



Margay User Guide

Contents

Contents	2
Introduction to Margay	3
Viewing authentication or data throughput statistics	4
Before you begin	4
Logging in	4
Setting the country code	5
Changing the data time period	6
Margay API commands	7
getAuthenticationInformation	8
Example	8
getCookieName	10
getThroughputInformation	11
Example	11
login	14
logout	15
ping	16
Coding example	17
PHP cURL	17
Troubleshooting Postman errors	18

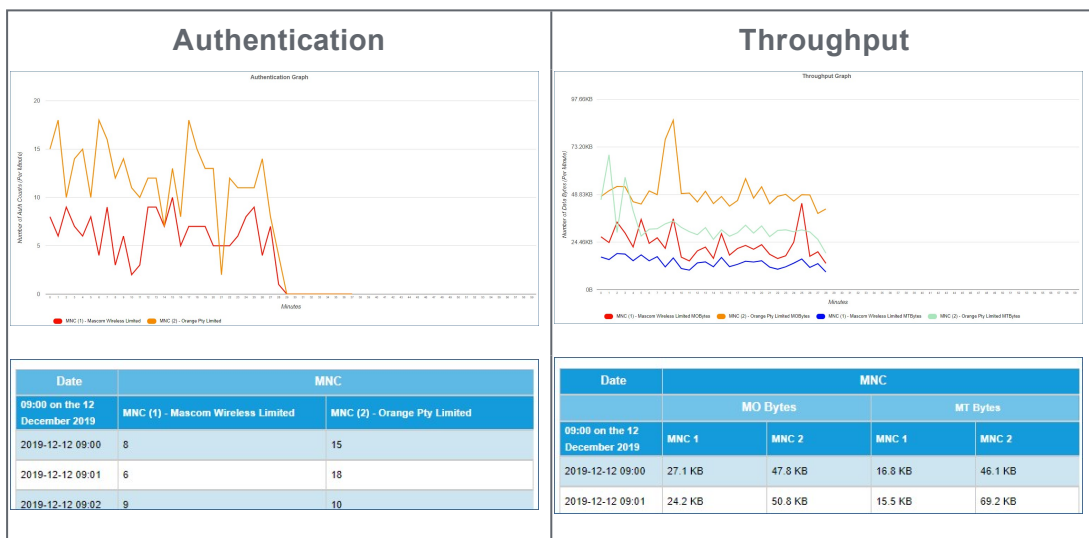
Introduction to Margay

Margay provides high level insight into device connection across different network operators, so that you can observe patterns, problems and anomalies with your network performance. For example, Margay can show if all the devices on a particular network have stopped sending or receiving information, or reveal server issues if there is a spike in authentications.

Margay provides authentication count and throughput information about your devices over a set period of time. The information is fetched from Eseye Ltd servers.

You access the connection metrics through either of the following:

- Eseye-hosted web interface



For more information, see [Viewing authentication or data throughput statistics](#) on page 4.

- JSON API that you can call using any API development app

```

HTTP
1 POST /api/niam/getAuthenticationInformation HTTP/1.1
2 Host: margay.eseye.com
3 Content-Type: application/json
4 User-Agent: PostmanRuntime/7.20.1
5 Accept: */*
6 Cache-Control: no-cache
7 Postman-Token: 0754e3f3-b283-410e-835c-cde1a4f18ac7,1c58c1d2-9139-402a-9fe0-688d56b5278c
8 Host: margay.eseye.com
9 Accept-Encoding: gzip, deflate
10 Content-Length: 85
11 Connection: keep-alive
12 cache-control: no-cache
13
  
```

For more information, see [Margay API commands](#) on page 7.

Viewing authentication or data throughput statistics

Using the Margay web interface, you can view the following:

- the number of successful device authentications on a network
- the rate of successful message delivery (throughput) in bytes on a network

You can view the statistics over three different time periods:

- Per minute in a selected hour (0 to 60 minutes)
- Per hour in a selected day (0 to 24h)
- Per day in a selected month (0 to the number of days in that month)

Before you begin

Contact your Eseye Account Manager about using Margay.

You need the following:

- Username
- Password

Logging in

1. Navigate to the Margay portal: <http://margay.eseye.com/Niam/index>.
2. Type your credentials, then select **Sign In**.

The Margay - Country Selector Page appears.

Setting the country code

Margay displays authentication and throughput information by geographic region using the three-digit Mobile Country Code (MCC). For more information and a list of country codes, see https://en.wikipedia.org/wiki/Mobile_country_code.

Margay - Country Selector Page

MCC *

To set the country code:

1. Type a country name or code number, or select the code from the **MCC** drop-down list.



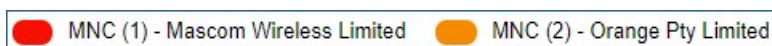
This field is mandatory.

