NEC ESMPRO Manager Ver.6
User's Guide
RESTful API Reference

E6.66-01-STD

Contents

Terminology

Term	Description
BMC	Baseboard Management Controller The management controller to manage the interface between system management software and platform hardware.
Component	An object that is managed by NEC ESMPRO Manager.
ESXi management	Managed connect VMware vSphere Web Service. The target of management is VMware ESXi8.
iLO	Integrated Lights-Out The management controller to manage the interface between system management software and platform hardware.
IML	Integrated Management Log The IML provides a record of historical events that have occurred on the server.
NEC ExpressUpdate	The function that manages versions of modules like firmware and software on the managed server and that updates the modules. Available since NEC ESMPRO Manager Ver5.1.
NEC ExpressUpdate Agent	Software that provides NEC ExpressUpdate function. It is installed on the managed component and communicates with NEC ESMPRO Manager.
SNMP management	Managed using Simple Network Management protocol. The target of management is NEC ESMPRO Agent, iStorage.
vPro	Intel platform brand for business users (Intel® vPro™ Technology).
WS-Man	WS-Management Open standard protocol for remotely accessing management data with any computer device.
WS-Man management	Managed using WS-Management protocol. The target of management is NEC ESMPRO ServerAgentService, VMware ESXi7 and earlier.
IPMI	Intelligent Platform Management Interface It is a standard interface specification for monitoring server hardware independently of system state or OS.
Redfish	Redfish is a standard interface specification defined by the DMTF (Distributed Management Task Force) that is newer than IPMI, which manages the operation of servers.

Trademarks

NEC EXPRESSBUILDER and NEC ESMPRO are registered trademarks of NEC Corporation.
Microsoft, Windows, Windows Server are registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.
Intel, Xeon and Intel vPro are registered trademarks or trademarks of Intel Corporation in the United States and other countries.

All other company, or product names used in this document are registered trademarks or trademarks of their respective trademark owners.

All names used in sample applications are fictitious. They are unrelated to existing product, organization, or individual names.

■ **Notes**

(1) No part of this document may be reproduced in any form without the prior written permission of NEC Corporation.
(2) The contents of this document may be revised without prior notice.
(3) The contents of this document shall not be copied or altered without the prior written permission of NEC Corporation
(4) All efforts have been made to ensure the accuracy of all information in this document. If you notice any part unclear, incorrect, or omitted in the document, contact your authorized NEC sales representative.
(5) NEC assumes no liability for damages arising from the use of this product, nor any liability for incidental or consequential damages arising from the use of this document regardless of (4)

About This Document

This document introduces "RESTful API" (hereinafter, this is called "REST API") of the component management utility "NEC ESMPRO Manager".
Before attempting to operate the REST API, read this document so as to gain an adequate understanding of the contents.

■ Attention

This document is intended for persons who are familiar with the operating system's functions and operations and the network's functions and setup. For operations and inquiries about the operating system, see its online help information.

This document covers universal information about generally managed components. The notes and restrictions on use of each product as a managed component are explained in the user's guide provided with the managed component.

Names used with screen images in this document are fictitious. They are unrelated to existing product names, names of organizations, or individual names. The setting values on the screen images are shown as examples, so setting values such as IP addresses on screen images are not guaranteed for operation.

■ About Symbols in This Document

The following explains three symbols that are used in this document:

IMPORTANT:	Points that are mandatory or require attention when using the software or the component.
CHECK:	Points that are require confirmation when using the software or the component.
TIP:	Helpful and convenient piece of information.

■ For other information about the NEC ESMPRO Manager

See the documents below.

NEC ESMPRO Manager Ver.6 Installation Guide
NEC ESMPRO Manager Ver.6 Setup Guide
NEC ESMPRO Manager Ver.6 Command Line Interface
NEC ESMPRO Manager Ver.6 Command Line Interface User’s Guide for NEC ExpressUpdate

Chapter1 About RESTful API

1.1 Overview

This document explains how to use RESTful API of NEC ESMPRO Manager.

IMPORTANT:

- NEC ESMPRO Manager Ver6.20 and later support REST API. Earlier version does not support the REST API. Please confirm it is Ver6.20 or later version before using REST API of NEC ESMPRO Manager.

1.2 How to use the RESTful API

To access the NEC ESMPRO Manager REST API, you need to send HTTP request according to the request form which is described later. The NEC ESMPRO Manager will return a JSON formatted response for the request.

IMPORTANT:

- To access the REST API, you should log in and establish a session. Please see 1.2.1.4 for more details.

1.2.1 Request Format

1.2.1.1 Endpoint

The following is request URL of REST API.

http://*<computer name or IP address of NEC ESMPRO Manager>*:*<port number>*/esmpro/api/

Example:

http://192.168.1.1:21112/esmpro/api

TIP:

- To change the port number of Web API, see "NEC ESMPRO Manager Ver.6 Installation Guide" for more details.

CHECK:

- If you change the setting of NEC ESMPRO Manager to use SSL, change "http" to "https" in request URL.

1.2.1.2 URL and Method

The URLs and Support Method (HTTP Command) of REST API which NEC ESMPRO Manager provides are referred to as follows.

URL	Support Method (HTTP Command)			
/esmpro/api/login-session	POST			DELETE
/esmpro/api/components	POST	GET		
/esmpro/api/components/{jobid}/result		GET		
/esmpro/api/components/server/connection-settings/{guid}		GET	PUT	
/esmpro/api/components/server/connection-settings/{guid}?force={true or false}				DELETE
/esmpro/api/components/server/power-control/{guid}		GET	PUT	
/esmpro/api/components/server/sensor-all/{guid}		GET		
/esmpro/api/components/server/sensor-all/{jobid}/result		GET		
/esmpro/api/components/server/sensor/{guid}?identifier={sensorId}		GET		
/esmpro/api/components/server/system-info/{guid}		GET		
/esmpro/api/components/server/hw/{guid}		GET		
/esmpro/api/components/server/networkinterface/{guid}		GET		
/esmpro/api/components/server/networkinterface/{jobid}/result		GET		
/esmpro/api/components/server/sel/{guid}		GET		
/esmpro/api/components/server/sel/{jobid}/result		GET		
/esmpro/api/components/server/fru-list/{guid}		GET		
/esmpro/api/components/server/fru/{guid}?fruid={fruid}		GET		
/esmpro/api/components/server/connection-check			PUT	
/esmpro/api/components/server/connection-check/{jobid}/result		GET		
/esmpro/api/event?recordId={ID}&severity={All or Information or Minor or Major}		GET		
/esmpro/api/job/status/{jobid}		GET		DELETE
/esmpro/api/sm		GET		
/esmpro/api/components/server/server-status/{guid}		GET		
/esmpro/api/components/server/server-status?name={srvname}		GET		
/esmpro/api/components/server/storage/{guid}		GET		
/esmpro/api/components/server/ilo-storage/{guid}		GET		
/esmpro/api/eem	POST	GET		
/esmpro/api/eem/{jobid}/result		GET		
/esmpro/api/eem?name={name}				DELETE
/esmpro/api/components/server/iml/{guid}		GET		
/esmpro/api/components/server/iml/{jobid}/result		GET		
/esmpro/api/groups/status?name={groupname}		GET		
/esmpro/api/groups/status/{jobid}/result		GET		

CHECK:

- The GUID of target component is set to {guid}.
- Use "-"(hyphen) or ":"(colon) as the separator of GUID.

TIP:

- There are 2 ways to refer to GUID of target component.

  - via REST API:
  By using Get Component List REST API, client can refer to GUID list registered on NEC ESMPRO Manager.

  - via NEC ESMPRO Manager Web GUI
  By accessing the following screen on NEC ESMPRO Manager Web GUI, client can refer to GUID of target component.
  [Constitution] tab - [Server Status] - [Product Information] - [GUID]

1.2.1.3 HTTP Header

REST API supports the following HTTP Header.

Header Field	Description
Cookie	In order to access REST API, client must add "Cookie" field to HTTP header and include a session ID in this field. Session ID is a unique string for login session. This header is necessary except for Login REST API. e.g. Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version	This header specifies API Version of NEC ESMPRO Manager REST API to secure the compatibility. If this header is not included, it works as the latest version is specified. * NEC ESMPRO Manager REST API can specify only version "1.0". e.g. X-ESMPRO-API-Version:1.0
Content-Type	This header specifies the media type of request body in HTTP request. NEC ESMPRO Manager REST API supports JSON format and utf-8, so include the header like an example. * GET/DELETE operations which has no request body will not require this header. e.g. Content-Type:application/json; charset=utf-8

1.2.1.4 Authentication and Session

To access REST API, you must log in and establish a session. To log in, perform HTTP POST to the URI of Login-session REST API.
Login-session REST API authorizes specified user account and password and confirms if the account has the privilege to do specified operation.

CHECK:

- NEC ESMPRO Manager Web GUI restricts Site Access by IP Address. The access via REST API is also restricted by IP Address, so you should add IP Address of REST Client to allow to access.

If the session is created successfully, the session ID is included in the response body of Login-session REST API request. REST Client can access the REST API using the same session by including the session ID in HTTP header while the session is valid.
Session ID will be invalid if you log out by step(4) in the following procedure, or if there is no access to REST API for more than 30 minutes.

The procedure to authorize and manage the session is as follows:

(1)REST Client performs the HTTP POST request to the URI of Login-session REST API including a user account and password of NEC ESMPRO Manager in request body.

(2)If a session is created successfully, REST API returns the response body including the session ID.

(3)REST Client should perform each HTTP requests including the session ID supplied by the login response in HTTP header.

(4)If REST Client wants to delete the session, REST Client performs the HTTP DELETE request to the URI of Login-session REST API including the session ID in HTTP header. Then the session ID becomes invalid.

1.2.2 Response Format

1.2.2.1 HTTP Status Code

HTTP Status Return Codes indicate if performed REST API succeeded or not.

Code	Status	Description
200	OK	Successful operation
400	Bad Request	Request is not correct. For example, Request Parameter is not correct.
401	Unauthorized	Access the resource without the appropriate credential.
403	Forbidden	Forbidden to perform the specified operation or the session ID in HTTP Header is incorrect.
404	Not Found	The specified resource is not found.
405	Method Not Allowed	The resource does not support the specified method.
500	Internal Server Error	Unexpected error has occurred during performing REST API.

