Chapter 1  About RESTful API

Chapter 2  API Reference
Contents

Terminology ......................................................................................................................................................................... 1
Trademarks ............................................................................................................................................................................. 2
About This Document .......................................................................................................................................................... 3
Chapter 1 ............................................................................................................................................................................ 4
  1.1 Overview .................................................................................................................................................................. 5
  1.2 How to use the RESTful API ........................................................................................................................................ 5
    1.2.1 Request Format .................................................................................................................................................. 5
    1.2.2 Response Format .................................................................................................................................................. 8
    1.2.3 Asynchronous REST API .................................................................................................................................... 11
Chapter 2 API Reference ...................................................................................................................................................... 13
  2.1 Job Management REST API ....................................................................................................................................... 13
    2.1.1 Get Job status ...................................................................................................................................................... 13
    2.1.2 Get Job result ....................................................................................................................................................... 14
    2.1.3 Cancel Job ............................................................................................................................................................ 14
  2.2 REST API .................................................................................................................................................................. 15
    2.2.1 Login ................................................................................................................................................................. 15
    2.2.2 Logout ................................................................................................................................................................. 16
    2.2.3 Auto Registration ............................................................................................................................................... 17
    2.2.4 Get Component List ........................................................................................................................................... 23
    2.2.5 Get Connection Setting ........................................................................................................................................ 25
    2.2.6 Set Connection Setting ....................................................................................................................................... 28
    2.2.7 Delete Component ............................................................................................................................................... 32
    2.2.8 Get Power State .................................................................................................................................................. 33
    2.2.9 Power Control ...................................................................................................................................................... 34
    2.2.10 Get All Sensors Information ............................................................................................................................. 35
    2.2.11 Get Sensor Information .................................................................................................................................... 38
    2.2.12 Get System Information ................................................................................................................................... 40
    2.2.13 Get Hardware Information ............................................................................................................................... 41
    2.2.14 Get Network Information ................................................................................................................................ 44
    2.2.15 Get SEL Information .......................................................................................................................................... 46
    2.2.16 Get FRU List ...................................................................................................................................................... 48
    2.2.17 Get FRU Record .................................................................................................................................................. 49
    2.2.18 Check Connection .............................................................................................................................................. 52
    2.2.19 Get Event Information ...................................................................................................................................... 55
    2.2.20 Get NEC ESMPRO Manager Information ......................................................................................................... 57
    2.2.21 Get Component Status ..................................................................................................................................... 58
    2.2.22 Get ExpEther Manager List ............................................................................................................................... 60
    2.2.23 Discover ExpEther Manager ............................................................................................................................. 62
    2.2.24 Delete ExpEther Manager .................................................................................................................................. 65
    2.2.25 Get IML Information ........................................................................................................................................... 66
    2.2.26 Get Group Status ................................................................................................................................................. 69
## Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
</table>
| BMC          | Baseboard Management Controller  
The management controller to manage the interface between system management software and platform hardware. |
| iLO          | Integrated Lights-Out  
The management controller to manage the interface between system management software and platform hardware. |
| vPro         | Intel platform brand for business users (Intel® vPro™ Technology).                                                                           |
| Component    | An object that is managed by NEC ESMPRO Manager.                                                                                               |
| WS-Man       | Web Service Management  
Open standard protocol for remotely accessing management data with any computer device.                                                      |
| 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.                     |
| IML          | Integrated Management Log  
The IML provides a record of historical events that have occurred on the server.                                                            |
Trademarks

NEC EXPRESSBUILDER and NEC ESMPRO are registered trademarks of NEC Corporation. Microsoft, Windows, Windows Vista, 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
Chapter 1  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.

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

1.2.1.3 HTTP Header

REST API supports the following HTTP Header:

<table>
<thead>
<tr>
<th>Header Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cookie</td>
<td>In order to access REST API, client must add &quot;Cookie&quot; 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</td>
</tr>
<tr>
<td>X-ESMPRO-API-Version</td>
<td>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 &quot;1.0&quot;. e.g. X-ESMPRO-API-Version:1.0</td>
</tr>
<tr>
<td>Content-Type</td>
<td>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</td>
</tr>
</tbody>
</table>
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.

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

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Message</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>*There are multiple messages corresponding to error code 1000. Please refer to actual message.</td>
<td>This error code will be returned when illegal request parameters are specified.</td>
</tr>
<tr>
<td>1001</td>
<td>The specified job cannot be found.</td>
<td></td>
</tr>
<tr>
<td>1003</td>
<td>Permission denied</td>
<td></td>
</tr>
<tr>
<td>1005</td>
<td>Session ID is invalid</td>
<td></td>
</tr>
<tr>
<td>1006</td>
<td>Authentication Error</td>
<td></td>
</tr>
<tr>
<td>1007</td>
<td>Management settings for obtaining the information is invalid.</td>
<td></td>
</tr>
<tr>
<td>1008</td>
<td>The specified job is not completed successfully.</td>
<td></td>
</tr>
<tr>
<td>1014</td>
<td>The specified component cannot be found.</td>
<td></td>
</tr>
<tr>
<td>1015</td>
<td>The specified ID is invalid.</td>
<td></td>
</tr>
<tr>
<td>1017</td>
<td>Directory Service connect Error.</td>
<td></td>
</tr>
<tr>
<td>1018</td>
<td>Sensor Identifier is not found.</td>
<td></td>
</tr>
<tr>
<td>1020</td>
<td>The specified groupName is not found.</td>
<td></td>
</tr>
<tr>
<td>1021</td>
<td>The range of IP address is too large.</td>
<td></td>
</tr>
<tr>
<td>1022</td>
<td>The other process of component registration is running.</td>
<td></td>
</tr>
<tr>
<td>1025</td>
<td>Session have already been authenticated.</td>
<td></td>
</tr>
<tr>
<td>1026</td>
<td>The set API version is not compatible with the current API version.</td>
<td></td>
</tr>
<tr>
<td>1027</td>
<td>HTTP header is invalid.</td>
<td></td>
</tr>
<tr>
<td>1029</td>
<td>The method is not allowed.</td>
<td></td>
</tr>
<tr>
<td>1032</td>
<td>An unrecognized parameter was found in the JSON data.</td>
<td></td>
</tr>
<tr>
<td>1033</td>
<td>Failed to parse the JSON data.</td>
<td></td>
</tr>
<tr>
<td>1034</td>
<td>Enter {0}</td>
<td></td>
</tr>
<tr>
<td>1035</td>
<td>{0} is illegal or invalid.</td>
<td></td>
</tr>
<tr>
<td>1036</td>
<td>{0} must be specified with alphanumeric characters.</td>
<td></td>
</tr>
<tr>
<td>1037</td>
<td>{0} must be {1} characters or more.</td>
<td></td>
</tr>
<tr>
<td>1038</td>
<td>{0} must be {1} characters or less.</td>
<td></td>
</tr>
<tr>
<td>1039</td>
<td>{0} must be in the range from {1} to {2}.</td>
<td></td>
</tr>
<tr>
<td>1040</td>
<td>{0} must be specified with numeric characters.</td>
<td></td>
</tr>
<tr>
<td>1041</td>
<td>The IP address is invalid.</td>
<td></td>
</tr>
<tr>
<td>1042</td>
<td>SNMP management status is not valid.</td>
<td></td>
</tr>
<tr>
<td>1043</td>
<td>OS management status is not valid.</td>
<td></td>
</tr>
<tr>
<td>1044</td>
<td>{0} must be in the range from {1} characters to {2} characters.</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>1045</td>
<td>This API isn't supporting it with the present OS.</td>
<td></td>
</tr>
<tr>
<td>1046</td>
<td>'{0}' isn't being supported with the present OS.</td>
<td></td>
</tr>
<tr>
<td>1047</td>
<td>Target machine does not support '{0}' feature</td>
<td></td>
</tr>
<tr>
<td>1048</td>
<td>Cannot read cache file</td>
<td></td>
</tr>
<tr>
<td>1049</td>
<td>Cannot execute the operation to the specified device.</td>
<td></td>
</tr>
<tr>
<td>1050</td>
<td>The specified device is not found.</td>
<td></td>
</tr>
<tr>
<td>1051</td>
<td>The component does not support alert test.</td>
<td></td>
</tr>
<tr>
<td>1052</td>
<td>Access is not permitted.</td>
<td></td>
</tr>
<tr>
<td>1053</td>
<td>'{0}' must be {1} or less.</td>
<td></td>
</tr>
<tr>
<td>1054</td>
<td>Management Controller type is mismatch.</td>
<td></td>
</tr>
<tr>
<td>1055</td>
<td>The user name or password of Management Controller is not entered.</td>
<td></td>
</tr>
<tr>
<td>1056</td>
<td>The user name or password of Management Controller is illegal length.</td>
<td></td>
</tr>
<tr>
<td>2012</td>
<td>Internal Application Error</td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Failed to read a JSON data from the request body.</td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Failed to generate a JSON data.</td>
<td></td>
</tr>
<tr>
<td>2015</td>
<td>Failed to write a JSON data to the response body.</td>
<td></td>
</tr>
<tr>
<td>3002</td>
<td>Internal Application Error</td>
<td></td>
</tr>
<tr>
<td>3003</td>
<td>Failed in execution of the job.</td>
<td></td>
</tr>
<tr>
<td>3004</td>
<td>Interrupt occurred to the job.</td>
<td></td>
</tr>
<tr>
<td>4003</td>
<td>Too many components exist.</td>
<td></td>
</tr>
<tr>
<td>4007</td>
<td>Failed to register the component.</td>
<td></td>
</tr>
<tr>
<td>4011</td>
<td>Failed to delete the component.</td>
<td></td>
</tr>
<tr>
<td>4012</td>
<td>Failed to get the sensor status.</td>
<td></td>
</tr>
<tr>
<td>4013</td>
<td>The auto registration process failed.</td>
<td></td>
</tr>
<tr>
<td>4014</td>
<td>Failed to get the event information.</td>
<td></td>
</tr>
<tr>
<td>4015</td>
<td>Failed to get the Information.</td>
<td></td>
</tr>
<tr>
<td>4016</td>
<td>Failed to release the component.</td>
<td></td>
</tr>
<tr>
<td>4017</td>
<td>Failed to set the connection settings.</td>
<td></td>
</tr>
<tr>
<td>4018</td>
<td>Failed to control the power status.</td>
<td></td>
</tr>
<tr>
<td>4019</td>
<td>Request failed because connection check was being executed.</td>
<td></td>
</tr>
<tr>
<td>4020</td>
<td>Request failed because installation process was being executed by NEC ExpressUpdate Agent.</td>
<td></td>
</tr>
<tr>
<td>4021</td>
<td>Failed to execute the RAID operation.</td>
<td></td>
</tr>
<tr>
<td>4022</td>
<td>Failed to execute ping process.</td>
<td></td>
</tr>
<tr>
<td>4023</td>
<td>Failed to get the result for ping process.</td>
<td></td>
</tr>
<tr>
<td>4024</td>
<td>Failed to get alert settings.</td>
<td></td>
</tr>
<tr>
<td>4025</td>
<td>Failed to set the alert settings.</td>
<td></td>
</tr>
<tr>
<td>4026</td>
<td>Failed to request the alert test.</td>
<td></td>
</tr>
<tr>
<td>4027</td>
<td>Failed to get the result for the alert test.</td>
<td></td>
</tr>
<tr>
<td>4028</td>
<td>Failed to get the resource URL.</td>
<td></td>
</tr>
<tr>
<td>9000</td>
<td>An application error occurred.</td>
<td></td>
</tr>
</tbody>
</table>


