1 - Manual network selection |

Example:

|  |
| --- |
| { "action": "get\_wwan\_preferred\_network\_mode" }  { "get\_wwan\_preferred\_network\_mode": { "errno": 0, "errmsg": "", "plmn\_mode": 0 } } |

## Network

Network APIs are for controlling the system network.

### get\_network\_connection\_status

This API is used to query the internet connection status currently in use.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

| Name | Type | Requirement | Description |
| --- | --- | --- | --- |
| Type | Integer | Mandatory | Connection type currently in use:  0 - 3G/LTE  1 - Router mode (Ethernet)  2 - Wi-Fi (hotspot) |
| State | Integer | Mandatory | Connection status:  0 - Disconnected  1 - Disconnecting  2 - Connecting  3 - Connected  4 - Disconnected, and PIN locked  5 - Disconnected, and SIM removed |
| connection\_time | String | Optional | Day:Hours:Minutes:Seconds  (This value is returned only when the connection state is connected.) |
| provider | String | Optional | Network provider name  Unknown - If the network is not registered |
| radio\_mode | Integer | Optional | Radio interface currently in use:  0 - No service  3 - 3G  4 - 4G |
| data\_bearer\_tech | String | Optional | Current data bearer technology.  (This value is returned only when the connection state is connected.) |
| roaming | Integer | Optional | Roaming status:  0 - Home  1 - Roaming  2 - Unknown |
| signal\_strength | Integer | Optional | Signal strength |
| Level | Integer | Optional | Signal level |
| lte\_rsrp | Integer | Optional | LTE rsrp. (This value is returned only when the connection type is LTE.) |
| wcdma\_rscp | Integer | Optional | WCDMA rscp. (This value is returned only when the connection type is 3G.) |
| ipv6 | String | Optional | IPv6 address  (This value is returned only when the IPv6 connection state is connected.) |

Example:

|  |
| --- |
| { "action" : "get\_network\_connection\_status" }  { "get\_network\_connection\_status": { "errno": 0, "errmsg": "", "type": 0, "state": 3, "connection\_time": "00:00:00:18", "ip": "10.120.157.220", "provider": "Chunghwa Telecom", "radio\_mode": 4, "data\_bearer\_tech": "LTE", "roaming": 0, "level": 5, "signal\_strength": -45, "lte\_rsrp": -68, "wcdma\_rscp": 0 } } |

### get\_network\_connection\_mode

This API is used to query the internet connection mode.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| mode | Integer | Mandatory | Connection mode:  0 - Always  1 - On-demand |
| ondemand\_timeout | Integer | Mandatory | Will disconnect in minutes if there is no Internet access |
| manual\_mode | Integer | Mandatory | Connection mode:  0 - Disconnect  1 - Connect |

Example:

|  |
| --- |
| { "action" : "get\_network\_connection\_mode" }  { "get\_network\_connection\_mode": { "errno": 0, "errmsg": "", "mode": 0, "ondemand\_timeout": 10, "manual\_mode": 1 } } |

### set\_network\_connection\_mode

This API is used to specify the internet connection mode.

**REQUEST ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| mode | Integer | Optional | Connection mode:  0 - Always  1 - On-demand |
| ondemand\_timeout | Integer | Optional | Disconnect in minutes if there is no Internet access |
| manual\_mode | Integer | Optional | Connection mode:  0 - Disconnect  1 - Connect (Always/on-demand)  2 - Connect once |

**RESPONSE ARGUMENTS:** None.

## System

### get\_system\_firmware\_version

This API is used to query the modem firmware version.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| version | String | Mandatory | Modem firmware version |

Example:

|  |
| --- |
| { "action" : "get\_system\_firmware\_version" }  { "get\_system\_firmware\_version": { "errno": 0, "errmsg": "", "version": "M18Q2\_v11.01 1 [Mar 21 2016 11:00:00]" } } |

### get\_system\_model\_id

This API is used to query the modem ID.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| model\_id | String | Mandatory | Modem ID |

Example:

|  |
| --- |
| { "action": "get\_system\_model\_id" }  { "get\_system\_model\_id": { "errno": 0, "errmsg": "", "model\_id": "D19QC" } } |

### get\_system\_imei

This API is used to query the IMEI (International Mobile Equipment Identity).

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

Additional valid response values are listed in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| imei | String | Mandatory | IMEI |

Example:

|  |
| --- |
| { "action": "get\_system\_imei" }  { "get\_system\_imei": { "errno": 0, "errmsg": "", "imei": "355407045000518" } } |

### get\_system\_imsi

This API is used to query the IMSI (International Mobile Subscriber Identity).

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

Additional valid response values are listed in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| imsi | String | Mandatory | IMSI |

Example:

|  |
| --- |
| { "action" : "get\_system\_imsi" }  { "get\_system\_imsi": { "errno": 0, "errmsg": "", "imsi": "466923302575936" } } |

### get\_system\_iccid

This API is used to query the ICCID (Integrate Circuit Card IDentity).

