



## REST API Guide

---

**EPMWARE®**

**EPMWARE, Inc.**

Updated: May 2025

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of EPMWARE Inc. EPMWARE, Inc. may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from EPMWARE, Inc., the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Author: Tony Kiratsous, [tony@EPMWARE.com](mailto:tony@EPMWARE.com)

© 2025 **EPMWARE**, Inc. All rights reserved. [www.EPMWARE.com](http://www.EPMWARE.com) | 408-614-0442

**EPMWARE®** is a registered trademark of **EPMWARE**, Inc. in the United States and/or other countries.

All other trademarks are property of their respective owners.

# Contents

---

- REST API Guide ..... 1**
- Contents..... 3**
- EPMWARE REST API Reference ..... 4**
  - Technical Details ..... 4
    - Authentication..... 4
    - URL ..... 4
    - Module: Task..... 6
    - Module: ERP ..... 8
    - Module: Deployment ..... 10
    - Module: Export ..... 12
    - Module: Security..... 17

# EPMWARE REST API Reference

---

The purpose of this appendix is to provide details for the REST API implementation for EPMWARE modules. Developers can connect to the EPMWARE application using the REST API to perform various tasks triggered from external applications such as execute ERP Import processes and run Deployment task.

## Technical Details

### Authentication

For authentication, you will need to generate token for a user in EPMWARE and pass it in REST API header during every call.

To generate a token in EPMware :

Go to Security Module → Users → Right-click on the user record → Select “Generate token”

The authentication header value should be:

Token <token generated in EPMWARE>

### URL

The REST API URL format is:

`http(s)://<EPMWARE_URL>/service/api/<module>/<action>`

The **EPMWARE\_URL** is different for every client and every instance. Refer to the following list of Modules and actions allowed within each module:

Modules and their related Actions are as listed below. Actions are listed inside Modules.

| Module            | Action                | Notes                                                                 |
|-------------------|-----------------------|-----------------------------------------------------------------------|
| <b>Task</b>       | get_status            | Get status for a given task ID                                        |
|                   | get_log_file          | Download Task Details                                                 |
| <b>ERP</b>        | upload_file           | Upload Metadata file and load its data in ERP Import interface table. |
|                   | run                   | Execute ERP Import (one time scheduled)                               |
| <b>Deployment</b> | run                   | Execute deployment (one time scheduled)                               |
|                   | get_execution_details | Get execution details for given deployment name                       |
| <b>Export</b>     | run                   | Execute export for a profile or book                                  |
|                   | Download_file         | To download export file                                               |
|                   | get_execution_details | Get execution details for given export name                           |
| <b>Security</b>   | create                | To create new SSO user.                                               |
|                   | update                | To update the details existing SSO user.                              |
|                   | disable               | To disable existing SSO user/                                         |
|                   | enable                | To enable existing SSO user                                           |
|                   | assign_groups         | To add SSO user to given security groups                              |
|                   | unassign_groups       | To remove SSO users from the given security groups                    |
|                   | users                 | To get all the details of native or saml users.                       |
|                   | usergroups            | To get the list of security groups assigned to the given user         |
|                   | groups                | To get details of all the security groups.                            |
|                   | groupuser             | To get the list of users tagged to the security group.                |

For example, [https://client.epmwarecloud.com/service/api.task/get\\_status](https://client.epmwarecloud.com/service/api.task/get_status)

## Module: Task

### Action: Get Task Status

This API will return % Complete, Status and message. You may need to call this API after other APIs which upon submission returns Task ID. API will return one of the following three values for the status code. If status code is N, then it means task is not complete yet and you may either need to wait or perform this check in a loop till status becomes either S or E.

| Status | Meaning | Notes                                                                                                                                                                       |
|--------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| N      | New     | This status code is the initial status applied when the task is submitted in the queue.                                                                                     |
| S      | Success | Task is completed successfully.                                                                                                                                             |
| E      | Error   | Task completed but with Error. Check the "Message" value for short Error Message. For complete task details, call "get_log_file" REST API as explained in the next section. |

### URL Construct:

`http(s)://<EPMWARE_URL>/service/api/task/get_status/{TASK_ID}`

### Method: GET

### Example:

```
curl GET 'http://demo.EPMWAREcloud.com/service/api/task/get_status/244591'  
-H 'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb7'
```

### Response:

```
{"message": "Load completed  
successfully", "percentCompleted": "100", "status": "S", "taskId": "2445  
91"}
```

**Action:** Get Log File

This API will return task details (Logfile) for given task id.