1.2.3 Asynchronous REST API

NEC ESMPRO Manager provides 2 kinds of REST API, the synchronous type and the asynchronous type. This chapter describes 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":"
   when the job is completed:
   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 example)
   - 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.
Chapter 2 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:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>REST API name</td>
</tr>
<tr>
<td>jobStatus</td>
<td>Job Status</td>
</tr>
<tr>
<td></td>
<td>&quot;Waiting&quot; : Waiting a job start</td>
</tr>
<tr>
<td></td>
<td>&quot;Running&quot; : Running</td>
</tr>
<tr>
<td></td>
<td>&quot;Cancel&quot; : Cancel</td>
</tr>
<tr>
<td></td>
<td>&quot;Completed&quot; : Normal end</td>
</tr>
<tr>
<td></td>
<td>&quot;Error&quot; : Abnormal end</td>
</tr>
<tr>
<td>errorCode</td>
<td>Error code</td>
</tr>
<tr>
<td>errorMessage</td>
<td>Error detail message</td>
</tr>
<tr>
<td>url</td>
<td>Asynchronous API is returned URL</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
</table>
| Result| true : Cancel succeeded  
false : Cancel failed |

**Example:**

**Request**

DELETE /esmpro/api/job/status/eem00125
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8A5C51B083
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 except Job Management REST API.

2.2.1 Login
Login for REST API authentication is performed.

URL:

```
POST /esmpro/api/login-session
```

Request:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>User name of NEC ESMPRO Manager account.</td>
</tr>
<tr>
<td>password</td>
<td>Password of NEC ESMPRO Manager account.</td>
</tr>
</tbody>
</table>

ResponseBody:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessionId</td>
<td>The ID to distinguish a session when accessing REST API.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE /esmpro/api/login-session</td>
</tr>
<tr>
<td>Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083</td>
</tr>
<tr>
<td>X-ESMPRO-API-Version:1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP 1.1 200 OK</td>
</tr>
<tr>
<td>X-ESMPRO-API-Version:1.0</td>
</tr>
</tbody>
</table>
## 2.2.3 Auto Registration

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

**URL:**

| POST /esmpro/api/components |

**Request:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>groupName</strong></td>
<td>Group name to register a component. (1-63 characters)</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>If this key is omitted, detected components are registered in &quot;root&quot; group.</td>
</tr>
<tr>
<td><strong>discoveryMode</strong></td>
<td>IP address range search or a network address search. [Mandatory]</td>
</tr>
<tr>
<td></td>
<td>JSON Type : Integer</td>
</tr>
<tr>
<td></td>
<td>0 : IP Address Range Search</td>
</tr>
<tr>
<td></td>
<td>1 : Network Address Search</td>
</tr>
<tr>
<td><strong>startAddress</strong></td>
<td>Start IP address (If discoveryMode is Network Address Search, this key is ignored.)</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>Enter the number of 4 sets of 0-255 divided by &quot;.&quot;</td>
</tr>
<tr>
<td><strong>endAddress</strong></td>
<td>End IP address (If discoveryMode is Network Address Search, this key is ignored.)</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>Enter the number of 4 sets of 0-255 divided by &quot;.&quot;</td>
</tr>
<tr>
<td><strong>networkAddress</strong></td>
<td>Network address (If discoveryMode is IP Address Range Search, this key is ignored.)</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>Enter the number of 4 sets of 0-255 divided by &quot;.&quot;</td>
</tr>
<tr>
<td><strong>networkMask</strong></td>
<td>Network mask (If discoveryMode is IP Address Range Search, this key is ignored.)</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>Enter the number of 4 sets of 0-255 divided by &quot;.&quot;</td>
</tr>
<tr>
<td><strong>snmpManagementValid</strong></td>
<td>Specify whether to register a component with management that uses NEC ESM PRO ServerAgent. [Mandatory]</td>
</tr>
<tr>
<td></td>
<td>JSON Type : Boolean</td>
</tr>
<tr>
<td></td>
<td>true if SNMP management is enabled, false otherwise</td>
</tr>
<tr>
<td><strong>snmpCommunity</strong></td>
<td>SNMP community name used to manage NEC ESM PRO ServerAgent. (Alphanumeric single-byte and sign (!#$%&amp;()*+-./<em>;&lt;&gt;?@[^</em>`{</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>If &quot;snmpManagementValid&quot; is true, this key is [Mandatory]</td>
</tr>
<tr>
<td><strong>wsManagementValid</strong></td>
<td>Specify whether to register a component with management that uses NEC ESM PRO ServerAgentService or VMWare ESXi5/6. [Mandatory]</td>
</tr>
<tr>
<td></td>
<td>JSON Type : Boolean</td>
</tr>
<tr>
<td></td>
<td>true if WS-Man management is enabled, false otherwise</td>
</tr>
<tr>
<td><strong>wsManAccount</strong></td>
<td>Array of user credential information of Windows or VMware ESXi5/6. It is possible to specify it up to five.</td>
</tr>
<tr>
<td></td>
<td>JSON Type : Object</td>
</tr>
<tr>
<td></td>
<td>If &quot;wsManagementValid&quot; is true, this key is [Mandatory]</td>
</tr>
<tr>
<td><strong>user</strong></td>
<td>User name of Windows or VMware ESXi5/6. (1-255 characters)</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>If &quot;wsManagementValid&quot; is true, this key is [Mandatory]</td>
</tr>
<tr>
<td><strong>password</strong></td>
<td>Password of Windows or VMware ESXi5/6. (1-255 characters)</td>
</tr>
<tr>
<td>JSON Type : String</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td></td>
</tr>
<tr>
<td>If &quot;wsManagementValid&quot; is true, this key is [Mandatory]</td>
<td></td>
</tr>
</tbody>
</table>

| 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 NEC 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] |