2. Select either:

Authentication	to view the number of devices that have successfully opened a data session in the selected country
Throughput	to view successful data rates for devices in the selected country

Based on your selection, Margay displays a line graph of the current statistics per minute, generated in the current hour.

Use the key at the bottom of the graph to understand which line colour represents which network operator within the selected country. The key lists the mobile operators by their Mobile Network Code (MNC).





Scroll down to see the table of values on which the graph is based.



You can change the time period for the data. For more information, see [Changing the data time period](#) below.

Changing the data time period

You can view data statistics for the following time periods:



- Per minute in a selected hour (0 to 60 minutes)
- Per hour in a selected day (0 to 24h)
- Per day in a selected month (0 to the number of days in that month)



When you log in, the initial graph and table display the number of successful authentications per minute in the current hour.

Change the data time period using the navigation bar:



- Select  to return to the month, day, or previous hour
- Select  to advance to the next hour

Alternatively, select a link in the data table to view statistics for that time period:

Date	
12 December 2019	MNC
2019-12-12 00:00	240
2019-12-12 01:00	285

Margay API commands

You can call the Margay API in any API development app using the following commands:

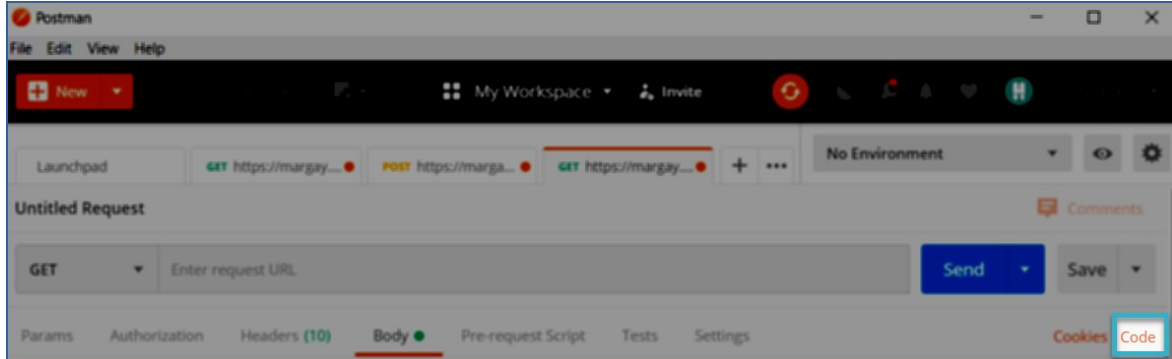
getAuthenticationInformation	8
getCookieName	10
getThroughputInformation	11
login	14
logout	15
ping	16

All information is sent in JSON and grouped by MCC and MNC.



The following examples are written for Postman. For more information about using Postman, see the *Postman User Guide*: <https://eseye.zendesk.com/hc/en-us/articles/360013066514-Postman-API-user-guide>.

Using Postman, if required, you can select **Code** to generate code snippets.



For information about troubleshooting Postman errors, see **Troubleshooting Postman errors** on page 18.

getAuthenticationInformation

You must first log in to the server before you can view Authentication information. Using the API, when you post a `getAuthenticationInformation` command, the system looks for a cookie token to validate the login before it returns information. For more information, [see login](#) on page 14.

Description	<p>Retrieves a summary of authentication information for the previous 10 minutes.</p> <p>Requires authorisation using either the login <code>CookieToken</code> or user credentials.</p> <p>Example</p> <p>Using Postman, on the Authorization tab, provide either the Bearer Token or Basic Auth details.</p>
Method	POST
Request URL	https://margay.eseye.com/Japi/Niam/getAuthenticationInformation
Successful Response	<pre> { "status": { "status": "OK", "errorCode": "", "errorMessage": "" }, "auth_count": [{ "mcc": "234", "mnc": "10", "start_date": "2019-09-14 16:12:00", "stop_date": "2019-09-14 16:22:00" "auth-accept": "3" }, { "mcc": "234", "mnc": "11", "start_date": "2019-09-14 16:12:00", </pre>


```

        "stop_date": "2019-09-14 16:22:00"
        "auth-accept": "6"
    }
  ]
}

```

where

JSON Key	Returned value
status	"OK" – the request processed successfully "ERR" – the request failed
errorCode	" " – the request processed successfully "ErrorCodeNumber" – the request failed because of this specific error For more information, see Troubleshooting Postman errors on page 18.
errorMessage	" " – the request processed successfully "ErrorMessage" – displays an explanation of the error if the request failed
auth_count	The authentication count heading. Each authentication count is grouped into MCC, MNC, the start and stop dates and times, and how many authentications were counted.
mcc	"MobileCountryCode" – the country where the authentications were counted, in three-digit numeric format
mnc	"MobileNetworkCode" – the network operator