1.2.2.2 Error Format

If an error has occurred performing a REST API, the REST API returns an error information in the following format.

```
HTTP/1.1    500 Internal Server Error
  Content-Type: application/json; charset=utf-8

  {
      "errorCode": <Error Code>,
      "errorMessage": "<Error Message>"
  }
```

Error codes and error messages are as follows:

Error Code	Error Message	Note
1000	*There are multiple messages corresponding to error code 1000. Please refer to actual message.	This error code will be returned when illegal request parameters are specified.
1001	The specified job cannot be found.	
1003	Permission denied	
1005	Session ID is invalid	
1006	Authentication Error	
1007	Management settings for obtaining the information is invalid.	
1008	The specified job is not completed successfully.	
1014	The specified component cannot be found.	
1015	The specified id is invalid.	
1017	Directory Service connect Error.	
1018	Sensor Identifier is not found.	
1020	The specified groupName is not found.	
1021	The range of IP address is too large.	
1022	The other process of component registration is running.	
1025	Session have already been authenticated.	
1026	The set API version is not compatible with the current API version.	
1027	HTTP header is invalid.	
1029	The method is not allowed.	
1032	An unrecognized parameter was found in the JSON data.	
1033	Failed to parse the JSON data.	
1034	Enter {0}	
1035	{0} is illegal or invalid.	
1036	{0} must be specified with alphanumeric characters.	
1037	{0} must be {1} characters or more.	
1038	{0} must be {1} characters or less.	
1039	{0} must be in the range from {1} to {2}.	
1040	{0} must be specified with numeric characters.	
1041	The IP address is invalid.	
1042	SNMP management status is not valid.	
1043	OS management status is not valid.	
1044	{0} must be in the range from {1} characters to {2} characters.	

1045	This API isn't supporting it with the present OS.	
1046	{0} isn't being supported with the present OS.	
1047	Target machine does not support {0} feature	
1048	Cannot read cache file	
1049	Cannot execute the operation to the specified device.	
1050	The specified device is not found.	
1051	The component does not support alert test.	
1052	Access is not permitted.	
1053	{0} must be {1} or less.	
1054	Management Controller type is mismatch.	
1055	The user name or password of Management Controller is not entered.	
1056	The user name or password of Management Controller is illegal length.	
2012	Internal Application Error	
2013	Failed to read a JSON data from the request body.	
2014	Failed to generate a JSON data.	
2015	Failed to write a JSON data to the response body.	
3002	Internal Application Error	
3003	Failed in execution of the job.	
3004	Interrupt occurred to the job.	
4003	Too many components exist.	
4007	Failed to register the component.	
4011	Failed to delete the component.	
4012	Failed to get the sensor status.	
4013	The auto registration process failed.	
4014	Failed to get the event information.	
4015	Failed to get the Information.	
4016	Failed to release the component.	
4017	Failed to set the connection settings.	
4018	Failed to control the power status.	
4019	Request failed because connection check was being executed.	
4020	Request failed because installation process was being executed by NEC ExpressUpdate Agent.	
4021	Failed to execute the RAID operation.	
4022	Failed to execute ping process.	
4023	Failed to get the result for ping process.	
4024	Failed to get alert settings.	
4025	Failed to set the alert settings.	
4026	Failed to request the alert test.	
4027	Failed to get the result for the alert test.	
4028	Failed to get the resource URI.	
9000	An application error occurred.	

1.2.3 Asynchronous REST API

NEC ESMPRO Manager provides 2 kinds of REST API, the synchronous type and the asynchronous type. This chapter describe the asynchronous REST API.

1.2.3.1 How to get the resource of Asynchronous REST API

How to get the resource of the asynchronous REST API is as follows:

(1) Client sends an asynchronous request of REST API.

for example)
request: HTTP GET /esmpro/api/components/server/iml/{GUID}
response body: {"url":"/esmpro/api/job/status/iml00001"}

(2) If the request in (1) is received successfully, server returns the response body which includes the URI Location of job status.

(3) In order to check the status of REST API job, client sends HTTP GET request to the URI of "Get Job Status REST API" retrieved by step (2). If the specified job is not executed, HTTP GET request fails.

for example)
request: HTTP GET /esmpro/api/job/status/iml00001
response body: {"description":"Get IML","jobStatus":"Running","errorCode":0,"errorMessage":""}
<u>when the job is completed:</u>
response body:
{"description":"Get IML",
"jobStatus":"Completed",
"errorCode":0,"errorMessage":"",
"url":"/esmpro/api/components/server/iml/iml00001/result"}

(4) Client can get the job status in the response body which is returned by step (2). Until the job status becomes "Completed", check the job status by sending HTTP GET to the URI of "Get Job Status REST API".

(5) When the job status becomes "completed", send HTTP GET to URL which is included in the response body in order to get the job result.

for example)
request: HTTP GET /esmpro/api/components/server/iml/iml00001/result

1.2.3.2 How to cancel Asynchronous REST API

You can cancel the asynchronous REST API request during running.
How to cancel the asynchronous REST API request is as follows:

CHECK:

- You may not be able to cancel the job request depending on the job type. So confirm if the request can be canceled or not by referring to the response body for the "Cancel Job REST API".

(1) Client sends an asynchronous REST API.
for exmaple)
request: HTTP GET /esmpro/api/components/server/iml/{GUID}
response body: {"url":"/esmpro/api/job/status/iml00001"}

(2) If the request in step (1) is received successfully, server returns the response body which includes the URL of job status.

(3) In order to cancel the job request, send HTTP DELETE request to the URI which is returned by step (2). If the specified job is not executed, HTTP DELETE request fails.
for example)
request: HTTP DELETE /esmpro/api/job/status/iml00001
when the cancel succeeds:
response body: {"result": true}
when the cancel fails:
response body: {"result": false}

(4) Server returns the cancel result, true or failure, in the response body for the HTTP DELETE request.

Chapter2 API Reference

2.1 Job Management REST API

When you request the asynchronous REST API, you check the job status and get the job result by using Job Management REST API. This section explains about each Job Management REST API.

2.1.1 Get Job status

This API gets the job status of asynchronous API.

URL:

GET /esmpro/api/job/status/*{jobid}*

Request:

Nothing

ResponseBody:

Key	Value
Description	REST API name
jobStatus	Job Status "Waiting" : Waiting a job start "Running" : Running "Cancel" : Cancel "Completed" : Normal end "Error" : Abnormal end
errorCode	Error code
errorMessage	Error detail message
url	Asynchronous API is returned URL

Example:

Request

```
GET /esmpro/api/job/status/eem00125
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "description" : " Discovery ExpEther Manager",
   "jobStatus" : "Completed",
   "errorCode" : 0,
   "errorMessage" : "",
   "url" : "/esmpro/api/eem/eem00125/result"
}
```

2.1.2 Get Job result

This API gets the job result of asynchronous API.
Refer to chapter in each asynchronous API for details of this API.

2.1.3 Cancel Job

This API cancels asynchronous API.

URL:

DELETE /esmpro/api/job/status/*{jobid}*

Request:

Nothing

ResponseBody:

Key	Value
Result	true : Cancel succeeded false : Cancel failed

Example:

Request
DELETE /esmpro/api/job/status/eem00125 Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083 X-ESMPRO-API-Version:1.0

Response

HTTP 1.1 200 OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8
 	
{
"result": true
}

2.2 REST API

This section explains about each REST API ignore Job Management REST API.

2.2.1 Login

Login for REST API authentication is performed.

URL:

POST /esmpro/api/login-session

Request:

Key	Value
user	User name of NEC ESMPRO Manager account.
password	Password of NEC ESMPRO Manager account.

ResponseBody:

Key	Value
sessionId	The ID to distinguish a session when accessing REST API.

Example:

Request

```
POST /esmpro/api/login-session
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "user":"loginuser",
   "password":"password123"
}
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "sessionId":"206C9F1D25E7AB9E1F1AFAA8AC51B083"
}
```

2.2.2 Logout

REST API authentication is released and session ID is invalidated.

URL:

```
DELETE /esmpro/api/login-session
```

Request:

Nothing

ResponseBody:

Nothing

Example:

Request

```
DELETE /esmpro/api/login-session
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
```

2.2.3 Auto Registration

This API searches the network for components and registers them as management targets on NEC ESMPRO Manager.

URL:

POST /esmpro/api/components

Request:

Key	Value
groupName	Group name to register a component. (1-63 characters) JSON Type : String If this key is omitted, detected components are registered in "root" group.
discoveryMode	IP address range search or a network address search.[Mandatory] JSON Type : Integer 0 : IP Address Range Search 1 : Network Address Search
startAddress	Start IP address (If discoveryMode is Network Address Search, this key is ignored.) JSON Type : String Enter the number of 4 sets of 0-255 divided by "."
endAddress	End IP address (If discoveryMode is Network Address Search, this key is ignored.) JSON Type : String Enter the number of 4 sets of 0-255 divided by "."
networkAddress	Network address (If discoveryMode is IP Address Range Search, this key is ignored.) JSON Type : String Enter the number of 4 sets of 0-255 divided by "."
networkMask	Network mask (If discoveryMode is IP Address Range Search, this key is ignored.) JSON Type : String Enter the number of 4 sets of 0-255 divided by "."
snmpManagementValid	Specify whether to register a component with management that uses NEC ESMPRO ServerAgent.[Mandatory] JSON Type : Boolean true if SNMP management is enabled, false otherwise
snmpCommunity	SNMP community name used to manage NEC ESMPRO ServerAgent. (Alphanumeric single-byte and sign (!#$%&()*+-./:;<>=?@[^_`{|}~]) of 1-255 characters) JSON Type : String If "snmpManagementValid" is true, this key is [Mandatory]
wsManagementValid	Specify whether to register a component with management that uses NEC ESMPRO ServerAgentService or VMware ESXi7 or earlier.[Mandatory] JSON Type : Boolean true if WS-Man management is enabled, false otherwise
wsManAccount	Array of user credential information of Windows or VMware ESXi7 or earlier. It is possible to specify it up to five. If "wsManagementValid" is true, this key is [Mandatory] JSON Type : Object
user	User name of Windows or VMware ESXi7 or earlier. (1-255 characters) JSON Type : String

	If "wsManagementValid" is true, this key is [Mandatory]
password	Password of Windows or VMware ESXi7 or earlier. (1-255 characters) JSON Type : String If "wsManagementValid" is true, this key is [Mandatory]
esxiManagementValid	Array of user credential information of Windows or VMware ESXi8. It is possible to specify it up to five. JSON Type : Boolean true if ESXi management is enabled, false otherwise
esxiAccount	Array of user credential information of Windows or VMware ESXi8. It is possible to specify it up to five. If "esxiManagementValid" is true, this key is [Mandatory] JSON Type : Object
user	User name of Windows or VMware ESXi8. (1-256 characters) If "esxiManagementValid" is true, this key is [Mandatory] JSON Type : String
password	Password of Windows or VMware ESXi8. (1-40 characters) If "esxiManagementValid" is true, this key is [Mandatory] JSON Type : String
raidManagementValid	Specify whether to register a component with management of RAID system.[Mandatory] JSON Type : Boolean true if RAID system management is enabled, false otherwise
euManagementEuaValid	Specify whether to register a component with management that uses NEC ExpressUpdate(Updates via NEC ExpressUpdate Agent).[Mandatory] JSON Type : Boolean true if ExpressUpdate(Updates via NEC ExpressUpdate Agent) management is enabled, false otherwise
bmcManagementValid	Specify whether to register a component with management that uses BMC.[Mandatory] JSON Type : Boolean true if BMC management is enabled, false otherwise
euManagementBmcValid	Specify whether to register a component with management that uses NEC ExpressUpdate(Updates via Management Controller). JSON Type : Boolean If "bmcManagementValid" is true, this key is [Mandatory] true if NEC ExpressUpdate(Updates via Management Controller) management is enabled, false otherwise
authKey	Authentication key to communicate with BMC of component. It is possible to specify it up to five. (1-255 characters) JSON Type : String If "bmcManagementValid" is true, this key is [Mandatory]
vproManagementValid	Specify whether to register a component with management that uses vPro.[Mandatory] JSON Type : Boolean true if vPro management is enabled, false otherwise
vproAccount	Array of account information to communicate with vPro of component. It is possible to specify it up to five. JSON Type : Object If "vproManagementValid" is true, this key is [Mandatory]
user	User name to communicate with vPro. (1-255 characters) JSON Type : String If "vproManagementValid" is true, this key is [Mandatory]
password	Password to communicate with vPro. (1-255 characters) JSON Type : String