<table>
<thead>
<tr>
<th>otherBmcManagementValid</th>
<th>Specify whether to register a component with management that uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Value</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>lid</td>
<td>BMC (Other)[Mandatory]</td>
</tr>
<tr>
<td></td>
<td>JSON Type: Boolean</td>
</tr>
<tr>
<td></td>
<td>true if BMC (Other) management is enabled, false otherwise</td>
</tr>
<tr>
<td>otherBmcAccount</td>
<td>Array of account information to communicate with BMC (Other) of</td>
</tr>
<tr>
<td></td>
<td>component. It is possible to specify it up to five.</td>
</tr>
<tr>
<td></td>
<td>JSON Type: Object</td>
</tr>
<tr>
<td></td>
<td>If &quot;otherBmcManagementValid&quot; is true, this key is [Mandatory]</td>
</tr>
<tr>
<td>user</td>
<td>User name to communicate with BMC (Other). (1-255 characters)</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>If &quot;otherBmcManagementValid&quot; is true, this key is [Mandatory]</td>
</tr>
<tr>
<td>password</td>
<td>Password to communicate with BMC (Other). (1-255 characters)</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>If &quot;otherBmcManagementValid&quot; is true, this key is [Mandatory]</td>
</tr>
</tbody>
</table>

**ResponseBody:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>The URL to get the job status.</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>/esmpro/api/job/status/{jobid}</td>
</tr>
</tbody>
</table>

**Example:**

**Request**

POST /esmpro/api/components

Cookie: JSESSIONID=206C9F1D25E7AB9E1F1FAA8AC51B083

X-ESMPRO-API-Version:1.0

Content-type: application/json; charset=utf-8

```json
{
    "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"
        }
    ],
    "raidManagementValid": false,
    "euManagementEuavValid": false,
    "bmcManagementValid": true,
    "euManagementBmcValid": true,
    "authKey": [  
        "guest"
    ],
    "iloManagementValid": true,
    "iloAccount": [
        {
            "user": "Administrator",
            "password": "Administrator"
        }
    ]
}
```
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:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>registrationResult</td>
<td>Array with data of automatic registration result</td>
</tr>
<tr>
<td>guid</td>
<td>GUID of registered component</td>
</tr>
<tr>
<td>ipAddress</td>
<td>OS IP address of registered component</td>
</tr>
<tr>
<td>bmcIpAddress</td>
<td>Management controller IP address of registered component</td>
</tr>
<tr>
<td>bmcManagementValid</td>
<td>It shows whether the BMC management function is valid or invalid.</td>
</tr>
<tr>
<td>euManagementBmcValid</td>
<td>It shows whether NEC ExpressUpdate (Updates via Management Controller) management function is valid or invalid.</td>
</tr>
<tr>
<td>euManagementEuaValid</td>
<td>It shows whether NEC ExpressUpdate (Updates via NEC ExpressUpdate Agent) management function is valid or invalid.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>raidManagementValid</td>
<td>It shows whether RAID system management function is valid or invalid.</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>1: Not Registered</td>
</tr>
<tr>
<td></td>
<td>2: Registered&lt;invalid&gt;</td>
</tr>
<tr>
<td></td>
<td>3: Registered&lt;valid&gt;</td>
</tr>
<tr>
<td>snmpManagementValid</td>
<td>It shows whether SNMP management function is valid or invalid.</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>1: Not Registered</td>
</tr>
<tr>
<td></td>
<td>2: Registered&lt;invalid&gt;</td>
</tr>
<tr>
<td></td>
<td>3: Registered&lt;valid&gt;</td>
</tr>
<tr>
<td>wsManagementValid</td>
<td>It shows whether WS-Man management function is valid or invalid.</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>1: Not Registered</td>
</tr>
<tr>
<td></td>
<td>2: Registered&lt;invalid&gt;</td>
</tr>
<tr>
<td></td>
<td>3: Registered&lt;valid&gt;</td>
</tr>
<tr>
<td>vproManagementValid</td>
<td>It shows whether vPro management function is valid or invalid.</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>1: Not Registered</td>
</tr>
<tr>
<td></td>
<td>2: Registered&lt;invalid&gt;</td>
</tr>
<tr>
<td></td>
<td>3: Registered&lt;valid&gt;</td>
</tr>
<tr>
<td>iloManagementValid</td>
<td>It shows whether iLO management function is valid or invalid.</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>1: Not Registered</td>
</tr>
<tr>
<td></td>
<td>2: Registered&lt;invalid&gt;</td>
</tr>
<tr>
<td></td>
<td>3: Registered&lt;valid&gt;</td>
</tr>
<tr>
<td>otherBmcManagementValid</td>
<td>It shows whether BMC (Other) management function is valid or invalid.</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>1: Not Registered</td>
</tr>
<tr>
<td></td>
<td>2: Registered&lt;invalid&gt;</td>
</tr>
<tr>
<td></td>
<td>3: Registered&lt;valid&gt;</td>
</tr>
</tbody>
</table>
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",
      "iloManagementValid": "3",
      "otherBmcManagementValid": "1"
    },
    {
      "guid": "5f3cc670-cf1a-11d0-8201-00255ac65b3e",
      "ipAddress": "192.168.14.10",
      "bmcIpAddress": "192.168.14.10",
      "bmcManagementValid": "1",
      "euManagementBmcValid": "1",
      "euManagementEuaValid": "3",
      "raidManagementValid": "3",
      "snmpManagementValid": "1",
      "wsManagementValid": "3",
      "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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>components</td>
<td>Array of management information every components</td>
</tr>
</tbody>
</table>
| 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:

<table>
<thead>
<tr>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /esmpro/api/components</td>
</tr>
<tr>
<td>Cookie: JSESSIONID=206C9F1D25E7AB9E1F1FAA8AC51B083</td>
</tr>
<tr>
<td>X-ESMPRO-API-Version:1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP 1.1 200 OK</td>
</tr>
<tr>
<td>X-ESMPRO-API-Version:1.0</td>
</tr>
<tr>
<td>Content-type: application/json; charset=utf-8</td>
</tr>
</tbody>
</table>

```json
{
  "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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Component name</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>groupName</td>
<td>Belonging group</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>bmcManagementValid</td>
<td>State of BMC management function</td>
</tr>
<tr>
<td></td>
<td>JSON Type : Integer</td>
</tr>
<tr>
<td></td>
<td>1 : Not Registered</td>
</tr>
<tr>
<td></td>
<td>2 : Registered&lt;invalid&gt;</td>
</tr>
<tr>
<td></td>
<td>3 : Registered&lt;valid&gt;</td>
</tr>
<tr>
<td>bmcIpAddress</td>
<td>BMC IP address</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>bmcSubnetMask</td>
<td>Subnet mask of BMC LAN</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>vproManagementValid</td>
<td>State of vPro management function</td>
</tr>
<tr>
<td></td>
<td>JSON Type : Integer</td>
</tr>
<tr>
<td></td>
<td>1 : Not Registered</td>
</tr>
<tr>
<td></td>
<td>2 : Registered&lt;invalid&gt;</td>
</tr>
<tr>
<td></td>
<td>3 : Registered&lt;valid&gt;</td>
</tr>
<tr>
<td>vproUser</td>
<td>User name of vPro management</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>vproIpAddress</td>
<td>IP address of vPro management</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>vproSubnetMask</td>
<td>Subnet mask of vPro management</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>iloManagementValid</td>
<td>State of iLO management function</td>
</tr>
<tr>
<td></td>
<td>JSON Type : Integer</td>
</tr>
<tr>
<td></td>
<td>1 : Not Registered</td>
</tr>
<tr>
<td></td>
<td>2 : Registered&lt;invalid&gt;</td>
</tr>
<tr>
<td></td>
<td>3 : Registered&lt;valid&gt;</td>
</tr>
<tr>
<td>iloUser</td>
<td>User name of iLO management</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>iloIpAddress</td>
<td>IP address of iLO management</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>iloSubnetMask</td>
<td>Subnet mask of iLO management</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>otherBmcManagementValid</td>
<td>State of BMC (Other) management function</td>
</tr>
<tr>
<td></td>
<td>JSON Type : Integer</td>
</tr>
<tr>
<td></td>
<td>1 : Not Registered</td>
</tr>
<tr>
<td></td>
<td>2 : Registered&lt;invalid&gt;</td>
</tr>
<tr>
<td></td>
<td>3 : Registered&lt;valid&gt;</td>
</tr>
<tr>
<td>otherBmcUser</td>
<td>User name of BMC (Other) management</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>otherBmcIpAddress</td>
<td>IP address of BMC (Other) management</td>
</tr>
<tr>
<td>ipAddress</td>
<td>OS IP address of a component</td>
</tr>
<tr>
<td>snmpManagementValid</td>
<td>State of SNMP management function</td>
</tr>
<tr>
<td></td>
<td>1 : Not Registered</td>
</tr>
<tr>
<td></td>
<td>2 : Registered&lt;invalid&gt;</td>
</tr>
<tr>
<td></td>
<td>3 : Registered&lt;valid&gt;</td>
</tr>
<tr>
<td>snmpCommunityNameGet</td>
<td>SNMP community name for read-only (GET) operations. It is used when getting information from SNMP Agent.</td>
</tr>
<tr>
<td>snmpCommunityNameSet</td>
<td>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 ('&quot;')</td>
</tr>
<tr>
<td>wsManagementValid</td>
<td>State of WS-Man management function</td>
</tr>
<tr>
<td></td>
<td>1 : Not Registered</td>
</tr>
<tr>
<td></td>
<td>2 : Registered&lt;invalid&gt;</td>
</tr>
<tr>
<td></td>
<td>3 : Registered&lt;valid&gt;</td>
</tr>
<tr>
<td>wsmanUser</td>
<td>User name of WS-Man management</td>
</tr>
<tr>
<td>wsmanProtocol</td>
<td>Communication protocol of WS-Man management (HTTP/HTTPS)</td>
</tr>
<tr>
<td>wsmanPort</td>
<td>Port number of WS-Man management</td>
</tr>
<tr>
<td>systemManagementType</td>
<td>The type of Agent managed by the SNMP management or the WS-Man management</td>
</tr>
<tr>
<td>osStatusWatch</td>
<td>Alive monitoring of OS (used SNMP)</td>
</tr>
<tr>
<td>registerAlertByStatus</td>
<td>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.</td>
</tr>
<tr>
<td>osStatusPingWatch</td>
<td>Alive monitoring of OS (used ICMP Ping)</td>
</tr>
<tr>
<td>registerAlertByPingStatus</td>
<td>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.</td>
</tr>
<tr>
<td>osStatusWatchInterval</td>
<td>Monitoring interval (minute) of alive monitoring used SNMP and ICMP Ping to OS.</td>
</tr>
<tr>
<td>osStatusWatchRetryCount</td>
<td>Retry count of alive monitoring used SNMP and ICMP Ping to OS.</td>
</tr>
</tbody>
</table>
Example:

<table>
<thead>
<tr>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /esmpro/api/components/server/connection-settings/5f3cc80-cf1b-11e0-8001-00255cc64b2e</td>
</tr>
<tr>
<td>Cookie: JSESSIONID=206C9F1D25E7AB9E1F1F1F9A8A51B083</td>
</tr>
<tr>
<td>X-ESMPRO-API-Version:1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP 1.1 200 OK</td>
</tr>
<tr>
<td>X-ESMPRO-API-Version:1.0</td>
</tr>
<tr>
<td>Content-type : application/json; charset=utf-8</td>
</tr>
</tbody>
</table>

```json
{
    "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": "",
    "otherBmcManagementValid": 1,
    "otherBmcUser": "",
    "otherBmcIpAddress": "",
    "euManagementEuaValid": 3
    "euManagementBmcValid": 2,
    "raidManagementValid": 2,
    "ipAddress": "192.168.14.3",
    "snmpManagementValid": 1,
    "snmpCommunityNameGet": "public",
    "snmpCommunityNameSet": "",
    "wsManagementValid": 1,
    "wsmanUser": "",
    "wsmanProtocol": "",
    "wsmanPort": "",
    "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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
</table>
| 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
<pre><code>             | (1-39 characters)                                                 |
</code></pre>
<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
<th>JSON Type</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>iloIpAddress</td>
<td>IP address of iLO management</td>
<td>String</td>
<td>(Enter the number of 4 sets of 0-255 divided by &quot;.&quot;)</td>
</tr>
<tr>
<td>iloSubnetMask</td>
<td>Subnet mask of iLO management</td>
<td>String</td>
<td>(Enter the number of 4 sets of 0-255 divided by &quot;.&quot;)</td>
</tr>
<tr>
<td>otherBmcManagementValid</td>
<td>State of BMC (Other) management function</td>
<td>Integer</td>
<td>1: invalid, 2 or 3: valid</td>
</tr>
<tr>
<td>otherBmcUser</td>
<td>User name of BMC (Other) management</td>
<td>String</td>
<td>(1-16 characters)</td>
</tr>
<tr>
<td>otherBmcPassword</td>
<td>Password of BMC (Other) management</td>
<td>String</td>
<td>(1-20 characters)</td>
</tr>
<tr>
<td>otherBmcIpAddress</td>
<td>IP address of BMC (Other) management</td>
<td>String</td>
<td>(Enter the number of 4 sets of 0-255 divided by &quot;.&quot;)</td>
</tr>
<tr>
<td>snmpManagementValid</td>
<td>State of SNMP management function</td>
<td>Integer</td>
<td>1: invalid, 2 or 3: valid</td>
</tr>
<tr>
<td>snmpComunityNameGet</td>
<td>SNMP community name for read-only (GET) operations. It is used when getting information from SNMP Agent.</td>
<td>String</td>
<td>(Alphanumeric single-byte and sign (!#$%&amp;)*+-.;&lt;=?@[^_`{</td>
</tr>
<tr>
<td>snmpComunityNameSet</td>
<td>SNMP community name for read-write (SET) operations. It is used when setting information to SNMP Agent. If SNMP community name for setting is not set, it is used SNMP community name for getting.</td>
<td>String</td>
<td>(Alphanumeric single-byte and sign (!#$%&amp;)*+-.;&lt;=?@[^_`{</td>
</tr>
<tr>
<td>wsManagementValid</td>
<td>State of WS-Man management function</td>
<td>Integer</td>
<td>1: invalid, 2 or 3: valid</td>
</tr>
<tr>
<td>wsmanUser</td>
<td>User name of WS-Man management</td>
<td>String</td>
<td>(1-31 characters)</td>
</tr>
<tr>
<td>wsmanPassword</td>
<td>Password of WS-Man management</td>
<td>String</td>
<td>(1-255 characters)</td>
</tr>
<tr>
<td>wsmanProtocol</td>
<td>Communication protocol of WS-Man management (HTTP/HTTPS)</td>
<td>String</td>
<td>Default value: &quot;HTTP&quot;</td>
</tr>
<tr>
<td>wsmanPort</td>
<td>Port number of WS-Man management</td>
<td>String</td>
<td>(Alphanumeric single-byte of 1-5 characters) Default value: &quot;5985/5986&quot;</td>
</tr>
<tr>
<td>osStatusWatch</td>
<td>State of Alive monitoring of OS (used SNMP)</td>
<td>Boolean</td>
<td>true if alive monitoring is enabled, false otherwise</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>JSON Type</td>
<td>Default value</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>registerAlertByStatus</td>
<td>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.</td>
<td>Boolean</td>
<td>false</td>
</tr>
<tr>
<td>osStatusPingWatch</td>
<td>Alive monitoring of OS (used ICMP Ping)</td>
<td>Boolean</td>
<td>false</td>
</tr>
<tr>
<td>registerAlertByPingStatus</td>
<td>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.</td>
<td>Boolean</td>
<td>false</td>
</tr>
<tr>
<td>osStatusWatchInterval</td>
<td>Monitoring interval (minute) of alive monitoring used SNMP and ICMP Ping to OS.</td>
<td>Integer</td>
<td>1</td>
</tr>
<tr>
<td>osStatusWatchRetryCount</td>
<td>Retry count of alive monitoring used SNMP and ICMP Ping to OS. (0-100 count)</td>
<td>Integer</td>
<td>0</td>
</tr>
</tbody>
</table>