**URL Construct:**

http(s) ://<EPMWARE\_URL>/service/api/task/get\_log\_file/{TASK\_ID}

**Method:** GET**Example:**

```
curl GET
'http://demo.EPMWAREcloud.com/service/api/task/get_log_file/244591' -H
'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb7' >
ew_task_244591_log.txt
```

**Response:**

The REST API will return log details for the given task ID as standard output on the console which can be redirected to be saved into a local file as shown in the example above.

25-NOV-2019 21:10:20 --> Task 244654 [ERP Import REST API] started

25-NOV-2019 21:10:20 --> Stage # 1 ASOALL\_account - file id : 138

25-NOV-2019 21:10:20 --> Load File and Execute ERP Import for Interface : ASOALL\_account

25-NOV-2019 21:10:20 --> Process file : ASOALL\_Account.csv

25-NOV-2019 21:10:20 --> Load File and Execute ERP Import for Interface : ASOALL\_account

25-NOV-2019 21:10:20 --> Column Separator ; ,

25-NOV-2019 21:10:20 --> Column Enclosed By : "

25-NOV-2019 21:10:20 --> Status : S - Message :

25-NOV-2019 21:10:20 --> Lines : 4

25-NOV-2019 21:10:20 --> Columns Count : 15

25-NOV-2019 21:10:21 --> Transfer Complete.

25-NOV-2019 21:10:21 --> Task status : Success

25-NOV-2019 21:10:21 --> Task 244654 Completed Successfully

## Module: ERP

### Action: Upload File for ERP Import

This API will upload a local metadata file for a given ERP Import interface and return Task ID which can be monitored for its status.

**URL Construct:** `http(s)://<EPMWARE_URL>/service/api/erp/upload_file`

**Method:** POST

REST API Parameters:

| Parameter Code | Value                         | Example          |
|----------------|-------------------------------|------------------|
| Name           | ERP Import configuration Name | ASOALL_Account   |
| delimiter_char | File Delimiter character      | , (comma)        |
| enclosing_char | Fields enclosed by character  | " (Double Quote) |

This call will return TASK\_ID in return. User will use this TASK\_ID to probe its status.

### API structure:

```
curl POST 'http://<EPMWARE_URL>/service/api/erp/upload_file' -H
'authorization: Token <TOKEN>' -H 'content-type: multipart/form-data' -F
'file=@"<FILE_WITH_PATH>' -F 'name="<ERP_IMPORT_NAME>"' -F
'delimiter_char=<COLUMN_DELIM_CHAR>' -F
'enclosing_char=<COLUMNS_ENCLOSED_BY_CHAR>'
```

### Example:

```
curl POST 'http://demo.EPMWAREcloud.com/service/api/erp/upload_file' -H
'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb7' -H 'content-
type: multipart/form-data' -F 'file=@"c:\test\ASOALL_Account.csv"' -F
'name="ASOALL_Account"' -F 'delimiter_char=","' -F 'enclosing_char="'
```

**Response:** It is in JSON format

```
{"message":"Task Submitted Succesfully.","status":"S","taskId":"244659"}
```

**Action:** Run ERP Import

This action allows the user to run ERP Import interface task on demand. ERP Import must be configured with “One Time Schedule” type in advance before calling this API. Also, ERP Import Service must be running otherwise this API will return Error.

**URL Construct:** `http(s)://<EPMWARE_URL>/service/api/erp/run`

**Method:** POST

**Parameters :**

| Parameter Code | Value                                              | Example        |
|----------------|----------------------------------------------------|----------------|
| name           | ERP Import configuration Name                      | ASOALL_Account |
| timeout_min    | # of minutes to wait for the execution to complete | 60 (Default)   |

This call will return TASK\_ID in return. User will use this TASK\_ID to probe its status.

**API structure:**

```
curl POST 'http://<EPMWARE_URL>/service/api/erp/run' -H 'authorization:
Token <TOKEN>' -H 'content-type: application/json' -d '{ "name" :
"<ERP_IMPORT_NAME>", "timeout_min" : <WAIT_MINUTES>}'
```

**Example**

```
curl POST 'http://demo.EPMWAREcloud.com/service/api/erp/run' -H
'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb7' -H
```

```
'content-type: application/json' -d '{"name" :  
"ASOALL_Account","timeout_min" : 60}'
```

**Response:**

It is in JSON format.

```
{"status":"S","taskId":"244659"}
```

## Module: Deployment

**Action:** Run Deployment

This action allows user to Deployment task on demand. Deployment must be configured with “One Time Schedule” type in advance before calling this API. Also, “Deployment” Service must be running otherwise this API will return Error.