		If "vproManagementValid" is true, this key is [Mandatory]
iloManagementValid		Specify whether to register a component with management that uses iLO.[Mandatory] JSON Type : Boolean true if iLO management is enabled, false otherwise
iloAccount		Array of account information to communicate with iLO of component. It is possible to specify it up to five. JSON Type : Object If "iloManagementValid" is true, this key is [Mandatory]
	user	User name to communicate with iLO. (1-39 characters) JSON Type : String If "iloManagementValid" is true, this key is [Mandatory]
	password	Password to communicate with iLO. (1-39 characters) JSON Type : String If "iloManagementValid" is true, this key is [Mandatory]
otherBmcManagementValid		Specify whether to register a component with management that uses BMC (Other).[Mandatory] JSON Type : Boolean true if BMC (Other) management is enabled, false otherwise
otherBmcAccessMode		Select the method to access BMC (Other). If not specified, the default value is "0: Auto". JSON Type : Integer 0:Auto 1:Redfish 2:IPMI
otherBmcAccount		Array of account information to communicate with BMC (Other) of component. It is possible to specify it up to five. JSON Type : Object If "otherBmcManagementValid" is true, this key is [Mandatory]
	user	User name to communicate with BMC (Other). (1-255 characters) JSON Type : String If "otherBmcManagementValid" is true, this key is [Mandatory]
	password	Password to communicate with BMC (Other). (1-255 characters) JSON Type : String If "otherBmcManagementValid" is true, this key is [Mandatory]

ResponseBody:

Key	Value
url	The URL to get the job status. JSON Type : String /esmpro/api/job/status/*{jobid}*

Example:

Request

```
POST /esmpro/api/components
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "groupName" : "Group0002",
   "discoveryMode" : 0,
   "startAddress" : "192.168.1.1",
   "endAddress" : "192.168.1.20",
   "snmpManagementValid" : false,
```

```
    "wsManagementValid" : true,
    "wsManAccount" : [
        {
            "user" : "Administrator",
            "password" : "Administrator"
        },
        {
            "user" : "user",
            "password" : "password"
        }
    ],
    "esxiManagementValid" : true,
    "esxiAccount" : [
        {
            "user" : "root",
            "password" : "root123"
        }
    ],
    "raidManagementValid" : false,
    "euManagementEuaValid" : false,
    "bmcManagementValid" : true,
    "euManagementBmcValid" : true,
    "authKey" : [
        "guest"
    ],
    "iloManagementValid" : true,
    "iloAccount" : [
        {
            "user" : "Administrator",
            "password" : "Administrator"
        }
    ],
    "vproManagementValid" : false
    "otherBmcManagementValid" : false
}
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
    "url" : "/esmpro/api/job/status/arc00001"
}
```

This API is one of the asynchronous REST API. So check the job status by sending HTTP GET request to the URL in response body. Refer to 2.1.1 for more detail.

When the job status becomes "completed", you can get the job result by sending HTTP GET to the following URL.

URL:

GET /esmpro/api/components/*{jobid}*/result

Request:

Nothing

ResponseBody:

Key	Value
registrationResult	Array with data of automatic registration result JSON Type : Object
guid	GUID of registered component JSON Type : String
ipAddress	OS IP address of registered component JSON Type : String It is not displayed if the all system managements are not registered.
bmcIpAddress	Management controller IP address of registered component JSON Type : String It is not displayed if the management controller management is not registered.
bmcManagementValid	It shows whether the BMC management function is valid or invalid. JSON Type : String 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
euManagementBmcValid	It shows whether NEC ExpressUpdate (Updates via Management Controller) management function is valid or invalid. JSON Type : String 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
euManagementEuaValid	It shows whether NEC ExpressUpdate (Updates via NEC ExpressUpdate Agent) management function is valid or invalid. JSON Type : String 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
raidManagementValid	It shows whether RAID system management function is valid or invalid. JSON Type : String 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
snmpManagementValid	It shows whether SNMP management function is valid or invalid. JSON Type : String 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
wsManagementValid	It shows whether WS-Man management function is valid or invalid. JSON Type : String 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
esxiManagementValid	It shows whether ESXi management function is valid or invalid. JSON Type : String 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
vproManagementValid	It shows whether vPro management function is valid or invalid. JSON Type : String

		1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
	iloManagementValid	It shows whether iLO management function is valid or invalid. JSON Type : String 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
	otherBmcManagementValid	It shows whether BMC (Other) management function is valid or invalid. JSON Type : String 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>

Example:

Request

```
GET /esmpro/api/components/arc00001/result
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "registrationResult" : [
      {
         "guid": "5f3cc680-cf1b-11e0-8001-00255cc64b2e",
         "ipAddress" : "192.168.14.2",
         "bmcIpAddress" : "192.168.14.3",
         "bmcManagementValid" : "3",
         "euManagementBmcValid" : "1",
         "euManagementEuaValid" : "3",
         "raidManagementValid" : "3",
         "snmpManagementValid" : "3",
         "vProManagementValid" : "1",
         "wsManagementValid" : "1",
         "esxiManagementValid" : "1",
         "iloManagementValid" : "3"
         "otherBmcManagementValid" : "1"
      },
      {
         "guid": "5f3cc670-cf1a-11d0-8201-00255ac65b3e",
         "ipAddress" : "192.168.14.10",
         "bmcManagementValid" : "1",
         "euManagementBmcValid" : "1",
         "euManagementEuaValid" : "3",
         "raidManagementValid" : "3",
         "snmpManagementValid" : "1",
         "wsManagementValid" : "3",
         "esxiManagementValid" : "1",
         "vproManagementValid" : "1"
         "iloManagementValid" : "3"
         "otherBmcManagementValid" : "1"
      }
   ]
}
```

2.2.4 Get Component List

This API gets the list of components registered on NEC ESMPRO Manager.

URL:

```
GET /esmpro/api/components
```

Request:

Nothing

ResponseBody:

Key		Value
components		Array of management information every components JSON Type : Array
	guid	GUID of component JSON Type : String (Character string of the GUID form. It is consisted of alphanumeric single-byte and '-') 35 character fixing
	bmcIpAddress	Management controller IP address. It is returned empty string ("") if IP address using to manage the management controller is not set. JSON Type : String Enter the number of 4 sets of 0-255 divided by "."
	osIpAddress	OS IP address of a component. It is returned empty string ("") if OS IP address using to manage the component is not set. JSON Type : String Enter the number of 4 sets of 0-255 divided by "."
	serverId	Component ID NEC ESMPRO Manager manages JSON Type : Integer

Example:

Request

```
GET /esmpro/api/components
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "components": [
      {
         "guid": "5f3cc680-cf1b-11e0-8001-00255cc64b2e",
         "bmcIpAddress":"192.168.1.1",
         "osIpAddress":"10.34.123.1",
         "serverId":1
      },
      {
         "guid": "5f3cc680-cf1b-11e0-8001-001234567890",
         "bmcIpAddress":"192.168.1.2",
         "osIpAddress":"",
         "serverId":2
      }
   ]
}
```

2.2.5 Get Connection Setting

This API gets connection setting of a target component.
Connection setting is saved in NEC ESMPRO Manager in order to manage the target component.

URL:

GET /esmpro/api/components/server/connection-settings/*{guid}*

Request:

Nothing

ResponseBody:

Key	Value
name	Component name JSON Type : String
groupName	Belonging group JSON Type : String
bmcManagementValid	State of BMC management function JSON Type : Integer 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
bmcIpAddress	BMC IP address JSON Type : String
bmcSubnetMask	Subnet mask of BMC LAN JSON Type : String
vproManagementValid	State of vPro management function JSON Type : Integer 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
vproUser	User name of vPro management JSON Type : String
vproIpAddress	IP address of vPro management JSON Type : String
vproSubnetMask	Subnet mask of vPro management JSON Type : String
iloManagementVallid	State of iLO management function JSON Type : Integer 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
iloUser	User name of iLO management JSON Type : String
iloIpAddress	IP address of iLO management JSON Type : String
iloSubnetMask	Subnet mask of iLO management JSON Type : String
iloProtocol	Protocol used by communication with iLO (HTTP/HTTPS) JSON Type : String
iloPort	Port number used by communication with iLO JSON Type : String
otherBmcManagementVallid	State of BMC (Other) management function JSON Type : Integer

	1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
otherBmcAccessMode	Access mode to BMC (Other) JSON Type : Integer 0 : Auto 1 : Redfish 2 : IPMI
otherBmcPort	Port number used when the BMC (Other) access mode is Redfish JSON Type : Integer Default value : 443
otherBmcUser	User name of BMC (Other) management JSON Type : String
otherBmcIpAddress	IP address of BMC (Other) management JSON Type : String
ipAddress	OS IP address of a component JSON Type : String
snmpManagementValid	State of SNMP management function JSON Type : Integer 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
snmpComunityNameGet	SNMP community name for read-only (GET) operations. It is used when getting information from SNMP Agent. JSON Type : String
snmpComunityNameSet	SNMP community name for read-write (SET) operations. It is used when setting information to SNMP Agent. If SNMP community name for setting is not set, it is returned empty string (""). JSON Type : String
wsManagementValid	State of WS-Man management function JSON Type : Integer 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
wsmanUser	User name of WS-Man management JSON Type : String
wsmanProtocol	Communication protocol of WS-Man management (HTTP/HTTPS) JSON Type : String
wsmanPort	Port number of WS-Man management JSON Type : String
esxiManagementValid	State of ESXi management function JSON Type : Integer 1 : Not Registered 2 : Registered<invalid> 3 : Registered<valid>
esxiUser	User name of ESXi management JSON Type : String
systemMagementType	The type of Agent managed by the SNMP management or the WS-Man management JSON Type : String
osStatusWatch	Alive monitoring of OS (used SNMP) JSON Type : Boolean true if alive monitoring is enabled, false otherwise
registerAlertByStatus	It is shown whether to register the alert when it is no response from a component by alive monitoring of SNMP, and recovering from the

	state. JSON Type : Boolean true if registering alert is enabled, false otherwise
osStatusPingWatch	Alive monitoring of OS (used ICMP Ping) JSON Type : Boolean true if alive monitoring is enabled, false otherwise
registerAlertByPingStatus	It is shown whether to register the alert when it is no response from a component by alive monitoring of ICMP Ping, and recovering from the state. JSON Type : Boolean true if registering alert is enabled, false otherwise
osStatusWatchInterval	Monitoring interval (minute) of alive monitoring used SNMP and ICMP Ping to OS. JSON Type : Integer
osStatusWatchRetryCount	Retry count of alive monitoring used SNMP and ICMP Ping to OS. JSON Type : Integer

