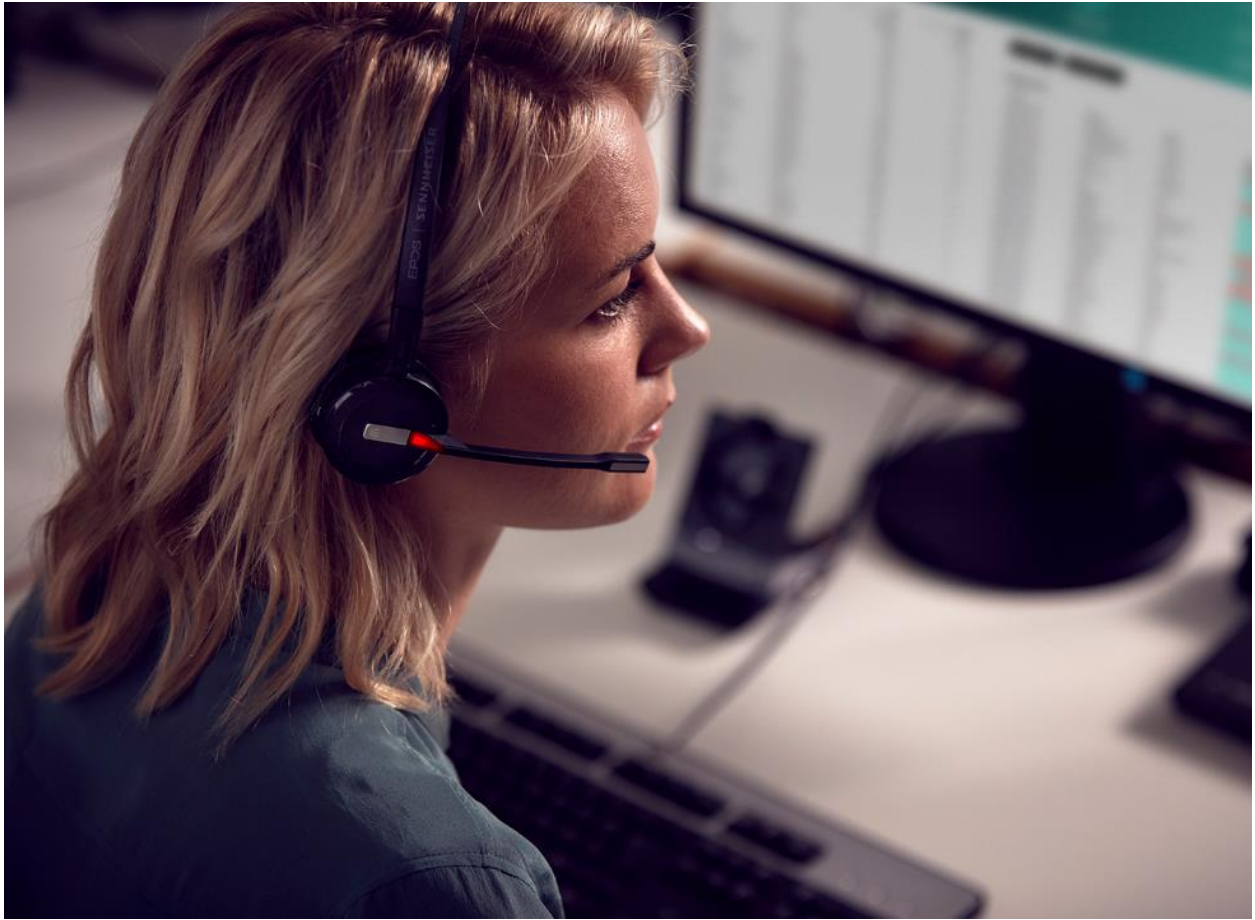


EPOS



EPOS API (On-premise) 1.0

Table of Contents

1.	Introduction	3
1.	Ucmp swagger.....	5
2.	How to enable your API.....	5
2.1	Request Token	6
2.2	Tenant Registration.....	7
2.3	Update Partner Key	8
2.4	Change Default Partner.....	8
2.5	Reset Password	9
3.	Tenant Management API.....	11
3.1	Get Tenants	11
4.	Asset Management API	11
4.1	GetDevices.....	11
5.	User Management API	11
5.1	GetUsers.....	11
5.2	GetManualGroups.....	12
5.3	GetAdGroups	12
5.4	Create Manual UserGroup	12
5.5	Create Active Directory UserGroup	13
5.6	Delete UserGroup.....	13
6.	Update Management API	13
6.1	Deployment Type	14
6.2	Deployment Status	15
6.3	Get FirmwareList	15
6.4	Deploy UpdateToGroup.....	15
6.5	Deploy UpdateToUser.....	16
6.6	DeleteUpdate	16
6.7	CancelJob	16
7.	Configuration Management.....	17
7.1	DeviceType List.....	17
7.2	DeviceConfiguration Custom List.....	18
7.3	DeviceConfiguration Default List	18