**URL Construct:**

```
http(s)://<EPMWARE_URL>/service/api/deployment/run
```

**Method: POST****Parameters**

| Parameter Code | Value                                              | Example      |
|----------------|----------------------------------------------------|--------------|
| Name           | Deployment Name                                    | “ASOALL”     |
| timeout_min    | # of minutes to wait for the execution to complete | 60 (Default) |

This call will return TASK\_ID in return. User will use this TASK\_ID to probe its status.

**Example:**

```
curl POST
'http://demo.EPMWAREcloud.com/service/api/deployment/run' -H
'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb7' -H
'content-type: application/json' -d '{"name" :
"ASOALL","timeout_min": 60}'
```

**Response:**

It is in JSON format.

```
{"status": "S", "taskId": "244596"}
```

**Action: Get Deployment Execution Details**

This REST API will allow users to download Execution details for a given Deployment name.

**URL Construct:**

```
http(s)://<EPMWARE_URL>/service/api/deployment/get_execution_details
```

**Parameters**

| Parameter Code | Value              | Example  |
|----------------|--------------------|----------|
| Name           | Deployment Name    | "ASOALL" |
| Latest_only    | Y or N (Default Y) | Y        |

**Example:**

```
curl
http://localhost:8080/EPMWARE/service/api/deployment/get_execution
_details?name=ASOALL&latest_only=Y -H 'authorization: Token
25388ad5-c9af-4cf3-af47-8021c1ab3fb6'
```

## Module: Export

### Action: Run Export

This action allows user to run Export task on demand. **Export Profile/ Export Book** specified in the REST API must be configured in advance before calling this API.

### URL Construct:

`http(s)://<EPMWARE_URL>/service/api/export/run`

### Method: POST

### Parameters:

If **Export profile** needs to be executed use below parameter:

| Parameter Code | Value               | Example |
|----------------|---------------------|---------|
| name           | Export Profile Name | PBCS    |

If **Export Book** needs to be executed use below parameter:

| Parameter Code | Value               | Example   |
|----------------|---------------------|-----------|
| book_name      | Export Profile Name | PBCS_BOOK |

This call will return TASK\_ID in return. User will use this TASK\_ID to probe its status.

### Example:

To Execute export profile -

```
curl POST 'http://demo.EPMWAREcloud.com/service/api/export/run' -H
'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb7' -H
'content-type: application/json' -d '{"name" : "PBCS"}'
```

**To execute Export Book**

```
curl POST 'http://demo.EPMWAREcloud.com/service/api/export/run' -H  
'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb7' -H  
'content-type: application/json' -d '{"book_name" : "PBCS_BOOK"}'
```

**Response:**

It is in JSON format.

```
{"status":"S","taskId":"244596"}
```

**Action: Download Export File**

This action allows user to download the Export Data File after the export execution has been successfully completed.

**URL Construct:**

```
http(s)://<EPMWARE_URL>/service/api/export/download_file/<Task ID>
```

**Method: GET****Example:**

```
curl GET  
'http://demo.EPMWAREcloud.com/service/api/export/download_file/123  
4' -H 'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb7'
```

**Response:**

It downloads a file in the directory where the API is being executed

**Action: Get Export Execution Details**

This REST API will allow users to download Execution details for a given export name.

**URL Construct:**

http(s)://<EPMWARE\_URL>/service/api/export/get\_execution\_details

**Method: GET****Parameters**

| Parameter Code | Value              | Example |
|----------------|--------------------|---------|
| Name           | Name of Export     | "PBCS"  |
| Latest_only    | Y or N (Default Y) | Y       |

**Example:**

```
curl GET
'http://localhost:8080/EPMWARE/service/api/export/get_execution_details?name=PBCS&latest_only=Y' -H 'authorization: Token 25388ad5-c9af-4cf3-af47-8021c1ab3fb6'
```

