
Network 1.0

September 9, 2024



Contents

Network	2
Set timeout	2
Pinging	2
Pinging multiple times	3
DNS lookup	3
Wake On Lan	3
ARP	3
Ipconfig	3
Traceroute	4
Netstat	4

Network

This module can do network operations

Set timeout

```
1 var network = Module.load("Network", { version: "v1.0.0" });
2 network.setTimeout(500); // default
```

the timeout value used for `ping`, `multiplePings` & `traceroute`

Pinging

```
1 var network = Module.load("Network", { version: "v1.0.0" });
2 var result = network.ping("sirenia.eu");
```

The `result` is a `PingResultInstance` JSON object with the following properties:

- `status` is the status of the attempted ping
- `time` is the round trip time of the ping
- `bytes` is the amount of bytes received
- `address` is the IP address of the reply
- `ttl` is the time to live
- `toString()` string representation of result

Pinging multiple times

```
1 var network = Module.load("Network", { version: "v1.0.0" });
2 var result = network.multiplePings("sirenia.eu", 7);
```

The `result` is a `MultiplePingInstance` JSON object with the following properties:

- `averageTime` is the average time for the pings
- `all` is an array containing `PingResultInstance` for each ping
- `toString()` string representation like in terminal.

DNS lookup

```
1 var network = Module.load("Network", { version: "v1.0.0" });
2 var result = network.dnsLookup("sirenia.eu");
```

The `result` is an array of strings which holds the IP addresses the host is known by, often just one result.

Wake On Lan

```
1 var network = Module.load("Network", { version: "v1.0.0" });
2 network.wakeOnLAN("04-7C-16-2C-61-C1");
```

Turns on the computer with the mac address "04-7C-16-2C-61-C1"

ARP

```
1 var network = Module.load("Network", { version: "v1.0.0" });
2 var res = network.arp();
```

The `result` is an `ArpInstance` JSON object, with the following properties:

- `table` returns an object with all IP's mapped to a mac address
 - eg. `res.table["10.0.0.1"]` to get the mac of ip 10.0.0.1
- `toString()` returns the table formatted as a string

Ipconfig

```
1 var network = Module.load("Network", { version: "v1.0.0" });
2 var res = network.ipconfig();
```

The `result` is a `IPConfigInstance` JSON object with the following properties:

- `mac` is your physical address / mac address
- `dnsSuffix` is your DNS suffix
- `iPv4` is your IPv4 Address
- `subnetMask` is your subnet mask
- `DHCP` is your DHCP server
- `dnsServers` is your DNS servers as a string split by a space
- `description` is description on your internet device
- `name` is the name of your connection
- `type` is the type of your connection
- `linkLocalIPv6` is your link-local IPv6 address
- `toString()` returns a string with all the information

Traceroute

```
1 var network = Module.load("Network", { version: "v1.0.0" });
2 var options = { ttl: 128 }; // default value
3 var res = network.traceroute("sirenia.eu", options);
```

the `result` is a `TracerouteInstance` JSON object with the following properties:

- `pings` is the array of `PingResultInstance` from the traceroute
- `toString()` returns a string formatted like traceroute in terminal

Netstat

```
1 var network = Module.load("Network", { version: "v1.0.0" });
2 var res = network.netstat("-a"); // -a is optional and does the same as
    -a for netstat in terminal
```

the `result` is a `NetStatInstance` JSON object with the following properties:

- `getActive()` returns an array of `ActiveConnection`
- `getTCPListeners()` returns an array of `ListeningConnection`. Note this is empty if “-a” flag is not used

- `getUDPListeners()` returns an array of `ListeningConnection`. Note this is empty if “-a” flag is not used
- `toString()` returns all results in a string

`ActiveConnection` is a JSON with the following properties:

- `getLocalAddress()` returns the local address
- `getForeignAddress()` returns the foreign address
- `getState()` returns the state of the connection
- `toString()` returns the connection formatted as a string

`ListeningConnection` is a JSON with the following properties:

- `getLocalAddress()` returns the local address
- `toString()` returns the connection formatted as a string