**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,
  "euManagementEuaValid": 3,
  "euManagementBmcValid": 2,
  "raidManagementValid": 2,
  "ipAddress": "192.168.14.3",
  "snmpManagementValid": 1,
  "wsManagementValid": 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:**

<table>
<thead>
<tr>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE</td>
</tr>
<tr>
<td>/esmpro/api/components/server/connection-settings/5f3cc680-cf1b-11e0-8001-00255cc64b2e?force=true</td>
</tr>
<tr>
<td>Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAAA8AC51B083</td>
</tr>
<tr>
<td>X-ESMPRO-API-Version:1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP 1.1 200 OK</td>
</tr>
<tr>
<td>X-ESMPRO-API-Version:1.0</td>
</tr>
</tbody>
</table>
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:
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>powerState</td>
<td>Power State</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>&quot;On&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Off&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Unknown&quot;</td>
</tr>
</tbody>
</table>

Example:

Request
GET /esmpro/api/components/server/power-control/5f3cc680-cflb-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1FAA8AC51B083
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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>powerState</td>
<td>Power Operation[Mandatory]</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>Possible operations are as follows:</td>
</tr>
<tr>
<td></td>
<td>&quot;On&quot;: Power On</td>
</tr>
<tr>
<td></td>
<td>&quot;Off&quot;: Power Off</td>
</tr>
<tr>
<td></td>
<td>&quot;Reset&quot;: Reset</td>
</tr>
<tr>
<td></td>
<td>&quot;Shutdown&quot;: Shutdown</td>
</tr>
<tr>
<td></td>
<td>&quot;PowerCycle&quot;: Power Cycle</td>
</tr>
<tr>
<td></td>
<td>&quot;Dump&quot;: Push Dump Switch</td>
</tr>
<tr>
<td></td>
<td>&quot;ForceShutdown&quot;: Shutdown via BMC</td>
</tr>
<tr>
<td></td>
<td>If you specify the operation other than the above operation, this API will fail.</td>
</tr>
</tbody>
</table>

ResponseBody:
Nothing

Example:
```
Request
PUT /esmpro/api/components/server/power-control/5f3cc680-cf1b-11e0-8001-00255ce64b2e
Cookie: JSESSIONID=206C9F1D25E7A9E1F1AFAA8AC51B083
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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>The URL to get the job status. JSON Type: String /esmpro/api/job/status/{jobid}</td>
</tr>
</tbody>
</table>

Example:

```
Request
GET /esmpro/api/components/server/sensor-all/5f3cc680-cf1b-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1FAA8A51B083
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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sensor</td>
<td>Array of sensor data.</td>
</tr>
<tr>
<td>identifier</td>
<td>Sensor Identifier</td>
</tr>
<tr>
<td>recordId</td>
<td>SDR Record ID in hexadecimal</td>
</tr>
<tr>
<td>name</td>
<td>Sensor Name. ID String or the name to identify the sensor.</td>
</tr>
<tr>
<td>sensorType</td>
<td>Sensor Type Name.</td>
</tr>
<tr>
<td>currentValue</td>
<td>Current Value</td>
</tr>
<tr>
<td>upperNonRecoverable</td>
<td>Upper non recoverable</td>
</tr>
<tr>
<td>upperCritical</td>
<td>Upper critical</td>
</tr>
<tr>
<td>upperNonCritical</td>
<td>Upper non critical</td>
</tr>
<tr>
<td>lowerNonRecoverable</td>
<td>Lower non recoverable</td>
</tr>
<tr>
<td>lowerCritical</td>
<td>Lower critical</td>
</tr>
<tr>
<td>lowerNonCritical</td>
<td>Lower non critical</td>
</tr>
<tr>
<td>currentStatus</td>
<td>Current sensor status</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>Sensor Identifier. &lt;br&gt;JSON Type : String</td>
</tr>
<tr>
<td>recordId</td>
<td>SDR Record ID &lt;br&gt;JSON Type : String</td>
</tr>
<tr>
<td>name</td>
<td>Sensor Name. ID String or the name to identify the sensor. &lt;br&gt;JSON Type : String</td>
</tr>
<tr>
<td>sensorType</td>
<td>Sensor type Name &lt;br&gt;JSON Type : String</td>
</tr>
<tr>
<td>currentValue</td>
<td>Current Value &lt;br&gt;JSON Type : String</td>
</tr>
<tr>
<td>upperNonRecoverable</td>
<td>Upper non recoverable &lt;br&gt;JSON Type : String</td>
</tr>
<tr>
<td>upperCritical</td>
<td>Upper critical &lt;br&gt;JSON Type : String</td>
</tr>
<tr>
<td>upperNonCritical</td>
<td>Upper non critical &lt;br&gt;JSON Type : String</td>
</tr>
<tr>
<td>lowerNonRecoverable</td>
<td>Lower non recoverable &lt;br&gt;JSON Type : String</td>
</tr>
<tr>
<td>lowerCritical</td>
<td>Lower critical &lt;br&gt;JSON Type : String</td>
</tr>
<tr>
<td>lowerNonCritical</td>
<td>Lower non critical &lt;br&gt;JSON Type : String</td>
</tr>
</tbody>
</table>
| currentStatus     | Current sensor status <br>JSON Type : String<br>In normal case, "Normal" will be returned.<br>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:<br>`- "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:

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /esmpro/api/components/server/sensor/5f3cc680-cf1b-11e0-8001-00255cc64b2e?identifier=0002</td>
<td>HTTP 1.1  200  OK</td>
</tr>
<tr>
<td>Cookie: JSESSIONID=206C9F1D25E7AB9E1F1FAAA8AC51B083</td>
<td>X-ESMPRO-API-Version:1.0</td>
</tr>
<tr>
<td>X-ESMPRO-API-Version:1.0</td>
<td>Content-type: application/json; charset=utf-8</td>
</tr>
</tbody>
</table>

```json
{
  "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:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>osType</td>
<td>OS Name of component, or OS Name and OS Version of component.</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>If value cannot be fetched from component then &quot;&quot; (empty string) will</td>
</tr>
<tr>
<td></td>
<td>be returned.</td>
</tr>
</tbody>
</table>

**Example:**

**Request**
```
GET /esmpro/api/components/server/system-info/5f3cc680-cflb-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAAA8A51B083
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 2012 Datacenter x64"
}
```
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:
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>productName</td>
<td>Product Name of the component</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>If product name cannot be fetched from component then &quot;&quot; (empty string) will be returned.</td>
</tr>
<tr>
<td>manufacturerId</td>
<td>Manufacturer ID</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>If manufacturer ID cannot be fetched from component then &quot;&quot; (empty string) will be returned.</td>
</tr>
<tr>
<td>productId</td>
<td>Product ID</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>physicalCPU</td>
<td>Physical CPU Information</td>
</tr>
<tr>
<td></td>
<td>JSON Type : Array</td>
</tr>
<tr>
<td>name</td>
<td>CPU Name</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>If CPU name cannot be fetched from component then &quot;&quot; (empty string) will be returned.</td>
</tr>
<tr>
<td>version</td>
<td>Version</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>If version cannot be fetched from component then &quot;&quot; (empty string) will be returned.</td>
</tr>
<tr>
<td>internalSpeed</td>
<td>Internal Speed</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>If internal speed cannot be fetched from component then &quot;&quot; (empty string) will be returned.</td>
</tr>
<tr>
<td>maxCoreNum</td>
<td>Number of Max cores</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>If Max Core Number cannot be fetched from component then &quot;&quot; (empty string) will be returned.</td>
</tr>
<tr>
<td>validCoreNum</td>
<td>Number of effective cores</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>If effective Core Number cannot be fetched from component then &quot;&quot; (empty string) will be returned.</td>
</tr>
<tr>
<td>validThreadNum</td>
<td>Number of effective Thread</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>If effective Thread Number cannot be fetched from component then &quot;&quot; (empty string) will be returned.</td>
</tr>
<tr>
<td>operationalStatus</td>
<td>State of CPU operation</td>
</tr>
<tr>
<td></td>
<td>JSON Type : Integer</td>
</tr>
<tr>
<td></td>
<td>1 - Other</td>
</tr>
<tr>
<td></td>
<td>2 - Unknown</td>
</tr>
<tr>
<td></td>
<td>3 - Normal</td>
</tr>
<tr>
<td></td>
<td>4 - Warning</td>
</tr>
<tr>
<td></td>
<td>5 - Abnormal</td>
</tr>
<tr>
<td>memoryTotalCapacity</td>
<td>Total value of physical memory</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>If total memory capacity cannot be fetched from component then &quot;&quot; (empty string) will be returned.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>memoryBank</th>
<th>Memory Bank Information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JSON Type: Array</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>size</th>
<th>Memory Bank Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>If memory bank size cannot be fetched from component then &quot;&quot; (empty string) will be returned.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>status</th>
<th>Status of Memory Bank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JSON Type: Integer</td>
</tr>
<tr>
<td></td>
<td>1 - Other</td>
</tr>
<tr>
<td></td>
<td>2 - Unknown</td>
</tr>
<tr>
<td></td>
<td>3 - Normal</td>
</tr>
<tr>
<td></td>
<td>4 - Warning</td>
</tr>
<tr>
<td></td>
<td>5 - Abnormal</td>
</tr>
<tr>
<td></td>
<td>6 - Non-recoverable</td>
</tr>
<tr>
<td></td>
<td>7 - In test</td>
</tr>
<tr>
<td></td>
<td>8 - Degraded</td>
</tr>
<tr>
<td></td>
<td>9 - Not attached</td>
</tr>
<tr>
<td></td>
<td>10 - Down</td>
</tr>
<tr>
<td></td>
<td>11 - Standby</td>
</tr>
</tbody>
</table>
Example:

**Request**