**Additional example to call REST API:**

Windows Powershell Script Example

```
#####
#####
# EPMWware Inc.
# Author: Deven Shah
# Date: 11-Sep-2021
# Purpose: REST API to get Export execution details.
#####
#####

Param(
    [Parameter(Mandatory=$True)]
```

```

        [string]$url,
        [Parameter (Mandatory=$True)]
        [string]$token,
        [Parameter (Mandatory=$True)]
        [string]$exportProfileName,
        [Parameter (Mandatory=$True)]
        [string]$latest_only
    )

$module = 'export'
$headers = @{"authorization"="Token $($token)"}
function log()
{
    Write-Host $args[0]
}
log "=====
log "Parameters"
log "=====
log "EPMWARE REST API URL   : $url"
log "User Token             : *****"
log "Export Profile Name    : $exportProfileName"
log "Latest Execution only  : $latest_only"
log "=====

#####
#####

# Get Export Execution Details

#####
#####

```

```
#
http://EPMWARE1.EPMWARE.com:8080/EPMWARE/service/api/export/get_execution_details?name=PBCS&latest_only=N

try {

    $taskURL="$url/$module/get_execution_details?name=$exportProfileName&latest_only=$latest_only"

    $restResponse = Invoke-RestMethod -Uri $taskURL -Method Get -Headers $headers

    foreach ($record in $restResponse.vEWVExportExecutions)
    {
        $exportName=$record.exportName
        $execId=$record.id
        $startDate=$record.startDate
        $endDate=$record.endDate
        $outputFileName=$record.outputFileName
        $outputFileSize=$record.outputFileSize
        $statusMeaning=$record.statusMeaning
        $message=$record.message

        log "Execution ID:$execId Status:$statusMeaning
StartDate:$startDate EndDate:$endDate
OutputFileName:$outputFileName OutputFileSize:$outputFileSize
Message:$message"
    }

} catch {

    log "EW_ERROR: EPMWARE REST API to get Export Execution
Details failed. $details"
```

```
        $exceptionDetails = $_.Exception
        log $exceptionDetails
        exit
    }
}
```

```
log "====="
```

```
log "Script completed"
```

Call this Powershell script ( example ew\_exp\_get\_details.ps1) file with parameters

```
powershell %CD%\ew_exp_get_details.ps1 -url "%URL%" -token %TOKEN%  
-exportProfileName "PBCS" -latest_only "Y"
```

Replace URL and TOKEN parameters.

## Module: Security

**Action:** Create SSO User

This API allows the creation of new SSO users.

**URL Construct:**

http(s)://<EPMWARE\_URL>/service/api/security/users/sso/create

**Method :** POST

**Parameters**

| Parameter Code | Value                   | *Required / Optional field |
|----------------|-------------------------|----------------------------|
| userName       | Name of the user        | Required                   |
| firstName      | First Name of the user  | Required                   |
| lastName       | Last name of the user   | Optional                   |
| email          | Email address           | Required                   |
| description    | Description of the user | Optional                   |

**Request Example:**

```
curl POST
'http://demo.EPMWAREcloud.com/service/api/security/users/sso/create' -H 'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb7'
-H 'content-type: application/json' -d '{"userName" :
"SSO_USER1","firstName" : "SSO","lastName" : "USER1","email" :
"SSO_USER1@example.com","description" : "New user creation" }'
```

**Response Example :**

It is in JSON format.

```
{ "id": "586",
  "createDate": "2025-06-06T13:13:54Z",
  "updateDate": "2025-06-06T13:13:54Z",
  "createdBy": "330",
  "updatedBy": "330",
  "userName": "SSO_USER1",
  "firstName": "SSO",
  "lastName": "USER1",
  "email": "SSO_USER1@example.com",
  "description": "New user creation",
  "enable": "Y",
  "samlEnabled": "false"
}
```

**Response Codes:**

- 200 Created: User successfully created.

401 Unauthorized: Missing or invalid authentication.

**Action:** Update SSO User

This action allows updating details of an existing SSO user.

**URL Construct:**

`http(s)://<EPMWARE_URL>/service/api/security/users/sso/update/<UserName>`

**Method :** PUT**Parameters :**

Include parameters which needs to be updated.

| Parameter Code | Value                   | *Required / Optional field |
|----------------|-------------------------|----------------------------|
| userName       | Name of the user        | Optional                   |
| firstName      | First Name of the user  | Optional                   |
| lastName       | Last name of the user   | Optional                   |
| email          | Email address           | Optional                   |
| description    | Description of the user | Optional                   |