JSON Key	Returned value
	where the authentications were counted, in one or two digit numeric format
start_date	"YYYY-MM-DD HH:MM:SS" – the authentication count start date and time in UTC
stop_date	"YYYY-MM-DD HH:MM:SS" –the authentication count end date and time in UTC
auth_accept	the total number of accepted authentications within the specified time period

getCookieName

You must first log in to the server before you can For more information, [see login on page 14](#).

Description	Returns the name of the cookie that is used to keep the session logged in. Does not require formatted text.
Method	POST
Request URL	https://margay.eseye.com/Japi/Niam/getCookieName
Successful Response	<pre>{ "cookie": "CookieToken", }</pre> <p>where</p>

JSON Key	Returned value
cookie	<p>"<i>CookieToken</i>" – displays the name of the cookie that is used to keep the session logged in.</p> <p>Could not get any response – either the Eseye server is down or the request was incorrect.</p>

getThroughputInformation

You must first log in to the server before you can view Throughput information. Using the API, when you post a `getThroughputInformation` command, the system looks for a cookie token to validate the login before it returns information. For more information, [see login on page 14](#).

Description	<p>Retrieves a summary of data sent and received for the previous 10 minutes.</p> <p>Requires authorisation using either the login <i>CookieToken</i> or user credentials.</p> <p>Example</p> <p>Using Postman, on the Authorization tab, provide either the Bearer Token or Basic Auth details.</p>
Method	POST
Request URL	https://margay.eseye.com/Japi/Niam/getThroughputInformation
Successful	<pre>{ "status": {</pre>

Response

```

    "status": "OK",
    "errorCode": "",
    "errorMessage": ""
  },
  "data_count": [
    {
      "mcc": "234",
      "mnc": "10",
      "databytes_mo": "517",
      "databytes_mt": "284",
      "start_date": "2016-09-14 16:12:37",
      "stop_date": "2016-09-14 16:22:37"
    }
  ]
}

```

where

JSON Key	Returned value
status	"OK" – the request processed successfully "ERR" – the request failed
errorCode	" " – the request processed successfully "ErrorCodeNumber" – the request failed because of this specific error For more information, see Troubleshooting Postman errors on page 18.
errorMessage	" " – the request processed successfully "ErrorMessage" – displays an explanation of the error if the request failed
data_count	The throughput heading. Each throughput is grouped into MCC, MNC, the start and stop dates and times, mobile originating bytes (databytes_mo), and mobile

JSON Key	Returned value
	terminating bytes (databytes_mt).
mcc	" <i>MobileCountryCode</i> " – the country where the authentications were counted, in three-digit numeric format
mnc	" <i>MobileNetworkCode</i> " – the network operator where the authentications were counted, in one or two digit numeric format
databytes_ mo	" <i>TotalNumberBytesOriginated</i> " – the total number of bytes sent by devices using the specified network operator in the specified time period
databytes_mt	" <i>TotalNumberBytesTerminated</i> " – the total number of bytes received by devices using the specified network operator in the specified time period
start_date	"YYYY-MM-DD HH:MM:SS" – the authentication count start date and time in UTC
stop_date	"YYYY-MM-DD HH:MM:SS" –the authentication count end date and time in UTC

login

Description	<p>Logs the user into Margay. Requires the following formatted Body JSON code:</p> <pre>{ "username" : "Example@Username.com" , "password" : "examplepassword" }</pre>								
Method	POST								
Request URL	https://margay.eseye.com/Japi/Niam/login								
Successful Response	<pre>{ "cookie": "CookieToken", "status": { "status": "OK", "errorCode": "", "errorMessage": "" } }</pre> <p>where</p> <table border="1"> <thead> <tr> <th>JSON Key</th> <th>Returned value</th> </tr> </thead> <tbody> <tr> <td>cookie</td> <td>"CookieToken" – displays the name of the cookie that is used to keep the session logged in. This line is the same as the getCookieName function.</td> </tr> <tr> <td>status</td> <td>"OK" – the request processed successfully "ERR" – the request failed</td> </tr> <tr> <td>errorCode</td> <td>" " – the request processed successfully</td> </tr> </tbody> </table>	JSON Key	Returned value	cookie	"CookieToken" – displays the name of the cookie that is used to keep the session logged in. This line is the same as the getCookieName function.	status	"OK" – the request processed successfully "ERR" – the request failed	errorCode	" " – the request processed successfully
JSON Key	Returned value								
cookie	"CookieToken" – displays the name of the cookie that is used to keep the session logged in. This line is the same as the getCookieName function.								
status	"OK" – the request processed successfully "ERR" – the request failed								
errorCode	" " – the request processed successfully								