Example:

Request

```
GET /esmpro/api/components/server/connection-settings/5f3cc680-cf1b-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "name" : "Server0001",
   "groupName" : "Group0002",
   "bmcManagementValid" : 3,
   "bmcIpAddress" : "192.168.14.2",
   "bmcSubnetMask" : "255.255.255.0",
   "vproManagementValid " : 1,
   "vproUser" : "",
   "vproIpAddress" : "",
   "vproSubnetMask" : "",
   "iloManagementValid " : 1,
   "iloUser" : "",
   "iloIpAddress" : "",
   "iloSubnetMask" : "",
   "iloProtocol" : "",
   "iloPort" : "",
   "otherBmcManagementValid " : 1,
   "otherBmcAccessMode" : 1,
   "otherBmcPort" : 443,
   "otherBmcUser" : "",
   "otherBmcIpAddress" : "",
   "euManagementEuaValid" : 3
   "euManagementBmcValid" : 2,
   "raidManagementValid" : 2,
   "ipAddress" : "192.168.14.3",
   "snmpManagementValid" : 1,
   "snmpComunityNameGet" : "public",
   "snmpComunityNameSet" : "",
   "wsManagementValid" : 1,
   "wsmanUser" : "",
   "wsmanProtocol" : "",
   "wsmanPort" : "",
   "esxiManagementValid" : 1,
   "esxiUser" : "",
   "osStatusWatch" : false,
   "osStatusPingWatch" : true,
   "osStatusWatchInterval" : 2,
   "osStatusWatchRetryCount" : 0,
   "systemManagementType" : "ServerAgent"
}
```

2.2.6 Set Connection Setting

This API sets connection setting of a target component.

URL:

PUT /esmpro/api/components/server/connection-settings/*{guid}*

Request:

Key	Value
name	Component name JSON Type : String (Alphanumeric single-byte, dot (.), hyphen (-) and underscore (_) of 1-63 characters. The big and small character is classified.)
groupName	Belonging group JSON Type : String (1-63 characters)
bmcManagementValid	State of BMC management function JSON Type : Integer 1 : invalid 2 or 3 : valid
bmcIpAddress	BMC IP address JSON Type : String (Enter the number of 4 sets of 0-255 divided by ".")
bmcSubnetMask	Subnet mask of BMC LAN JSON Type : String (Enter the number of 4 sets of 0-255 divided by ".")
bmcAuthKey	BMC auth key JSON Type : String (Alphanumeric single-byte of 1-20 characters)
vproManagementValid	State of vPro management function JSON Type : Integer 1 : invalid 2 or 3 : valid
vproUser	User name of vPro management JSON Type : String (1-16 characters)
vproPassword	Password of vPro management JSON Type : String (1-32 characters)
vproIpAddress	IP address of vPro management JSON Type : String (Enter the number of 4 sets of 0-255 divided by ".")
vproSubnetMask	Subnet mask of vPro management JSON Type : String (Enter the number of 4 sets of 0-255 divided by ".")
iloManagementValid	State of iLO management function JSON Type : Integer 1 : invalid 2 or 3 : valid
iloUser	User name of iLO management JSON Type : String (1-39 characters)
iloPassword	Password of iLO management JSON Type : String (1-39 characters)

iloIpAddress	IP address of iLO management JSON Type : String (Enter the number of 4 sets of 0-255 divided by ".")
iloSubnetMask	Subnet mask of iLO management JSON Type : String (Enter the number of 4 sets of 0-255 divided by ".")
iloProtocol	Protocol used by communication with iLO (HTTP/HTTPS) JSON Type : String Default value : "HTTPS"
iloPort	Port number used by communication with iLO JSON Type : String (Alphanumeric single-byte of 1-5 characters) Default value : "80" for HTTP "443" for HTTPS Default value is depending on the value of "iloProtocol".
otherBmcManagementValid	State of BMC (Other) management function JSON Type : Integer 1 : invalid 2 or 3 : valid
otherBmcAccessMode	Access mode to BMC (Other) JSON Type : Integer 0 : Auto 1 : Redfish 2 : IPMI
otherBmcPort	Port number used when the BMC (Other) access mode is Redfish JSON Type : Integer
otherBmcUser	User name of BMC (Other) management JSON Type : String (1-16 characters)
otherBmcPassword	Password of BMC (Other) management JSON Type : String (1-20 characters)
otherBmcIpAddress	IP address of BMC (Other) management JSON Type : String (Enter the number of 4 sets of 0-255 divided by ".")
snmpManagementValid	State of SNMP management function JSON Type : Integer 1 : invalid 2 or 3 : valid
snmpComunityNameGet	SNMP community name for read-only (GET) operations. It is used when getting information from SNMP Agent. JSON Type : String (Alphanumeric single-byte and sign (!#$%&()*+-./:;<>=?@[^_`{|}~] of 1-255 characters) Default value : "public"
snmpComunityNameSet	SNMP community name for read-write (SET) operations. It is used when setting information to SNMP Agent. If SNMP community name for setting is not set, it is used SNMP community name for getting. JSON Type : String (Alphanumeric single-byte and sign (!#$%&()*+-./:;<>=?@[^_`{|}~]) of 1-255 characters)
wsManagementValid	State of WS-Man management function JSON Type : Integer 1 : invalid 2 or 3 : valid

wsmanUser	User name of WS-Man management JSON Type : String (1-31 characters)
wsmanPassword	Password of WS-Man management JSON Type : String (1-255 characters)
wsmanProtocol	Communication protocol of WS-Man management (HTTP/HTTPS) JSON Type : String Default value : "HTTP"
wsmanPort	Port number of WS-Man management JSON Type : String (Alphanumeric single-byte of 1-5 characters) Default value : "5985/5986"
esxiManagementValid	State of ESXi management function JSON Type : Integer 1 : invalid 2 or 3 : valid
esxiUser	User name of ESXi management JSON Type : String (1-256 characters)
esxiPassword	Password of ESXi management JSON Type : String (1-40 characters)
osStatusWatch	State of Alive monitoring of OS (used SNMP) JSON Type : Boolean true if alive monitoring is enabled, false otherwise
registerAlertByStatus	It is shown whether to register the alert when it is no response from a component by alive monitoring of SNMP, and recovering from the state. JSON Type : Boolean true if registering alert is enabled, false otherwise Defalut value : false
osStatusPingWatch	Alive monitoring of OS (used ICMP Ping) JSON Type : Boolean true if alive monitoring is enabled, false otherwise
registerAlertByPingStatus	It is shown whether to register the alert when it is no response from a component by alive monitoring of ICMP Ping, and recovering from the state. JSON Type : Boolean true if registering alert is enabled, false otherwise Defalut value : false
osStatusWatchInterval	Monitoring interval (miniute) of alive monitoring used SNMP and ICMP Ping to OS. JSON Type : Integer Defalut value : 1
osStatusWatchRetryCount	Retry count of alive monitoring used SNMP and ICMP Ping to OS. (0-100 count) JSON Type : Integer Defalut value : 0

ResponseBody:

Nothing

Example:

Request

```
PUT /esmpro/api/components/server/connection-settings/5f3cc680-cf1b-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "name" : "Server0001",
   "groupName" : "Group0002",
   "bmcManagementValid" : 3,
   "bmcIpAddress" : "192.168.14.2",
   "bmcSubnetMask" : "255.255.255.0",
   "bmcAuthKey" : "bmcAuthKey123",
   "vproManagementValid" : 1,
   "iloManagementValid" : 1,
   "otherBmcManagementValid" : 1,
   "otherBmcAccessMode" : 1,
   "otherBmcPort":443,
   "euManagementEuaValid" : 3,
   "euManagementBmcValid" : 2,
   "raidManagementValid" : 2,
   "ipAddress" : "192.168.14.3",
   "snmpManagementValid" : 1,
   "wsManagementValid" : 1,
   "esxiManagementValid" : 1,
   "osStatusWatch" : false,
   "registerAlertByStatus" : false,
   "osStatusPingWatch" : true,
   "registerAlertByPingStatus" : true,
   "osStatusWatchInterval" : 2,
   "osStatusWatchRetryCount" : 1
}
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
```

2.2.7 Delete Component

This API deletes a specified component that is registered on NEC ESMPRO Manager.

URL:

DELETE /esmpro/api/components/server/connection-settings/*{guid}?force={true or false}*

If you specify "force", this API delete the component as follows:
true : delete a specified component forcibly.
false : not delete a specified component forcibly.

*If you omit "force", "force" is specified to "false".

*If the value other than "true" and "false" is specified, this API will return error code.

*When you delete the managed component that has been set schedule running, the schedule is deleted from NEC ESMPRO Agent Extension. But if the schedule deletion is failed (such as NEC ESMPRO Agent Extension is uninstalled) and you specify "force=true", this API will delete the component forcibly.

*If you perform this API during "Check Connection" against the specified component, or executing Update of FW/SW on the specified component, this API will fail.

Request:

Nothing

ResponseBody:

Nothing

Example:

Request
DELETE /esmpro/api/components/server/connection-settings/5f3cc680-cf1b-11e0-8001-00255cc64b2e?force=true Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083 X-ESMPRO-API-Version:1.0

Response
HTTP 1.1 200 OK X-ESMPRO-API-Version:1.0 ,

2.2.8 Get Power State

This API gets the power state of a specified component.

URL:

GET /esmpro/api/components/server/power-control/*{guid}*

Request:

Nothing

ResponseBody:

Key	Value
powerState	Power State JSON Type : String "On" "Off" "Unknown"

Example:

Request

```
GET /esmpro/api/components/server/power-control/5f3cc680-cf1b-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "powerState" : "On"
}
```

2.2.9 Power Control

This API requests a specified component to perform power operation. This API only sends the request and does not guarantee power state transition.

URL:

PUT /esmpro/api/components/server/power-control/*{guid}*

Request:

Key	Value
powerState	Power Operation[Mandatory] JSON Type : String Possible operations are as follows: "On": Power On "Off" : Power Off "Reset" : Reset "Shutdown" : Shutdown "PowerCycle" : Power Cycle "Dump" : Push Dump Switch "ForceShutdown" : Shutdown via BMC If you specify the operation other than the above operation, this API will fail.

ResponseBody:

Nothing

Example:

Request

PUT /esmpro/api/components/server/power-control/5f3cc680-cf1b-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8
 	
{
"powerState": "On"
}

Response
HTTP 1.1 200 OK X-ESMPRO-API-Version:1.0

2.2.10 Get All Sensors Information

This API gets all sensors information of a specified component.

URL:

GET /esmpro/api/components/server/sensor-all/*{guid}*

Request:

Nothing

ResponseBody:

Key	Value
url	The URL to get the job status. JSON Type : String /esmpro/api/job/status/*{jobid}*

Example:

Request
GET /esmpro/api/components/server/sensor-all/5f3cc680-cf1b-11e0-8001-00255cc64b2e Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083 X-ESMPRO-API-Version:1.0

Response

HTTP 1.1 200 OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8
 	
{
"url" : "/esmpro/api/job/status/gss00001"
}

This API is one of the asynchronous REST API. So check the job status by sending HTTP GET request to the URL in response body. Refer to 2.1.1 for more detail.