7.4	Duplicate configuration package.....	18
7.5	Edit configuration package	18
7.6	Package Details.....	19
7.7	ConfigurationOverview.....	19
6.9	Deploy Configuration to User	20
6.10	DeleteConfiguration	20
6.11	Cancel Configuration JOB.....	20
7	Data insights API	20
7.1	DeviceAdoptionReport	20
7.2	UpdateStatusOverview Report	20
7.3	FirmwareOverview Report.....	21
7.4	CallActivity Data	21
7.5	Dashboard data	21
7.5.1	ActiveDevices.....	22
7.5.2	FirmwareCompliance.....	22
7.5.3	SoftwareCompliance.....	22
7.5.4	CallActivity	22
8	Troubleshooting.....	23
8.1	Response Types.....	23
8.2	Check for token generation	23
8.3	Failed login.....	24

1. Introduction

The document provides an overview of EPOS API and how to enable EPOS API for your solution integration.

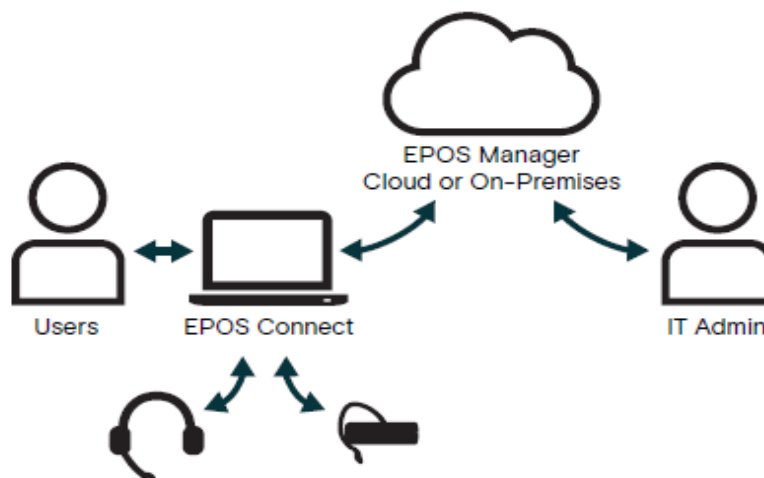
The target audience of the document is EPOS integration partners, such as Unified Communication Management Platform (UCMP) providers, UC performance monitoring solution providers, etc. The document also targets to support any API consumer like IT admins who plans to integrate EPOS device data and device management into your existing IT solutions.

EPOS Manager

EPOS Manager is full-featured asset management software for audio endpoints. The software offers device configuration, asset overview and tracking, reporting plus device firmware management.

EPOS Connect

End-user PC client, providing update/configuration service for connected devices.