```plaintext
GET /esmpro/api/components/server/hw/5f3cc680-cf11e0-8001-00255cc64b2e
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

```json
{
  "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
    }
  ],
  "memoryTotalCapacity": "8388608 KB",
  "memoryBank": [
    {
      "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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>The URL to get the job status.</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>/esmpro/api/job/status/{jobid}</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>networkInterface</td>
<td>Network Interface Information</td>
</tr>
<tr>
<td></td>
<td>JSON Type: Array</td>
</tr>
<tr>
<td>id</td>
<td>Network Interface ID</td>
</tr>
<tr>
<td></td>
<td>JSON Type: Integer</td>
</tr>
<tr>
<td></td>
<td>If ID cannot be fetched from component then null will be returned.</td>
</tr>
<tr>
<td>macAddress</td>
<td>MAC Address</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>If MAC Address cannot be fetched from component then &quot;&quot; (empty string) will be returned.</td>
</tr>
<tr>
<td>adapterType</td>
<td>Network Adapter Type</td>
</tr>
<tr>
<td></td>
<td>JSON Type: Integer</td>
</tr>
<tr>
<td></td>
<td>0 : Virtual Network Adapter</td>
</tr>
<tr>
<td></td>
<td>1 : Physical Network Adapter</td>
</tr>
<tr>
<td></td>
<td>2 : Unknown (If an adapterType cannot be fetched, 2 will be returned.)</td>
</tr>
<tr>
<td>manufacturer</td>
<td>Manufacturer of Network Adapter</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>If manufacturer cannot be fetched from component then &quot;&quot; (empty string) will be returned.</td>
</tr>
</tbody>
</table>

**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

```json
{
  "networkInterface" : [
    {
      "id":1,
      "adapterType": 0,
      "manufacturer": "Microsoft"
    },
    {
      "id":2,
      "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:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>The URL to get the job status. JSON Type: String /esmpro/api/job/status/{jobid}</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
</table>
| 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:  
            | EN) 2/18/2015 17:05:22 (MM/DD/YYYY HH:MM:SS)                        |
| detail     | Detail  
            | JSON Type : String                                                  |
| dump       | SEL Dump Data in hexadecimal  
            | 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 6c 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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>frulist</td>
<td>FRU List</td>
</tr>
<tr>
<td>frulist[n].fruId</td>
<td>FRU Device ID</td>
</tr>
<tr>
<td></td>
<td>JSON Type: Integer</td>
</tr>
<tr>
<td></td>
<td>Numerical value of 0 to 255</td>
</tr>
<tr>
<td>frulist[n].description</td>
<td>FRU Name</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>FRU Name</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>chassisInformation</td>
<td>Chassis Information</td>
</tr>
<tr>
<td></td>
<td>If &quot;Chassis Information&quot; is not present, &quot;&quot; (empty string) is returned in each field.</td>
</tr>
<tr>
<td>chassisType</td>
<td>Chassis Type</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>chassisSerialNumber</td>
<td>The serial number of Chassis</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>chassisCharacteristics</td>
<td>The characteristics of Chassis</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>boardInformation</td>
<td>Board Information</td>
</tr>
<tr>
<td></td>
<td>If &quot;Board Information&quot; is not present, &quot;&quot; (empty string) is returned in each field.</td>
</tr>
<tr>
<td>mfgDateTime</td>
<td>Manufacturing Date Time</td>
</tr>
<tr>
<td></td>
<td>JSON Type : string</td>
</tr>
<tr>
<td></td>
<td>Format :</td>
</tr>
<tr>
<td></td>
<td>EN) 2/18/2015 17:05:22 (MM/DD/YYYY HH:MM:SS)</td>
</tr>
<tr>
<td>boardManufacturer</td>
<td>Board Manufacturer</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>boardProductName</td>
<td>Board Product Name</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>boardSerialNumber</td>
<td>Board Serial Number</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>boardPartNumber</td>
<td>Board Part Number</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>boardVersion</td>
<td>Board Version</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>productInformation</td>
<td>Product Information</td>
</tr>
<tr>
<td></td>
<td>If &quot;Product Information&quot; is not present, &quot;&quot; (empty string) is returned in each field.</td>
</tr>
<tr>
<td>mfgDateTime</td>
<td>Manufacturing Date Time</td>
</tr>
<tr>
<td></td>
<td>JSON Type : string</td>
</tr>
<tr>
<td></td>
<td>Format :</td>
</tr>
<tr>
<td></td>
<td>EN) 2/18/2015 17:05:22 (MM/DD/YYYY HH:MM:SS)</td>
</tr>
<tr>
<td>productName</td>
<td>Product Name</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>productPartModelNumber</td>
<td>Product Part Model Number</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>productVersion</td>
<td>FR Version</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>productSerialNumber</td>
<td>Product Serial Number</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>Key</td>
<td>Value</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>assertTag</td>
<td>Assert Tag</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>dump</td>
<td>Hexadecimal dump data of FRU Record</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
</tbody>
</table>

FRU Type: Memory

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>FRU Name</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>memoryType</td>
<td>Memory Type</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>dump</td>
<td>Hexadecimal dump data of FRU Record</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
</tbody>
</table>

FRU Type: CPU

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>FRU Name</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>dump</td>
<td>Hexadecimal dump data of FRU Record</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
</tbody>
</table>

Example:

Request

GET /esmpro/api/components/server/fru/5f3cc680-cf1b-11e0-8001-00255cc64b2e?fruId=0
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

```json
{
  "boardInformation": {
    "boardManufacturer": "GIGABYTE",
    "boardProductName": "GA-6UASVI",
    "boardSerialNumber": "0BM550700148",
    "boardVersion": "1.1H",
    "boardPartNumber": "56-131335-001"
  },
  "productInformation": {
    "productVersion": "FR1.3",
    "assertTag": "_____________",
    "productSerialNumber": "1700121",
    "manufacturerName": "NEC",
    "productName": "Express5800/R110d-1E",
    "productPartModelNumber": "[N8100-1764]"
  },
  "description": "Primary FRU Device",
  "chassisInformation": {
    "chassisType": "Rack Mount Chassis",
    "chassisPartNumber": "856-131336-001",
    "chassisSerialNumber": "Rack Mount",
    "chassisCharacteristics": "01h"
  }
}
```
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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>componentName</td>
<td>Name of the target component. JSON Type : String</td>
</tr>
<tr>
<td>componentGUID</td>
<td>GUID of the target component JSON Type : String</td>
</tr>
</tbody>
</table>