When the job status becomes "completed", you can get the job result by sending HTTP GET to the following URL.

URL:

GET /esmpro/api/components/server/sensor-all/*{jobid}*/result

Request:

Nothing

ResponseBody:

Key	Value
sensor	Array of sensor data. JSON Type : Array
identifier	Sensor Identifier JSON Type : String
recordId	SDR Record ID in hexadecimal JSON Type : String
name	Sensor Name. ID String or the name to identify the sensor. JSON Type : String
sensorType	Sensor Type Name. JSON Type : String
currentValue	Current Value JSON Type : String
upperNonRecoverable	Upper non recoverable JSON Type : String
upperCritical	Upper critical JSON Type : String
upperNonCritical	Upper non critical JSON Type : String
lowerNonRecoverable	Lower non recoverable JSON Type : String
lowerCritical	Lower critical JSON Type : String
lowerNonCritical	Lower non critical JSON Type : String
currentStatus	Current sensor status JSON Type : String In normal case, "Normal" will be returned. In abnormal case, the returned value depends on the definition of SDR. For example, the candidates of returned value of SDR Type01 are as follows: -"Upper Non-critical - going high" -"Upper Critical - going high" -"Upper Non-recoverable - going high" -"Lower Critical - going low" -"Lower Non-critical - going low" -"Lower Non-recoverable - going low"

Example:

Request

```
GET /esmpro/api/components/server/sensor-all/gss00001/result
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "sensor":[
      {
         "identifier": "0001",
         "recordId": "0001",
         "name": "Processor1 VCCIN",
         "sensorType": "Voltage",
         "currentValue": "1.81 Volts",
         "upperNonRecoverable": "---",
         "upperCritical": "2.12 Volts (Hysteresis:2.11 Volts)",
         "upperNonCritical": "2.03 Volts (Hysteresis:2.01 Volts)",
         "lowerNonRecoverable": "---",
         "lowerCritical": "1.24 Volts (Hysteresis:1.26 Volts)",
         "lowerNonCritical": "1.32 Volts (Hysteresis:1.33 Volts)",
         "currentStatus": "Normal"
      },
      {
         "identifier": "0002",
         "recordId": "0002",
         "name": "Processor2 VCCIN",
         "sensorType": "Voltage",
         "currentValue": "Unknown",
         "upperNonRecoverable": "---",
         "upperCritical": "2.12 Volts (Hysteresis:2.11 Volts)",
         "upperNonCritical": "2.03 Volts (Hysteresis:2.01 Volts)",
         "lowerNonRecoverable": "---",
         "lowerCritical": "1.24 Volts (Hysteresis:1.26 Volts)",
         "lowerNonCritical": "1.32 Volts (Hysteresis:1.33 Volts)",
         "currentStatus": "Unknown"
      }
   ]
}
```

2.2.11 Get Sensor Information

This API gets a specified sensor information of a specified component. Specify SDR Record ID in Sensor Identifier.

URL:

GET /esmpro/api/components/server/sensor/*{guid}?identifier={Sensor Identifier}*

Request:

Nothing

ResponseBody:

Key	Value
identifier	Sensor Identifier. JSON Type : String
recordId	SDR Record ID JSON Type : String
name	Sensor Name. ID String or the name to identify the sensor. JSON Type : String
sensorType	Sensor type Name JSON Type : String
currentValue	Current Value JSON Type : String
upperNonRecoverable	Upper non recoverable JSON Type : String
upperCritical	Upper critical JSON Type : String
upperNonCritical	Upper non critical JSON Type : String
lowerNonRecoverable	Lower non recoverable JSON Type : String
lowerCritical	Lower critical JSON Type : String
lowerNonCritical	Lower non critical JSON Type : String
currentStatus	Current sensor status JSON Type : String In normal case, "Normal" will be returned. In abnormal case, the returned value depends on the definition of SDR. For example, the candidates of returned value of SDR Type01 are as follows: -"Upper Non-critical - going high" -"Upper Critical - going high" -"Upper Non-recoverable - going high" -"Lower Critical - going low" -"Lower Non-critical - going low" -"Lower Non-recoverable - going low"

Example:

Request

```
GET
/esmpro/api/components/server/sensor/5f3cc680-cf1b-11e0-8001-00255cc64b2e?identifier=0002
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
        "identifier": "0002",
        "recordId": "0002",
        "name": "Processor2 VCCIN",
        "sensorType": "Voltage",
        "currentValue": "Unknown",
        "upperNonRecoverable": "---",
        "upperCritical": "2.12 Volts (Hysteresis:2.11 Volts)",
        "upperNonCritical": "2.03 Volts (Hysteresis:2.01 Volts)",
        "lowerNonRecoverable": "---",
        "lowerCritical": "1.24 Volts (Hysteresis:1.26 Volts)",
        "lowerNonCritical": "1.32 Volts (Hysteresis:1.33 Volts)",
        "currentStatus": "Unknown"
}
```

2.2.12 Get System Information

This API gets the System Information of a specified component.

URL:

GET /esmpro/api/components/server/system-info/*{guid}*

Request:

Nothing

ResponseBody:

Key	Value
osType	OS Name of component, or OS Name and OS Version of component. JSON Type : String If value cannot be fetched from component then "" (empty string) will be returned.

Example:

Request

```
GET /esmpro/api/components/server/system-info/5f3cc680-cf1b-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "osType" : "Microsoft Windows Server 2022 Standard"
}
```

2.2.13 Get Hardware Information

This API gets Hardware Information of a specified component.

URL:

GET /esmpro/api/components/server/hw/*{guid}*

Request:

Nothing

ResponseBody:

Key	Value
productName	Product Name of the component JSON Type : String If product name cannot be fetched from component then "" (empty string) will be returned.
manufacturerId	Manufacturer ID JSON Type : String If manufacturer ID cannot be fetched from component then "" (empty string) will be returned.
productId	Product ID JSON Type : String
physicalCPU	Physical CPU Information JSON Type : Array
name	CPU Name JSON Type : String If CPU name cannot be fetched from component then "" (empty string) will be returned.
version	Version JSON Type : String If version cannot be fetched from component then "" (empty string) will be returned.
internalSpeed	Internal Speed JSON Type : String If internal speed cannot be fetched from component then "" (empty string) will be returned.
maxCoreNum	Number of Max cores JSON Type : String If Max Core Number cannot be fetched from component then "" (empty string) will be returned.
validCoreNum	Number of effective cores JSON Type : String If effective Core Number cannot be fetched from component then "" (empty string) will be returned.
validThreadNum	Number of effective Thread JSON Type : String If effective Thread Number cannot be fetched from component then "" (empty string) will be returned.
operationalStatus	State of CPU operation JSON Type : Integer 1 - Other 2 - Unknown 3 - Normal 4 - Warning 5 - Abnormal

		6 - Non-recoverable 7 - In test 8 - Degraded 9 - Not attached 10 - Down 11 - Standby
memoryTotalCapacity		Total value of physical memory JSON Type : String If total memory capacity cannot be fetched from component then "" (empty string) will be returned.
memoryBank		Memory Bank Information JSON Type : Array
	size	Memory Bank Size JSON Type : String If memory bank size cannot be fetched from component then "" (empty string) will be returned.
	status	Status of Memory Bank JSON Type : Integer 1 - Other 2 - Unknown 3 - Normal 4 - Warning 5 - Abnormal 6 - Non-recoverable 7 - In test 8 - Degraded 9 - Not attached 10 - Down 11 - Standby

Example:

Request

```
GET /esmpro/api/components/server/hw/5f3cc680-cf1b-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "productName" : "Express5800/R110d-1M [N8100-1807Y]",
   "manufacturerId":"119",
   "productId": "0548h",
   "physicalCPU" : [
      {
         "name" : "Intel(R) Xeon(R) processor",
         "version" : "Intel64 Family 6 Model 44 Stepping 2",
         "internalSpeed" : "2400 MHz ",
         "maxCoreNum" : "4",
         "validCoreNum" : "4",
         "validThreadNum" : "8",
         "operationalStatus" : 3
      }
   ],
   [
      {
         "name" : "Intel(R) Xeon(R) processor",
         "version" : "Intel64 Family 6 Model 44 Stepping 2",
         "internalSpeed" : "2400 MHz ",
         "maxCoreNum" : "4",
         "validCoreNum" : "4",
         "validThreadNum" : "8",
         "operationalStatus" : 3
      }
   ],
   "memoryTotalCapacity" : "8388608 KB ",
   "memoryBank":[
      {
         "size":"4194304 KB",
         "status":3
      }
   ],
   [
      {
         "size":"4194304 KB",
         "status":3
      }
   ]
}
```

2.2.14 Get Network Information

This API gets Network Information of a specified component.

URL:

```
GET /esmpro/api/components/server/networkinterface/{guid}
```

Request:

Nothing

ResponseBody:

Key	Value
url	The URL to get the job status. JSON Type : String /esmpro/api/job/status/*{jobid}*

Notes:

When changing the network adapter of ESXi management, execute REST API of the connection check first. Refer to 2.2.18 for more detail.Correct results may not be returned.

Click the target component from NEC ESMPRO Manager Web GUI to display [Constitution] tab or restarts the service of NEC ESMPRO Manager, too, the right data becomes able to be acquired.

For the order of restarting services, refer to the following section of NEC ESMPRO Manager Ver.6 Installation Guide.

Chapter 4 Appendix

3. Services

Order of starting or stopping services

Example:

Request

```
GET /esmpro/api/components/server/networkinterface/5f3cc680-cf1b-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1 200 OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
  "url" : "/esmpro/api/job/status/gni00001"
}
```

This API is one of the asynchronous REST API. So check the job status by sending HTTP GET request to the URL in response body. Refer to 2.1.1 for more detail.

When the job status becomes "completed", you can get the job result by sending HTTP GET to the following URL.

URL:

```
GET /esmpro/api/components/server/networkinterface/{jobid}/result
```

Request:

Nothing

ResponseBody:

Key		Value
networkInterface		Network Interface Information JSON Type : Array
	id	Network Interface ID JSON Type : Integer If ID cannot be fetched from component then null will be returned.
	macAddress	MAC Address JSON Type : String If MAC Address cannot be fetched from component then "" (empty string) will be returned.
	adapterType	Network Adapter Type JSON Type : Integer 0 : Virtual Network Adapter 1 : Physical Network Adapter 2 : Unknown (If an adapter type cannot be fetched, 2 will be returned.)
	manufacturer	Manufacturer of Network Adapter JSON Type : String If manufacturer cannot be fetched from component then "" (empty string) will be returned.