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

Additional valid response values are listed in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| iccid | String | Mandatory | ICCID |

Example:

|  |
| --- |
| { "action" : "get\_system\_iccid" }  { "get\_system\_iccid": { "errno": 0, "errmsg": "", "iccid": "89886920033025759360" } } |

### get\_system\_msisdn

This API is used to query the MSISDN (Mobile Subscriber International ISDN number).

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

Additional valid response values are listed in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| msisdn | String | Mandatory | msisdn |

Example:

|  |
| --- |
| { "action" : "get\_system\_msisdn" }  { "get\_system\_msisdn": { "errno": 0, "errmsg": "", "msisdn": "7666555544778888" } } |

### get\_operating\_mode

This API is used to query the operating status of the LTE module.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| operating\_mode | Integer | Mandatory | 0 - Online  1 - Low power  2 - Factory Test mode  3 - Offline  4 - Resetting  5 - Shutting down |

Example:

|  |
| --- |
| { "action" : "get\_operating\_mode" }  { "get\_operating\_mode": { "errno": 0, "errmsg": "", "operating\_mode": 0 } } |

### set\_operating\_mode

This API is used to specify the operation of the LTE module.

**REQUEST ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| operating\_mode | Integer | Mandatory | 0 - Online  1 - Low power  2 - Factory Test mode  3 - Offline  4 - Resetting  5 - Shutting down |

**RESPONSE ARGUMENTS:** None.

### get\_mal\_manager\_version

This API is used to query the MAL manager version.

**REQUEST ARGUMENTS:** None

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| version | String | Mandatory | Indicates the version of the MAL manager |

Example:

|  |
| --- |
| { "action" : "get\_mal\_manager\_version" }  { "get\_mal\_manager\_version": { "errno": 0, "errmsg": "", "version": "malm\_10\_v01.06.1501200" } } |

### set\_host\_info

This API is used to set host information to the module.

**REQUEST ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| fw\_vesion | String | Mandatory | The host firmware version |
| manufacturer | String | Mandatory | The host manufacturer |
| model\_name | String | Mandatory | The host model name |
| model\_id | String | Mandatory | The host model id |

**RESPONSE ARGUMENTS:** None.

Example:

|  |
| --- |
| { "action": "set\_host\_firmware\_version", "args": { "fw\_vesion": "Host version is v1.00" }, { "manufacturer": "Company" },{ "model\_name": "module\_1" },{ "model\_id": "12345" }}  { "set\_host\_host\_info": { "errno": 0, "errmsg": "" } } |

## LOC

LOC APIs are used for controlling the GPS information.

### get\_loc\_config

This API is used to get the configuration of the LOC.

**REQUEST ARGUMENTS:** None

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| Loc | Boolean | Mandatory | true/false  The LOC is enabled or disabled. |
| mode | Integer | Mandatory | Operation mode:  2 – MS-based mode  3 – MS-assisted mode  4 – Standalone mode |
| nmea\_filter | String | Mandatory | Bitmasks of NMEA types. |

Example:

|  |
| --- |
| { "action": "get\_loc\_config" }  { "get\_loc\_config": { "errno": 0, "errmsg": "", "loc": true, "mode": 4, "nmea\_filter": "0x00007f9f" } } |

### get\_loc\_position\_info

This API is used to get the position report.

**REQUEST ARGUMENTS:** None.

**RESPONSE ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| loc\_status | Integer | Mandatory | LOC session status:  0 – Session was ended  1 – Session is in progress |
| latitude | Float | Optional | Latitude |
| longitude | Float | Optional | Longitude |
| timestamp | Integer | Optional | UTC Timestamp |
| altitude | Float | Optional | Altitude with respect to the WGS84 Ellipsoid. |
| speed | Float | Optional | Speed |
| bearing | Float | Optional | Heading |
| accuracy | Float | Optional | Horizontal position uncertainty (circular). |

Example:

|  |
| --- |
| { "action": "get\_loc\_position\_info" }  { "get\_loc\_position\_info": { "errno": 0, "errmsg": "", "loc\_status": 1, "latitude": 22.990002036094666, "longitude": 119.34999704360962, "altitude": 0 } } |

### set\_loc\_config

This API is used to enable or disable the LOC.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| loc | Boolean | Mandatory | true/false  The LOC is enabled or disabled. |

**REQUEST ARGUMENTS:**

**RESPONSE ARGUMENTS:** None

### set\_loc\_mode

This API is used to specify the operation mode to enact position fixes.

**REQUEST ARGUMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Requirement | Description |
| mode | Integer | Mandatory | Operation mode:  2 – MS-based mode  3 – MS-assisted mode  4 – Standalone mode |

**RESPONSE ARGUMENTS:** None.

# Revision History

|  |  |  |
| --- | --- | --- |
| Date | Version | Revision |
| 28 Sept 17 | 00 | Initial Draft |

1. [↑](#footnote-ref-1)