**Request Example:**

To Update details of SSO user , username : "SSO\_USER1"

```
curl --request PUT
'http://demo.EPMWAREcloud.com/service/api/security/users/sso/
update/SSO_USER1' -H 'authorization: Token 15388ad5-c9af-4cf3-
af47-8021c1ab3fb7' -H 'content-type: application/json' -d
'{"userName" : "SSO_USER1","firstName" : "SSOUPD","LastName" :
```

```
"USER1UPD", "email" :  
"SSO_USER1_UPDATED@example.com", "description" : "Updating Details"  
}'
```

**Response:** In JSON format

```
{ "id": "586",  
  "createDate": "2025-06-06T13:13:54Z",  
  "updateDate": "2025-06-06T14:10:26Z",  
  "createdBy": "330",  
  "updatedBy": "330",  
  "userName": "SSO_USER1",  
  "firstName": "SSOUPD",  
  "lastName": "USER1",  
  "email": "SSO_USER1_UPDATED@example.com",  
  "description": "Updating Details",  
  "enable": "Y",  
  "samlEnabled": "false" }
```

**Response Codes:**

- 200 Updated: User updated successfully.
- 401 Unauthorized: Missing or invalid authentication.

**Action:** Disable SSO User

This API is used to deactivate or disable existing SSO user account.

**URL Construct:**

http(s)://<EPMWARE\_URL>/service/api/security/users/sso/disable/<UserName>

**Method :** PATCH

**Parameters :** Not required.

**Request Example:**

Disable SSO user : "SSO\_USER1"

```
curl -request PATCH
'http://demo.EPMWAREcloud.com/service/api/security/users/sso/disable/SSO_USER1' -H 'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb7'
```

**Response:** In JSON format

```
{ "value": "User status disabled successfully." }
```

**Response Codes:**

- 200 Updated : User status disabled successfully.
- 401 Unauthorized: Missing or invalid authentication.

**Action:** Enable SSO User

This API is used to activate or enable existing SSO user account.

**URL Construct:**

http(s)://<EPMWARE\_URL>/service/api/security/users/sso/enable/<UserName>

**Method : PATCH**

**Parameters :** Not required.

**Request Example:**

Enable SSO user , username : "SSO\_USER1"

```
curl --request PATCH
'https://testnewplatform.epmwarecloud.com/service/api/security/users/sso/enable/SSO_USER1' -H 'Authorization: Token 26435747-0f89-48dc-8283-db62908fc7d9'
```

**Response:** In JSON format

```
{ "value": "User status enabled successfully." }
```

**Response Codes:**

- 200 Updated : User enabled successfully.
- 401 Unauthorized: Missing or invalid authentication.

**Action:** Add SSO User to Security Groups

This action allows to add SSO users to one or more security groups.

**URL Construct:**

```
http(s)://<EPMWARE_URL>/service/api/security/users/sso/assign_groups
```

**Method : POST**

**Parameters :**

| Parameter Code | Value                                           | *Required / Optional field |
|----------------|-------------------------------------------------|----------------------------|
| userName       | Name of the SSO user                            | Required                   |
| groupNames     | List of Security groups to assigned to the user | Required                   |

**Request Example:**

Add SSO user "SSO\_USER1" to Security groups: Approver, Reviewer, Admin

```
curl POST
'http://demo.EPMWAREcloud.com/service/api/security/users/sso/assign_groups' -H 'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb7' -H 'content-type: application/json' -d '{"userName": "SSO_USER1", "groupNames" : ["Approver" , "Reviewer" , "Admins" ] }'
```

**Response:** In JSON format

```
{ "value": "User successfully assigned to the specified groups." }
```

**Response Codes:**

- 200 Updated: User successfully assigned to the specified groups.
- 401 Unauthorized: Missing or invalid authentication.

**Action:** Remove SSO User from Security Groups

This action allows to remove SSO users from one or more security groups.

**URL Construct:**

```
http(s)://<EPMWARE_URL>/service/api/security/users/sso/unassign_groups
```

**Method :** POST

**Parameters :**

| Parameter Code | Value                                                | *Required / Optional field |
|----------------|------------------------------------------------------|----------------------------|
| userName       | Name of the SSO user                                 | Required                   |
| groupNames     | List of security groups to be removed from the user. | Required                   |

**Request Example:**

Remove SSO user "SSO\_USER1" from security groups : Approver, Reviewer

```
curl POST
'http://demo.EPMWAREcloud.com/service/api/security/users/sso/unassign_groups' -H 'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb7' -H 'content-type: application/json' -d '{"userName": "SSO_USER1", "groupNames" : ["Approver" , "Reviewer" ] }'
```

**Response:** In JSON format

```
{
  "value": "User successfully removed from the specified groups."
}
```

**Response Codes:**

- 200 Updated: User successfully removed from the specified groups..
- 401 Unauthorized: Missing or invalid authentication.