Example:

Request

```
GET /esmpro/api/components/server/networkinterface/gni00001/result
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   " networkInterface ":[
      {
         "id":1,
         "macAddress": "94:DE:80:52:4F:66",
         "adapterType":0,
         "manufacturer": "Microsoft"
      },
      {
         "id":2
         "macAddress": "94:DE:80:52:4F:67",
         "adapterType":1,
         "manufacturer": ""
      },
      {
         "id":null,
         "macAddress": "94:DE:80:52:4F:68",
         "adapterType":1,
         "manufacturer": "Broadcom"
      }
   ]
}
```

2.2.15 Get SEL Information

This API gets SEL information of a specified component.
You can perform this API for the component that is registered with BMC Management.

URL:

```
GET /esmpro/api/components/server/sel/{guid}
```

Request:

Nothing

ResponseBody:

Key	Value
url	The URL to get the job status. JSON Type : String /esmpro/api/job/status/*{jobid}*

Example:

Request

```
GET /esmpro/api/esmpro/api/components/server/sel/5f3cc680-cf1b-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "url" : "/esmpro/api/job/status/gas00001"
}
```

This API is one of the asynchronous REST API. So check the job status by sending HTTP GET request to the URL in response body. Refer to 2.1.1 for more detail.

When the job status becomes "completed", you can get the job result by sending HTTP GET to the following URL.

URL:

```
GET /esmpro/api/components/server/sel/{jobid}/result
```

Request:

Nothing

ResponseBody:

Key	Value
freeSpace	Free Space in bytes. JSON Type : Integer
sel	Array of SEL Record JSON Type : Array This API returns all entry of SEL.
recordId	Record ID JSON Type : String
severity	Severity JSON Type : String
date	Date JSON Type : String Format: JP) 2015/04/08 12:06:47 (YYYY/MM/DD HH:MM:SS) EN) 2/18/2015 17:05:22 (MM/DD/YYYY HH:MM:SS)
detail	Detail JSON Type : String
dump	SEL Dump Data in hexdecimal JSON Type : String

Example:

Request

```
GET /esmpro/api/components/server/sel/gas00001/result
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "freeSpace":63180,
   sel:[
      {
         "recordId":"0002h",
         "severity":"information",
         "date":2014/09/12 14:08:27",
         "detail":"OS Boot Information : C: boot completed",
         "dump":"02 00 02 5b fe 12 54 20 00 04 12 87 6f 41 8f ff"
      },
      {
         "recordId":"0001h",
         "severity":"information",
         "date":2014/09/12 14:08:23",
         "detail":" System Boot/Restart Initiated Information : Initiated by power up",
         "dump":"3e 00 02 6e 38 de 54 20 00 04 12 87 6f 41 8f ff"
      }
   ]
}
```

2.2.16 Get FRU List

This API gets FRU List of a specified component.

URL:

GET /esmpro/api/components/server/fru-list/*{guid}*

Request:

Nothing

ResponseBody:

Key		Value
frulist		FRU List JSON Type : Array
	frulist[n].fruId	FRU Device ID JSON Type : Integer Numerical value of 0 to 255
	frulist[n].description	FRU Name JSON Type : String

Example:

Request
GET /esmpro/api/components/server/fru-list/5f3cc680-cf1b-11e0-8001-00255cc64b2e Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083 X-ESMPRO-API-Version:1.0

Response

HTTP 1.1 200 OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8
 	
{
{
frulist:[
{ "fruId":0, "description":"Primary FRU Device" },
{ "fruId":1, "description":"DIMM1 SPD" },
{ "fruId":2, "description":"DIMM2 SPD" },
{ "fruId":3, "description":"DIMM3 SPD" },
{ "fruId":4, "description":"DIMM4 SPD" }
]
}
}

2.2.17 Get FRU Record

This API gets FRU Record Request that is specified by fruid. Specify the "FRU Device ID" in fruid. Refer to 2.2.16 Get FRU List about the "FRU Device ID".

URL:

GET /esmpro/api/components/server/fru/*{guid}?fruid={fruid}*

Request:

Nothing

ResponseBody:

FRU Type: Generic

Key	Value
description	FRU Name JSON Type : String
chassisInformation	Chassis Information If "Chassis Information" is not present, "" (empty string) is returned in each field.
chassisType	Chassis Type JSON Type : String
chassisSerialNumber	The serial number of Chassis JSON Type : String
chassisCharacteristics	The characteristics of Chassis JSON Type : String
boardInformation	Board Information If "Board Information" is not present, "" (empty string) is returned in each field.
mfgDateTime	Manufacturing Date Time JSON Type : string Format : JP) 2015/04/08 12:06:47 (YYYY/MM/DD HH:MM:SS) EN) 2/18/2015 17:05:22 (MM/DD/YYYY HH:MM:SS)
boardManufacturer	Board Manufacturer JSON Type : String
boardProductName	Board Product Name JSON Type : String
boardSerialNumber	Board Serial Number JSON Type : String
boardPartNumber	Board Part Number JSON Type : String
boardVersion	Board Version JSON Type : String
productInformation	Product Information If "Product Information" is not present, "" (empty string) is returned in each field.
manufacturerName	Manufacturer Name JSON Type : String
productName	Product Name JSON Type : String
productPartModelNumber	Product Part Model Number JSON Type : String
productVersion	FR Version JSON Type : String
productSerialNaumber	Product Serial Number JSON Type : String

assertTag	Assert Tag JSON Type : String
dump	Hexadecimal dump data of FRU Record JSON Type : String

FRU Type: Memory

Key	Value
description	FRU Name JSON Type : String
memoryType	Memory Type JSON Type : String
dump	Hexadecimal dump data of FRU Record JSON Type : String

FRU Type: CPU

Key	Value
description	FRU Name JSON Type : String
dump	Hexadecimal dump data of FRU Record JSON Type : String

Example:

Request

```
GET /esmpro/api/components/server/fru/5f3cc680-cf1b-11e0-8001-00255cc64b2e?fruid=0
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

FRU Type: Generic

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   {
      "boardInfomation": {
         "boardManufacturer": "GIGABYTE",
         "boardProducatName": "GA-6UASV1",
         "boardSerialNumber": "0BM550700148",
         "boardVersion": "1.1H",
         "baordPartNumber": "56-131335-001"
      },
         "productInfomation": {
         "productVersion": "FR1.3",
         "assertTag": "_______________",
         "productSerialNumber": "1700121",
         "manufacturerName": "NEC",
         "proudctName": "Express5800/R110d-1E",
         "proudctPartModelNumber": "[N8100-1764]"
      },
      "description": "Primary FRU Device",
      "chassisInformation": {
         "chassisType": "Rack Mount Chassis",
         "chassisPartNumber": "856-131336-001 ",
         "chassisSerialNumber": "Rack Mount",
         "chassisCharacteristics": "01h"
      },
```

```
        "dump": "01 19 01 05 0e 00 00 d2 01 04 17 ce 38 35 36 2d 31 33 31 33 33 36 2d 30 30 31 c2 30
31 c2 30 31 "
    }
}
```


FRU Type: Memory

Response


```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
    {
        "description": "DIMM1 SPD",
        "memoryType": "DDR3 SDRAM",
        "dump": "92 10 0b 02 03 19 00 09 0b 52 01 08 0c 00 3c 00 "
    }
}
```


FRU Type: CPU

Response


```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
    {
        "description": "CPU 1 Info",
        "dump": "92 10 0b 02 03 19 00 09 0b 52 01 08 0c 00 3c 00"
    }
}
```

2.2.18 Check Connection

This API confirms connection with the management target on a specified component.

URL:

PUT /esmpro/api/components/server/connection-check

This API fails when you specify both componentName and componentGUID or you do not specify both.

Request:

Key	Value
componentName	Name of the target component. JSON Type : String
componentGUID	GUID of the target component JSON Type : String

Response Body:

Key	Value
url	The URL to get the job status. JSON Type : String /esmpro/api/job/status/*{jobid}*

Example:

Request (Example1)

```
PUT/esmpro/api/components/server/connection-check
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "componentName": "ServerA"
}
```

Request (Example2)

```
PUT /esmpro/api/components/server/connection-check
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "componentGUID": "5f3cc680-cf1b-11e0-8001-00255cc64b2e"
}
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "url" : "/esmpro/api/job/status/csc00001"
}
```

This API is one of the asynchronous REST API. So check the job status by sending HTTP GET request to the URL in response body. Refer to 2.1.1 for more detail.

When the job status becomes "completed", you can get the job result by sending HTTP GET to the following URL.

URL:

GET /esmpro/api/components/server/connection-check/*{jobid}*/result

Request:

Nothing

ResponseBody:

Key	Value
snmpManagement	
detected	The result of detection of SNMP management. JDON Type : String
detail	A detail message of result of connection check execution. JSON type : String
wsmanManagement	
detected	The result of detection of WSMAN management. JSON Type : String
detail	A detail message of result of connection check execution. JSON type : String
esxiManagement	
detected	The result of detection of ESXi management. JSON Type : String
detail	A detail message of result of connection check execution. JSON type : String
raidManagement	
detected	The result of detection RAID System management. JSON Type : String
detail	A detail message of result of connection check execution. JSON type : String
expressUpdateManagement	
detected	The results of detection EU Management. JSON Type : String
detail	A detail message of result of connection check execution. JSON type : String
bmcManagement	
detected	The results of detection BMC Management. JSON Type : String
detail	A detail message of result of connection check execution. JSON Type : String
vproManagement	
detected	The result of detection vPro Management. JSON Type : String
detail	A detail message of result of connection check execution. JSON Type : String
iloManagement	
detected	The result of detection iLO Management. JSON Type : String
detail	A detail message of result of connection check execution. JSON Type : String
otherBmcManagement	

	detected	The result of detection BMC (Other) Management. JSON Type : String
	detail	A detail message of result of connection check execution. JSON Type : String

Example:

Request

```
GET /esmpro/api/components/server/connection-check/csc00001/result
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "wsmanManagement": {
      "detected": "true",
      "detail": "WS-MAN management can be used."
   }
   "raidManagement": {
      "detected": "true",
      "detail": "RAID system management can be used."
   }
   "expressUpdateManagement": {
      "detected": "true",
      "detail": "Using the NEC ExpressUpdate Agent NEC ExpressUpdate function can be used."
   }
   "bmcManagement": {
      "detected": "true",
      "detail":"SNMP Alert configuration is not completed on BMC (EXPRESSSCOPE Engine). Setting up SNMP Alert configuration is required to send hardware error alert to NEC ESMPRO Manager. Please set NEC ESMPRO Manager's address as 'Alert Receiver'. Using the BMC (EXPRESSSCOPE Engine) NEC ExpressUpdate function can be used. "
   }
}
```

2.2.19 Get Event Information

This API gets the information about the Event information which matches specified severity and has record ID after the specified record ID. Those events has been received by NEC ESMPRO Manager. If you omit to specify the record ID, this API returns the event which has MAX Record ID.

You can perform this API request without session ID in HTTP Header.