JSON Key	Returned value
	" <i>ErrorCodeNumber</i> " – the request failed because of this specific error
errorMessage	" " – the request processed successfully " <i>ErrorCodeMessage</i> " – displays an explanation of the error if the request failed

logout

Description	Logs the user out of Margay.						
Method	POST						
Request URL	https://margay.eseye.com/Japi/Niam/logout						
Successful Response	<pre>{ "status": { "status": "OK", "errorCode": "", "errorMessage": "" } }</pre> <p>where</p> <table border="1"> <thead> <tr> <th>JSON Key</th> <th>Returned value</th> </tr> </thead> <tbody> <tr> <td>status</td> <td>"OK" – the request processed successfully "ERR" – the request failed</td> </tr> <tr> <td>errorCode</td> <td>" " – the request processed successfully</td> </tr> </tbody> </table>	JSON Key	Returned value	status	"OK" – the request processed successfully "ERR" – the request failed	errorCode	" " – the request processed successfully
JSON Key	Returned value						
status	"OK" – the request processed successfully "ERR" – the request failed						
errorCode	" " – the request processed successfully						

JSON Key	Returned value
	" <i>ErrorCodeNumber</i> " – the request failed because of this specific error
errorMessage	" " – the request processed successfully " <i>ErrorCodeMessage</i> " – displays an explanation of the error if the request failed

ping

Description	Tests whether the server is running.
Method	POST
Request URL	https://margay.eseye.com/Japi/Niam/ping
Successful Response	pong

Coding example

This example is based on the `getThroughputInformation` command.

PHP cURL

```
<?php
$curl = curl_init(); curl_setopt_array($curl, array(
    CURLOPT_URL =>
"https://margay.eseye.com/Japi/Niam/getthroughputinformation"
,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "POST",
    CURLOPT_POSTFIELDS => "{\r\n\"username\" :
\"ExampleUsername\" ,\r\n\"password\" : \"ExamplePassword\"
,\r\n\"portfolioID\" : \"ExamplePortfolioID\"\r\n}",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "content-type: application/json",
        "postman-token: 823e2f1c-d843-c90f-9d07-107061f5e417"
    ),
));

$response = curl_exec($curl);
$error = curl_error($curl);

curl_close($curl);


if ($error) {
    echo "cURL Error #:" . $error;
} else {
    echo $response;
}
```

Troubleshooting Postman errors

When an error occurs, an error code and message are generated in the returned result:

```
{
  "status": {
    "status": "ERR",
    "errorCode": "ErrorCodeNumber",
    "errorMessage": "ErrorCodeMessage"
  },
}
```

The following table can help you diagnose the error:

Error Code and Message	Reason for error
E0000 – Unknown username or password	The password value is incorrect.
E0006 – Permission denied	Indicates that the user is already logged out.
E0100 – Unknown username or password	The username or password value is incorrect.
E0101 – Invalid JSON	<p>The JSON text is structured incorrectly. Check that the JSON text follows the format described.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;">  <p>You may receive this error if you copied and pasted text from Microsoft Word. Some MS Word characters are slightly different from ASCII text. To avoid this, either type the text directly into Postman, or configure Microsoft Word to use basic ASCII text.</p> </div>
E9992 – User disabled for configuration	Indicates that the PortfolioID parameter is incorrect.

Error Code and Message	Reason for error
E9998 - Other error	Restart the session. If this error appears repeatedly, contact Eseye Support: support@eseYE.com.
Could not get any response	<p>Error connecting to the server, for any of the following reasons:</p> <ul style="list-style-type: none"> • You are not connected to the internet • The Request URL is incorrect in one or more of the following sections: <ul style="list-style-type: none"> <code>http</code> <code>://</code> <code>.eseYE.com</code> • The server backend is not working properly • SSL connections were blocked: Fix this by importing SSL certificates in Chrome • Cookies were not sent: Fix this by using the Postman Interceptor extension • Restricted headers were not sent: Fix this by using the Postman Interceptor extension • Request timeout: Fix this by changing request timeout settings
<!DOCTYPE html>	<p>Error message begins as follows:</p> <pre><!DOCTYPE html> <!--[if IE 6]> <html id=ie6" land="en-US"> <![endif]--> <!--[if IE 7]> <html id="ie7" lang="en-US"> ...</pre> <p>Indicates that the Request URL servername (Margay) is incorrect.</p>