**Action:** Get Users Details

This action allows to fetch all the details of SAML or NATIVE users from the application.

**URL Construct:**

To get the details of all the users:

`http(s)://<EPMWARE_URL>/service/api/security/users`

To get the details of only Native users:

`http(s)://<EPMWARE_URL>/service/api/security/users?userType=NATIVE`

To get the details of SAML users:

`http(s)://<EPMWARE_URL>/service/api/security/users?userType=SAML`

**Method:** GET

**Query Parameters:**

| Parameter Code | Value                                                              | *Required / Optional field |
|----------------|--------------------------------------------------------------------|----------------------------|
| userType       | Filter user by types. Values :<br>SAML , NATIVE                    | Optional                   |
| pageSize       | Number of users to return per<br>page. Default value set is 10000. | Optional                   |

**Request Example:**

```
curl GET
'https://demo.epmwarecloud.com/service/api/security/users' -H
'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb9'
```

**Response:** In JSON format

```
[{"userName":"ramesh","firstName":"Ramesh","lastName":"Tendulkar","email":"ram@test.com","enable":"Y","userType":"NATIVE",description:""} ,
{"userName":"sachin","firstName":"Sachin","lastName":"Tendulkar","email":"sachin@test.com","enable":"Y","userType":"SAML",description:""} ]
```

**Response Codes:**

- 200 : Success
- 401 Unauthorized: Missing or invalid authentication.

**Action:** Get User Groups

This action allows to fetch the list of security groups assigned to the given user.

**URL Construct:**

```
http(s)://<EPMWARE_URL>/service/api/security/users/groups?userName=<user_name>
```

**Method:** GET**Query Parameters :**

| Parameter Code | Value                                              | *Required / Optional field |
|----------------|----------------------------------------------------|----------------------------|
| userName       | Provide username to fetch assigned security groups | Required                   |

**Request Example:**

```
curl GET  
'https://demo.epmwarecloud.com/service/api/security/users/groups?u  
serName=AKSHAY' -H 'authorization: Token 15388ad5-c9af-4cf3-af47-  
8021c1ab3fb9'
```

**Response:** In JSON format

```
{ "userName": "AKSHAY", "groups" : [ "Developers", "Reviewers" ] }
```

**Response Codes:**

- 200 : Success

- 401 Unauthorized: Missing or invalid authentication.

**Action:** Get Groups Details

This action allows to fetch the details of all the security groups present in the application.

**URL Construct:**

`http(s)://<EPMWARE_URL>/service/api/security/groups`

**Method:** GET

**Query Parameters:** No Parameter required.

**Request Example:**

```
curl GET
'https://demo.epmwarecloud.com/service/api/security/groups' -H
'authorization: Token 15388ad5-c9af-4cf3-af47-8021c1ab3fb9'
```

**Response:** In JSON format

```
[{"name":"Developers", description:"Developers groups", "enable":"N"},
{"name":"Reviewers", description:"Reviewers groups", "enable":"N"}]
```

**Response Codes:**

- 200 : Success
- 401 Unauthorized: Missing or invalid authentication.

**Action:** Get Group Users

This action retrieves the list of users assigned to the security groups.

**URL Construct:**

Use URL below to retrieve all security groups along with their assigned list of users.

```
http(s)://<EPMWARE_URL>/service/api/security/groups/users
```

Use URL below to retrieve list of users assigned to passed security group.

```
http(s)://<EPMWARE_URL>/service/api/security/groups/users?groupNam  
e=<group_name>
```

**Method : GET****Query Parameters :**

| Parameter Code | Value                                                    | *Required / Optional field |
|----------------|----------------------------------------------------------|----------------------------|
| groupName      | Provide groupname to fetch list of users assigned to it. | Optional                   |

**Request Example:**

Get list of users assigned to Approver group.

```
curl GET  
'https://demo.epmwarecloud.com/service/api/security/groups/users?g  
roupName=APPROVER' -H 'authorization: Token 15388ad5-c9af-4cf3-  
af47-8021c1ab3fb9'
```

**Response:** In JSON format

```
[{"groupName": "Approver", "users": [ "TONY" , "DEVEN" ]}]
```

**Response Codes:**

- 200 : Success
- 401 Unauthorized: Missing or invalid authentication.