URL:

GET /esmpro/api/event?recordId=*{ID}*&severity=*{All or Information or Minor or Major}*

Request:

Nothing

Response Body:

Key		Value
events		Array of event information in ascending order of record ID. JSON type : Array
	recordId	Record ID JSON Type : Integer
	componentName	component name JSON Type : String
	ipAddress	IP Address of the component JSON Type : String
	summary	Summary of alert JSON Type : String
	severity	Severity of alert (Unknown, Information, Minor, Major) JSON type : String
	detail	Detail of alert JSON Type : String
	recovery	Recovery action of alert JSON Type : String
	productName	The product name that event has been generated. JSON Type : String
	source	The service name that event have been generated. JSON Type : String
	eventID	Event ID JSON Type : String
	alertType	Type of Alert JSON Type : String
	receiveTime	The time that NEC ESMPRO Manager received the event. JSON Type : String(yyyy/mm/dd hh:mm:ss)
	occurTime	The time when event has been generated. JSON Type : String(yyyy/mm/dd hh:mm:ss)
	guid	GUID of the component. JSON Type : String
	identifier	Identifier of event. (Not supported.) JSON Type : String

Example:

Request

```
GET /esmpro/api/event?recordId=1&severity=Information
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "events": [
      {
         "recordId" : 1,
         "componentName" : "SERVER1",
         "ipAddress" : "172.16.0.61",
         "summary" : "HW eventlog",
         "severity" : "Information",
         "detail" : "HW eventlog",
         "recovery": "Check it.",
         "productName" : "ESMPRO/SM",
         "source" : "ESMCommonService",
         "eventID": "0xc004057a(1402)",
         "alertType" : "Server Recovery",
         "receiveTime" : "2014/09/18 11:30:38",
         "occurTime" : "2014/09/18 11:30:38",
         "guid": "5f3cc680-cf1b-11e0-8001-00255cc64b2e"
      }
   ]
}
```

2.2.20 Get NEC ESMPRO Manager Information

This API gets the information of NEC ESMPRO Manager.

URL:

```
GET /esmpro/api/sm
```

Request:

Nothing

Response Body:

Key	Value
smName	Name of NEC ESMPRO Manager. JSON Type : String
smVersion	Version number of NEC ESMPRO Manager. JSON Type : String

Example:

Request

```
GET /esmpro/api/sm
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   " smName " : "mgr_PC00001" ,
   " smVersion " : "6.20"
}
```

2.2.21 Get Component Status

This API gets the status of a specified component. There are two ways to specify the component: One is to specify by GUID or the other is to specify by component name.

URL:

GET /esmpro/api/components/server/server-status/*{guid}*

Request:

Nothing

Response Body:

Key	Value
status	This shows the status of a specified component. JSON Type : Integer 0:NO_MONITORING 1:NORMAL 2:UNKNOWN 3:DC-OFF, POST, OS Panic 4:WARNING 5:ERROR

Example:

Request

```
GET /esmpro/api/components/server/server-status/5f3cc680-cf1b-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "status":1
}
```

URL:

GET /esmpro/api/components/server/server-status?name=*{Component Name}*

Request:

Key	Value
name	Component name registered on NEC ESMPRO Manager

ResponseBody:

Key	Value
status	Component Status JSON Type : Integer 0 : NO_MONITORING 1 : NORMAL 2 : UNKNOWN 3 : DC-OFF, POST, OS Panic 4 : WARNING 5 : ERROR

Example:

Request
GET /esmpro/api/components/server/server-status?name=ManagementController Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083 X-ESMPRO-API-Version:1.0

Response

HTTP 1.1 200 OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8
 	
{
"status":1
}

2.2.22 Get Storage Information

This API gets the Storage Information of a specified component.
N8103-239 480GB OS boot-only SSD board and N8103-240 RAID controller and It is also possible to obtain information on the storage of the iLO6-equipped device.
In the case of Express5800/R110k-1, information on the same controller may be acquired more than once.

URL:

GET /esmpro/api/components/server/storage/*{guid}*

Request:

Nothing

Response Body:

Key	Value
url	The URL to get the job status. JSON Type : String /esmpro/api/job/status/*{jobid}*

Example:

Request

```
GET /esmpro/api/components/server/storage/d3eceb00-91ba-11e2-8001-902b34341528
Cookie: JSESSIONID=342886D32A4295936188A6A8410C1E94
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "url" : "/esmpro/api/job/status/gsi00001"
}
```

This API is one of the asynchronous REST API. So check the job status by sending HTTP GET request to the URL in response body. Refer to 2.1.1 for more detail.

When the job status becomes "completed", you can get the job result by sending HTTP GET to the following URL.

URL:

```
GET /esmpro/api/components/server/storage/{jobid}/result
```

Request:

Nothing

ResponseBody:

Key	Value
controllers	Array of Controller information JSON Type : Array
controllerId	Controller ID JSON Type : String
model	Model name of Controller JSON Type : String
interfaceType	Interface type of Controller JSON Type : String（"IDE"、"SCSI"、"Other"）
driverName	Driver name JSON Type : Array
deviceMap	Device ID connected a specified controller JSON Type : Array
devices	Array of Device information JSON Type : Array
index	Device ID JSON Type : String
deviceType	Device Type JSON Type : String（"Hard Disk"）
capacity	Total capacity of Device JSON Type : String
model	Model name JSON Type : String
firmwareRevision	Firmware Revision JSON Type : String
serialNumber	Serial Number JSON Type : String
status	Status of Device JSON Type : String（"normal"、"warning"、"unknown"）
controllerId	Id of connected Controller JSON Type : String
driverName	Driver Name JSON Type : String
iLO	Array of Storage information from iLO JSON Type : Array
controllers	Array of controller information JSON Type : Array
controllerId	Controller ID JSON Type : String
model	Model JSON Type : String

health	Health JSON Type : String
state	State JSON Type : String
serialNumber	Serial Number JSON Type : String
hardwareRevision	Hardware Version JSON Type : String
location	Location JSON Type : String
firmwareVersion	Firmware Version JSON Type : String
adapterType	Adapter Type JSON Type : String
cacheModuleSize	Cache Module Size JSON Type : String
cacheRatio	Cache Ratio JSON Type : String
enableWriteCacheWhenBatteryNotPresentOrNotCompletelyCharged	Enable Write Cache when Battery not Present or not Completely Charged JSON Type : String
writeCacheBypassThreshold	Threshold of Write Cache Bypassed JSON Type : String
logicalDrives	Array of logical Drives Information JSON Type : Array
id	Id of the logical drive JSON Type : String
capacity	Capacity JSON Type : String
type	type JSON Type : String
name	Name JSON Type : String
raidLevel	Raid Level JSON Type : String
health	Health JSON Type : String
state	State JSON Type : String
physicalDevice	Composed Physical Device JSON Type : Array
diskDrives	Array of Physical Drives Information JSON Type : Array
id	Id of the Physical drive JSON Type : String
firmwareVersion	Firmware Version JSON Type : String
capacity	Capacity JSON Type : String
currentTemperature	Temperature JSON Type : String
interfaceType	Interface JSON Type : String
model	Model

				JSON Type : String
			serialNumber	Serial Number JSON Type : String
			location	Location JSON Type : String
			mediaType	Media Type JSON Type : String
			utilization	Utilization JSON Type : String
			health	Health JSON Type : String
			state	State JSON Type : String
		storageEnclosures		Array of Storage Enclosure information JSON Type : Array
			id	Storage Enclosure Id JSON Type : String
			firmwareVersion	Version of firmware JSON Type : String
			serialNumber	Serial Number JSON Type : String
			driveBay	Drive Bay JSON Type : String
			location	Location JSON Type : String
			health	Health JSON Type : String
			state	State JSON Type : String
	batteries			Array of batteries information JSON Type : Array
		id		Battery Id JSON Type : String
		chargeLevelPercent		Charge Level Percentage JSON Type : String
		firmwareVersion		Firmware Version JSON Type : String
		maxPowerCapacity		Maximum power capacity JSON Type : String
		model		Model JSON Type : String
		productName		Product Name JSON Type : String
		serialNumber		Serial Number JSON Type : String
		sparePartNumber		Spare Part Number JSON Type : String
		health		Health JSON Type : String
		state		State JSON Type : String

Example:

Request

```
GET /esmpro/api/components/server/storage/gsi00001/result
Cookie: JSESSIONID=342886D32A4295936188A6A8410C1E94
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "controllers": [
      {
         "interfaceType": "IDE",
         "controllerId": "1",
         "deviceMap": [
            "1"
         ],
         "model": "IDE Controller(storahci)",
         "driverName": "storahci.sys"
      }
   ],
   "devices": [
      {
         "deviceType": "HardDisk",
         "serialNumber": "WD-WMAYP3264344",
         "controllerId": "1",
         "index": "1",
         "model": "WDC          WD5003ABYX-20WER",
         "driverName": "",
         "capacity": "465.7 GB",
         "firmwareRevision": "01.0",
         "status": "normal"
      }
   ]
}
```

2.2.23 Get Storage Information（N8103-239, N8103-240, iLO6 device）

Retrieves controller information for N8103-239 480GB OS boot-only SSD board/N8103-240 RAID controller and iLO6 unit storage.

URL:

```
GET /esmpro/api/components/server/ilo-storage/{guid}
```

Request:

Nothing

Response Body:

Key	Value
url	The URL to get the job status. JSON Type : String /esmpro/api/job/public-rest-status/*{jobid}*

Example:

Request

```
GET /esmpro/api/components/server/ilo-storage/d3eceb00-91ba-11e2-8001-902b34341528
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "url" : "/esmpro/api/job/public-rest-status/gsi00001"
}
```

This API is one of the asynchronous REST API. So check the job status by sending HTTP GET request to the URL in response body. Refer to 2.1.1 for more detail. Note that the cookie header line described in 2.1.1 is not required.

When the job status becomes "completed", you can get the job result by sending HTTP GET to the following URL.