Response Body:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>The URL to get the job status. JSON Type : String</td>
</tr>
</tbody>
</table>

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": "5f3c680-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:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmpManagement</td>
<td>The result of detection of SNMP management.</td>
</tr>
<tr>
<td>detected</td>
<td>The result of detection of SNMP management.</td>
</tr>
<tr>
<td>detail</td>
<td>A detail message of result of connection check execution.</td>
</tr>
<tr>
<td>JSON Type : String</td>
<td></td>
</tr>
<tr>
<td>wsmanManagement</td>
<td>The result of detection of WSMAN management.</td>
</tr>
<tr>
<td>detected</td>
<td>The result of detection of WSMAN management.</td>
</tr>
<tr>
<td>detail</td>
<td>A detail message of result of connection check execution.</td>
</tr>
<tr>
<td>JSON Type : String</td>
<td></td>
</tr>
<tr>
<td>raidManagement</td>
<td>The result of detection RAID System management.</td>
</tr>
<tr>
<td>detected</td>
<td>The result of detection RAID System management.</td>
</tr>
<tr>
<td>detail</td>
<td>A detail message of result of connection check execution.</td>
</tr>
<tr>
<td>JSON Type : String</td>
<td></td>
</tr>
<tr>
<td>expressUpdateManage-</td>
<td>The results of detection EU Management.</td>
</tr>
<tr>
<td>detected</td>
<td>The results of detection EU Management.</td>
</tr>
<tr>
<td>detail</td>
<td>A detail message of result of connection check execution.</td>
</tr>
<tr>
<td>JSON Type : String</td>
<td></td>
</tr>
<tr>
<td>bmcManagement</td>
<td>The results of detection BMC Management.</td>
</tr>
<tr>
<td>detected</td>
<td>The results of detection BMC Management.</td>
</tr>
<tr>
<td>detail</td>
<td>A detail message of result of connection check execution.</td>
</tr>
<tr>
<td>JSON Type : String</td>
<td></td>
</tr>
<tr>
<td>vproManagement</td>
<td>The result of detection vPro Management.</td>
</tr>
<tr>
<td>detected</td>
<td>The result of detection vPro Management.</td>
</tr>
<tr>
<td>detail</td>
<td>A detail message of result of connection check execution.</td>
</tr>
<tr>
<td>JSON Type : String</td>
<td></td>
</tr>
<tr>
<td>iloManagement</td>
<td>The result of detection iLO Management.</td>
</tr>
<tr>
<td>detected</td>
<td>The result of detection iLO Management.</td>
</tr>
<tr>
<td>detail</td>
<td>A detail message of result of connection check execution.</td>
</tr>
<tr>
<td>JSON Type : String</td>
<td></td>
</tr>
<tr>
<td>otherBmcManagement</td>
<td>The result of detection BMC (Other) Management.</td>
</tr>
<tr>
<td>detected</td>
<td>The result of detection BMC (Other) Management.</td>
</tr>
<tr>
<td>detail</td>
<td>A detail message of result of connection check execution.</td>
</tr>
<tr>
<td>JSON Type : String</td>
<td></td>
</tr>
</tbody>
</table>
Example:

<table>
<thead>
<tr>
<th>Request</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /esmpro/api/components/server/connection-check/csc00001/result</td>
<td></td>
</tr>
<tr>
<td>Cookie: JSESSIONID=206C9F1D25E7AB9E1F1FAA8AC51B083</td>
<td></td>
</tr>
<tr>
<td>X-ESMPRO-API-Version: 1.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP 1.1 200 OK</td>
<td></td>
</tr>
<tr>
<td>X-ESMPRO-API-Version: 1.0</td>
<td></td>
</tr>
<tr>
<td>Content-type: application/json; charset=utf-8</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>&quot;wsmanManagement&quot;: {</td>
<td></td>
</tr>
<tr>
<td>&quot;detected&quot;: &quot;true&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;detail&quot;: &quot;WS-MAN management can be used.&quot;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>&quot;raidManagement&quot;: {</td>
<td></td>
</tr>
<tr>
<td>&quot;detected&quot;: &quot;true&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;detail&quot;: &quot;RAID system management can be used.&quot;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>&quot;expressUpdateManagement&quot;: {</td>
<td></td>
</tr>
<tr>
<td>&quot;detected&quot;: &quot;true&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;detail&quot;: &quot;Using the NEC ExpressUpdate Agent NEC ExpressUpdate function can be used.&quot;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>&quot;bmcManagement&quot;: {</td>
<td></td>
</tr>
<tr>
<td>&quot;detected&quot;: &quot;true&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;detail&quot;: &quot;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.&quot;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>
**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:**

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

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>smName</td>
<td>Name of NEC ESMPRO Manager. JSON Type: String</td>
</tr>
<tr>
<td>smVersion</td>
<td>Version number of NEC ESMPRO Manager. JSON Type: String</td>
</tr>
</tbody>
</table>

**Example:**

**Request**

GET /esmpro/api/sm
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1FAA8AC51B083
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

```json
{
    "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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>This shows the status of a specified component.</td>
</tr>
<tr>
<td></td>
<td>JSON Type : Integer</td>
</tr>
<tr>
<td></td>
<td>0: NO_MONITORING</td>
</tr>
<tr>
<td></td>
<td>1: NORMAL</td>
</tr>
<tr>
<td></td>
<td>2: UNKNOWN</td>
</tr>
<tr>
<td></td>
<td>3: DC-OFF, POST, OS Panic</td>
</tr>
<tr>
<td></td>
<td>4: WARNING</td>
</tr>
<tr>
<td></td>
<td>5: ERROR</td>
</tr>
</tbody>
</table>

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:

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

### Request:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Component name registered on NEC ESMPRO Manager</td>
</tr>
</tbody>
</table>

### ResponseBody:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Component Status</td>
</tr>
<tr>
<td></td>
<td>JSON Type: Integer</td>
</tr>
<tr>
<td></td>
<td>0: NO_MONITORING</td>
</tr>
<tr>
<td></td>
<td>1: NORMAL</td>
</tr>
<tr>
<td></td>
<td>2: UNKNOWN</td>
</tr>
<tr>
<td></td>
<td>3: DC-OFF, POST, OS Panic</td>
</tr>
<tr>
<td></td>
<td>4: WARNING</td>
</tr>
<tr>
<td></td>
<td>5: ERROR</td>
</tr>
</tbody>
</table>

### Example:

#### Request

```plaintext
GET /esmpro/api/components/server/server-status?name=ManagementController
```

```plaintext
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1FAA8A51B083
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 ExpEther Manager List

This API gets an information list of ExpEther Manager.

URL:

GET /esmpro/api/eem

Request:
Nothing

ResponseBody:

<table>
<thead>
<tr>
<th>Key</th>
<th>Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>eems</td>
<td>Array of management information every EEM</td>
</tr>
<tr>
<td>status</td>
<td>state of ExpEther Manager</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>unknown</td>
</tr>
<tr>
<td></td>
<td>error</td>
</tr>
<tr>
<td></td>
<td>warning</td>
</tr>
<tr>
<td></td>
<td>normal</td>
</tr>
<tr>
<td>monitoring</td>
<td>monitoring status</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>enabled</td>
</tr>
<tr>
<td></td>
<td>disabled</td>
</tr>
<tr>
<td>name</td>
<td>Registered ExpEther Manager name</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>interval</td>
<td>monitoring interval(2nd)</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>ipAddress</td>
<td>IP address for ExpEther Manager access</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>Enter the number of 4 sets of 0-255 divided by &quot;.&quot;</td>
</tr>
<tr>
<td>protocol</td>
<td>HTTP or HTTPS</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>port</td>
<td>Port number for ExpEther Manager access</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>user</td>
<td>User name of Basic authentication of ExpEther Manager access</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
</tbody>
</table>
Example:

<table>
<thead>
<tr>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /esmpro/api/eem</td>
</tr>
<tr>
<td>Cookie: JSESSIONID=206C9F1D25E7AB9E1F1FAA8AC51B083</td>
</tr>
<tr>
<td>X-ESMPRO-API-Version:1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP 1.1 200 OK</td>
</tr>
<tr>
<td>X-ESMPRO-API-Version:1.0</td>
</tr>
<tr>
<td>Content-type: application/json; charset=utf-8</td>
</tr>
</tbody>
</table>

```json
{
  "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"
    }
  ]
}
```
2.2.23 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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>discoveryMode</td>
<td>IP address range search or a network address search. [Mandatory]</td>
</tr>
<tr>
<td></td>
<td>JSON Type: Integer</td>
</tr>
<tr>
<td></td>
<td>0: IP Address Range Search</td>
</tr>
<tr>
<td></td>
<td>1: Network Address Search</td>
</tr>
<tr>
<td></td>
<td>The value of other than the above will be an error.</td>
</tr>
<tr>
<td>startAddress</td>
<td>Start IP address (If discoveryMode is Network Address Search, this key is ignored.)</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>Enter the number of 4 sets of 0-255 divided by &quot;.&quot;</td>
</tr>
<tr>
<td>endAddress</td>
<td>End IP address (If discoveryMode is Network Address Search, this key is ignored.)</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>Enter the number of 4 sets of 0-255 divided by &quot;.&quot;</td>
</tr>
<tr>
<td>networkAddress</td>
<td>Network address (If discoveryMode is IP Address Range Search, this key is ignored.)</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>Enter the number of 4 sets of 0-255 divided by &quot;.&quot;</td>
</tr>
<tr>
<td>networkMask</td>
<td>Network mask (If discoveryMode is IP Address Range Search, this key is ignored.)</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>Enter the number of 4 sets of 0-255 divided by &quot;.&quot;</td>
</tr>
<tr>
<td>port</td>
<td>Port number (Optional and default value is 30500)</td>
</tr>
<tr>
<td></td>
<td>JSON Type: Integer</td>
</tr>
<tr>
<td>accounts</td>
<td>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.</td>
</tr>
<tr>
<td>user</td>
<td>User name of Basic authentication of ExpEther Manager access (*)</td>
</tr>
<tr>
<td></td>
<td>[Mandatory]</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String (1-255 character)</td>
</tr>
<tr>
<td>password</td>
<td>Password of Basic authentication of ExpEther Manager access (*)</td>
</tr>
<tr>
<td></td>
<td>[Mandatory]</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String (1-255 character)</td>
</tr>
</tbody>
</table>

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

ResponseBody:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>The URL to get the job status.</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>/esmpro/api/job/status/{jobid}</td>
</tr>
</tbody>
</table>
### Example:

#### Request

<table>
<thead>
<tr>
<th>POST /esmpro/api/eem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8AC51B083</td>
</tr>
<tr>
<td>X-ESMPRO-API-Version: 1.0</td>
</tr>
<tr>
<td>Content-type: application/json; charset=utf-8</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>&quot;discoveryMode&quot;: 0,</td>
</tr>
<tr>
<td>&quot;startAddress&quot;: &quot;192.168.1.1&quot;,</td>
</tr>
<tr>
<td>&quot;endAddress&quot;: &quot;192.168.1.20&quot;,</td>
</tr>
<tr>
<td>&quot;accounts&quot;: [</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>&quot;user&quot;: &quot;Administrator&quot;,</td>
</tr>
<tr>
<td>&quot;password&quot;: &quot;Administrator&quot;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>]</td>
</tr>
<tr>
<td>}</td>
</tr>
</tbody>
</table>

#### Response

<table>
<thead>
<tr>
<th>HTTP 1.1 200 OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-ESMPRO-API-Version: 1.0</td>
</tr>
<tr>
<td>Content-type: application/json; charset=utf-8</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>&quot;url&quot;: &quot;/esmpro/api/job/status/eem00001&quot;</td>
</tr>
<tr>
<td>}</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>result</td>
<td>Array with data of a registration result</td>
</tr>
<tr>
<td></td>
<td>JSON Type : Object</td>
</tr>
<tr>
<td>name</td>
<td>Registered ExpEther Manager name</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td>ipAddress</td>
<td>IP address for ExpEther Manager access</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>Enter the number of 4 sets of 0-255 divided by &quot;.&quot;</td>
</tr>
<tr>
<td>status</td>
<td>state of ExpEther Manager</td>
</tr>
<tr>
<td></td>
<td>JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>unknown</td>
</tr>
<tr>
<td></td>
<td>error</td>
</tr>
<tr>
<td></td>
<td>warning</td>
</tr>
<tr>
<td></td>
<td>normal</td>
</tr>
</tbody>
</table>

Example:

Request:

GET /esmpro/api/eem/eem00001/result
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1FAA8AC51B083
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

```json
{
  "result": [
    {
      "name": "ExpressEtherManager001",
      "ipAddress": "192.168.1.4",
      "status": "normal"
    }
  ]
}
```
2.2.24 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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Registered ExpEther Manager name</td>
</tr>
</tbody>
</table>

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 ,
```
2.2.25 **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:**

```plaintext
GET /esmpro/api/components/server/iml/{guid}
```

**Request:**

Nothing

**ResponseBody:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>The URL to get the job status. JSON Type : String</td>
</tr>
<tr>
<td></td>
<td>/esmpro/api/job/status/{jobid}</td>
</tr>
</tbody>
</table>

**Example:**

**Request**

```plaintext
GET /esmpro/api/esmpro/api/components/server/iml/5f3cc680-cf1b-11e0-8001-00255cc64b2e
Cookie: JSESSIONID=206C9F1D25E7AB9E1F1AFAA8A C51B083
X-ESMPRO-API-Version:1.0
```

**Response**

```plaintext
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:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>iml</td>
<td>Array of IML</td>
</tr>
<tr>
<td>id</td>
<td>Id of IML</td>
</tr>
<tr>
<td>severity</td>
<td>Severity of IML</td>
</tr>
<tr>
<td>eventClass</td>
<td>Event Class Code (in decimal). Identifies the type of event that occurred.</td>
</tr>
<tr>
<td>className</td>
<td>Event Class Name. The name of class that identifies the type of event that</td>
</tr>
<tr>
<td></td>
<td>occurred, for example, network, maintenance, and so on.</td>
</tr>
<tr>
<td>eventCode</td>
<td>Event code (in decimal)</td>
</tr>
<tr>
<td>repaired</td>
<td>&quot;Repaired&quot; shows that an event has undergone corrective action.</td>
</tr>
<tr>
<td></td>
<td>true : repaired</td>
</tr>
<tr>
<td></td>
<td>false : not repaired</td>
</tr>
<tr>
<td>lastUpdate</td>
<td>The date and time when the latest event of this type occurred.</td>
</tr>
<tr>
<td>initialUpdate</td>
<td>The date and time when the first event of this type occurred.</td>
</tr>
<tr>
<td>count</td>
<td>The number of times this event has occurred.</td>
</tr>
<tr>
<td>description</td>
<td>Description of event.</td>
</tr>
<tr>
<td></td>
<td>recommendedAction : Recommended recovery action.</td>
</tr>
</tbody>
</table>
### 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
    }
  ]
}
```
2.2.26 Get Group Status
This API gets the status of a specified component group.

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

Request:
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Group Name</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>If this key is omitted, &quot;root&quot; is specified.</td>
</tr>
</tbody>
</table>

Response Body:
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>The URL to get the job status.</td>
</tr>
<tr>
<td></td>
<td>JSON Type: String</td>
</tr>
<tr>
<td></td>
<td>/esmpro/api/job/status/{jobid}</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>This value shows the group status.</td>
</tr>
<tr>
<td></td>
<td>JSON Type : Integer</td>
</tr>
<tr>
<td></td>
<td>0 : NO_MONITORING</td>
</tr>
<tr>
<td></td>
<td>1 : NORMAL</td>
</tr>
<tr>
<td></td>
<td>2 : UNKNOWN</td>
</tr>
<tr>
<td></td>
<td>3 : DC-OFF</td>
</tr>
<tr>
<td></td>
<td>4 : WARNING</td>
</tr>
<tr>
<td></td>
<td>5 : ERROR</td>
</tr>
</tbody>
</table>

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
}
```