URL:

```
GET /esmpro/api/components/server/ilo-storage/{jobid}/result
```

Request:

Nothing

ResponseBody:

Key	Value
iLO	Array of Storage information from iLO JSON Type : Array
controllers	Array of controller information JSON Type : Array
controllerId	Controller ID JSON Type : String
model	Model JSON Type : String

health	Health JSON Type : String
state	State JSON Type : String
serialNumber	Serial Number JSON Type : String
hardwareRevision	Hardware Version JSON Type : String
location	Location JSON Type : String
firmwareVersion	Firmware Version JSON Type : String
adapterType	Adapter Type JSON Type : String
cacheModuleSize	Cache Module Size JSON Type : String
cacheRatio	Cache Ratio JSON Type : String
enableWriteCacheWhenBatteryNotPresentOrNotCompletelyCharged	Enable Write Cache when Battery not Present or not Completely Charged JSON Type : String
writeCacheBypassThreshold	Threshold of Write Cache Bypassed JSON Type : String
logicalDrives	Array of logical Drives Information JSON Type : Array
id	Id of the logical drive JSON Type : String
capacity	Capacity JSON Type : String
type	type JSON Type : String
name	Name JSON Type : String
raidLevel	Raid Level JSON Type : String
health	Health JSON Type : String
state	State JSON Type : String
diskDrives	Array of Physical Drives Information JSON Type : Array
id	Id of the Physical drive JSON Type : String
firmwareVersion	Firmware Version JSON Type : String
capacity	Capacity JSON Type : String
currentTemperature	Temperature JSON Type : String
interfaceType	Interface JSON Type : String
model	Model JSON Type : String
serialNumber	Serial Number

				JSON Type : String
			location	Location JSON Type : String
			mediaType	Media Type JSON Type : String
			utilization	Utilization JSON Type : String
			health	Health JSON Type : String
			state	State JSON Type : String

Example:

Request

```
GET /esmpro/api/components/server/ilo-storage/gsi00001/result
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8
{
   "iLO": [
         "controllers": [
            {
               "controllerId": "DE00xxxx",
               "model": "HPE NS204i-p Gen10+ Boot Controller",
               "health": "OK",
               "state": "Enabled",
               "serialNumber": "xxxxxxxxxx1234",
               "hardwareRevision": "",
               "location": "Slot 1",
               "firmwareVersion": "1.0.14.1055",
               "adapterType": "",
               "cacheModuleSize": "0 MB",
               "cacheRatio": "",
               "enableWriteCacheWhenBatteryNotPresentOrNotCompletelyCharged": "",
               "writeCacheBypassThreshold": "",
               "logicalDrives": [
                  {
                     "id": "1",
                     "capacity": "447.0 GB",
                     "type": "RAID1",
                     "name": "NS Volume",
                     "raidLevel": "",
                     "health": "OK",
                     "state": "Enabled"
                  }
               ],
               "diskDrives": [
                  {
                     "id": "1",
                     "firmwareVersion": "",
                     "capacity": "447.1 GB",
                     "currentTemperature": "",
                     "interfaceType": "",
                     "model": "xxxxxxxxxx123",
```

```
                    "serialNumber": "xxxxxxxxx1234567",
                    "location": "Slot=1:Bay=1",
                    "mediaType": "SSD",
                    "utilization": "",
                    "health": "OK",
                    "state": "Enabled" }
                ]
            }
        ]
    ]
}
```

2.2.24 Get ExpEther Manager List

This API gets an information list of ExpEther Manager.

URL:

```
GET /esmpro/api/eem
```

Request:

Nothing

ResponseBody:

Key		Body
eems		Array of management information every EEM JSON Type : Array
	status	state of ExpEther Manager JSON Type : String unknown error warning normal
	monitoring	monitoring status JSON Type : String enabled disabled
	name	Registered ExpEther Manager name JSON Type : String
	interval	monitoring interval(second) JSON Type : Integer
	ipAddress	IP address for ExpEther Manager access JSON Type : String Enter the number of 4 sets of 0-255 divided by "."
	protocol	HTTP or HTTPS JSON Type : String
	port	Port number for ExpEther Manager access JSON Type : String
	user	User name of Basic authentication of ExpEther Manager access JSON Type : String

Example:

Request

```
GET /esmpro/api/eem
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "eems": [
   {
         "status": "normal",
         "monitoring": "enabled",
         "name": "ExpressEtherManager001",
         "interval":1800,
         "ipAddress":"1.2.3.4",
         "protocol": "http"
         "port":"30050"
         "user":"admin"
      },
      {
         "status": "normal",
         "monitoring": "enabled",
         "name": "ExpressEtherManager002",
         "interval":1800,
         "ipAddress":"5.6.7.8",
         "protocol": "http"
         "port":"30050"
         "user":"eem"
      }
   ]
}
```

2.2.25 Discover ExpEther Manager

This API discovers and registers ExpEther Manager as a management target in NEC ESMPRO manager.
This API is the asynchronous method. The result of the API can be got using URL of response data.

URL:

POST /esmpro/api/eem

Request:

Key	Value
discoveryMode	IP address range search or a network address search. [Mandatory] JSON Type : Integer 0 : IP Address Range Search 1 : Network Address Search The value of other than the above will be an error.
startAddress	Start IP address (If discoveryMode is Network Address Search, this key is ignored.) JSON Type : String Enter the number of 4 sets of 0-255 divided by "."
endAddress	End IP address (If discoveryMode is Network Address Search, this key is ignored.) JSON Type : String Enter the number of 4 sets of 0-255 divided by "."
networkAddress	Network address (If discoveryMode is IP Address Range Search, this key is ignored.) JSON Type : String Enter the number of 4 sets of 0-255 divided by "."
networkMask	Network mask (If discoveryMode is IP Address Range Search, this key is ignored.) JSON Type : String Enter the number of 4 sets of 0-255 divided by "."
port	Port number (Optional and default value is 30500) JSON Type : Integer
accounts	Array of account information used by the communication with ExpEther Manager. Enter the account on Basic authentication of ExpEther Manager. It is possible to enter the account at most 5 cases.
user	User name of Basic authentication of ExpEther Manager access (*) [Mandatory] JSON Type : String (1-255 character)
password	Password of Basic authentication of ExpEther Manager access (*) [Mandatory] JSON Type : String (1-255 character)

* For the Basic authentication setting of ExpEther Manager, see the user's guide of ExpEther IO Expansion Unit (40G).

ResponseBody:

Key	Value
url	The URL to get the job status. JSON Type : String /esmpro/api/job/status/*{jobid}*

Example:

Request

```
POST /esmpro/api/eem
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "discoveryMode" : 0,
   "startAddress" : "192.168.1.1",
   "endAddress" : "192.168.1.20",
   "accounts" : [
      {
         "user" : "Administrator",
         "password" : "Administrator"
      }
   ],
}
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "url" : "/esmpro/api/job/status/eem00001"
}
```

This API is one of the asynchronous REST API. So check the job status by sending HTTP GET request to the URL in response body. Refer to 2.1.1 for more detail.

When the job status becomes "completed", you can get the job result by sending HTTP GET to the following URL.

URL:

```
GET /esmpro/api/eem/{jobid}/result
```

Request:

Nothing

ResponseBody:

Key		Value
result		Array with data of a registration result JSON Type : Object
	name	Registered ExpEther Manager name JSON Type : String
	ipAddress	IP address for ExpEther Manager access JSON Type : String Enter the number of 4 sets of 0-255 divided by "."
	status	state of ExpEther Manager JSON Type : String unknown error warning normal

Example:

Request

```
GET /esmpro/api/eem/eem00001/result
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "result" : [
      {
         "name": "ExpressEtherManager001",
         "ipAddress":"192.168.1.4",
         "status":"normal"
      }
   ]
}
```

2.2.26 Delete ExpEther Manager

This API deletes ExpEther Manager from a management target in NEC ESMPRO Manager.

URL:

DELETE /esmpro/api/eem?*name={Registration name}*

Request:

Key	Value
name	Registered ExpEther Manager name

ResponseBody:

Nothing

Example:

Request
DELETE /esmpro/api/eem?name=ExpressEtherManager001 Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083 X-ESMPRO-API-Version:1.0

Response
HTTP 1.1 200 OK X-ESMPRO-API-Version:1.0 ,

2.2.27 Get IML Information

This API gets IML information of a specified component.
You can perform this API for the component that is registered with iLO Management.

URL:

GET /esmpro/api/components/server/iml/*{guid}*

Request:

Nothing

ResponseBody:

Key	Value
url	The URL to get the job status. JSON Type : String /esmpro/api/job/status/*{jobid}*

Example:

Request

```
GET /esmpro/api/esmpro/api/components/server/iml/5f3cc680-cf1b-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   "url" : " /esmpro/api/job/status/iml00001"
}
```

This API is one of the asynchronous REST API. So check the job status by sending HTTP GET request to the URL in response body. Refer to 2.1.1 for more detail.

When the job status becomes "completed", you can get the job result by sending HTTP GET to the following URL.

URL:

GET /esmpro/api/components/server/iml/*{jobid}*/result

Request:

Nothing

Response Body:

Key	Description
iml	Array of IML JSON Type : Array
id	Id of IML JSON Type : String
severity	Severity of IML JSON Type : String
eventClass	Event Class Code (in decimal). Identifies the type of event that occurred. JSON Type : Integer
className	Event Class Name. The name of class that identifies the type of event that occurred, for example, network, maintenance, and so on. JSON Type : String
eventCode	Event code(in decimal) JSON Type : Integer
repaired	"Repaired" shows that an event has undergone corrective action. JSON Type : Boolean true : repaired false : not repaired
lastUpdate	The date and time when the latest event of this type occurred. JSON Type : String Format : 2015/02/18 17:05:22 (YYYY/MM/DD HH:MM:SS)
initialUpdate	The date and time when the first event of this type occurred. JSON Type : String Format : 2015/02/18 17:05:22 (YYYY/MM/DD HH:MM:SS)
count	The number of times this event has occurred. JSON Type : Integer
description	Description of event. JSON Type : String
recommendedAction	Recommended recovery action. JSON Type : String

Example:

Request

```
GET /esmpro/api/components/server/iml/iml00001/result
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083
X-ESMPRO-API-Version:1.0
```

Response

```
HTTP 1.1    200    OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8

{
   iml:[
      {
         "id" : "4",
         "severity" : "OK",
         "eventClass" : 33,
         "className" : "Maintenance",
         "eventCode" : 2,
         "repaired" : false,
         "lastUpdate" : "10/11/2016 14:08:27",
         "initialUpdate" : "10/11/2016 14:08:27",
         "count" : 1,
         "description" : "Maintenance note: Trap test",
         "recommendedAction" : null
      },
      {
         "id" : "5",
         "severity" : "Critical",
         "eventClass" : 10,
         "className" : "POST Message",
         "eventCode" : 1,
         "repaired" : false,
         "lastUpdate" : "10/12/2016 14:08:59",
         "initialUpdate" : "10/12/2016 14:08:59",
         "count" : 1,
         "description" : "POST Error"
         "recommendedAction" : null
      }
   ]
}
```

2.2.28 Get Group Status

This API gets the status of a specified component group.

URL:

GET /esmpro/api/groups/status?name=*{group name}*

Request:

Key	Value
name	Group Name JSON Type : String If this key is omitted, "root" is specified.

Response Body:

Key	Value
url	The URL to get the job status. JSON Type : String /esmpro/api/job/status/*{jobid}*

Example:

Request
GET /esmpro/api/groups/status?name=root Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083 X-ESMPRO-API-Version:1.0

Response

HTTP 1.1 200 OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8
 	
{
"url" : "/esmpro/api/job/status/gst00001"
}

This API is one of the asynchronous REST API. So check the job status by sending HTTP GET request to the URL in response body. Refer to 2.1.1 for more detail.

When the job status becomes "completed", you can get the job result by sending HTTP GET to the following URL.

URL:

GET /esmpro/api/groups/status/*{jobid}*/result

Request:

Nothing

Response Body:

Key	Value
status	This value shows the group status. JSON Type : Integer 0 : NO_MONITORING 1 : NORMAL 2 : UNKNOWN 3 : DC-OFF, POST, OS Panic 4 : WARNING 5 : ERROR If there is no component which has enable management in a specified group, this API returns "0" (NO_MONITORING).

Example:

Request
GET /esmpro/api/groups/status/gst00001/result Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083 X-ESMPRO-API-Version:1.0

Response

HTTP 1.1 200 OK
X-ESMPRO-API-Version:1.0
Content-type : application/json; charset=utf-8
 	
{
"status":1
}