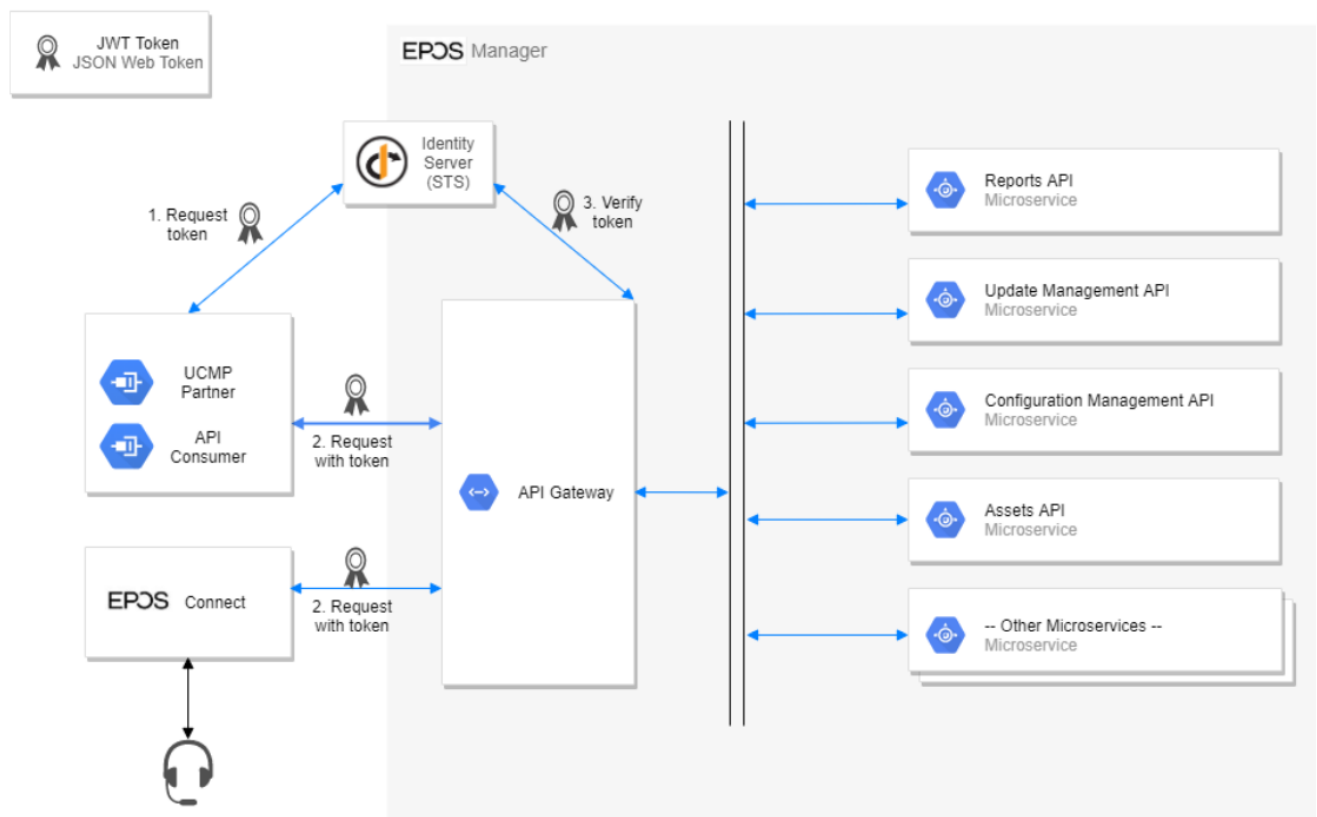
Deploy EPOS Connect

EPOS Connect is the mandatory to install in order to enable data from devices. IT admin can use IT tools like SCCM to deploy EPOS Connect.

Below are the requirements for installing EPOS Connect:

- Hardware requirement
 - To run the EPOS Connect application, your computer must meet the following minimum requirements: – Processor: Minimum 1.6 GHz or faster processor – RAM: 2 GB RAM – Hard disk space: 500 MB 2.2
- Operating system requirements
 - Windows operating systems Win 7 and newer are supported.

EPOS API Overview



EPOS API provides data insights including asset data and reports data to provide an overview of audio device information, headset usage and call data in the organization, and provides APIs for device firmware and configuration management.

- Asset data API
- Report data API
- Update management API
- Configuration management API

All EPOS APIs are REST APIs with JSON response and accessed over HTTPS.

The diagram illustrates the UCMP solution from a high-level perspective. API consumers (clients), i.e. UCMP Partners, EPOS Connect, or any other API Consumer, will connect through an API Gateway. In order to issue requests to our API, any client need to request a security token by providing client credentials which is issued to register and accepted clients. The security token (JSON Web Token, JWT) is then used to get access to our API. For every request, the JWT token is verified before routing any requests to our backend microservices.

- On creation of a UCMP partner in the system, the partner is provided with a unique id and password (key).
- Partner should make a request to the connect API of identity server using the id and key.
- On successful authentication and authorization, a JSON Web Token (JWT) is provided.
 - JWT has a session timeout period after which token has to re-request.
- JWT token has to be sent in headers for further API calls without which request will fail.
- All API requests must be made over HTTPS.

Prerequisites

Refer to EPOS Manager Onpremise pdf document for Docker installation procedure.

Note:

- Refer to docker-compose file for OnPremise Client Id (Administrator__ClientId under **epos_manager_identity** section) and .env file in the downloaded EPOS Manager Package for OnPremiseClientKey.
- A Default partner is created as a part of docker-compose file execution. Refer to below details:

Partner Name: DefaultPartner

PartnerId/Client Id: 9E8F4402-943A-4D92-8DE2-D29615E68D0D

Partner Key: Refer to section 4 “Getting Started” in OnPremise pdf document for the partner key.

1. Ucmp swagger

Refer below URL for more details on API like input parameters, request body and response body.

<http://localhost:8082/swagger/index.html>

2. How to enable your API

Below are the steps for enabling API:

Step 1: Using the partner Id and key shared make a request to connect API for authentication.

Step 2: On successful authentication a JWT token is shared.

Step 3: JWT token should be added to the request headers for further API calls.

Step 4: Refer to section 2.4 for changing the default partner.

2.1 Request Token

For every request to the API, a token need to be provided for authentication. Use your partner id and secret to obtain a token. Send a request to the token endpoint, attach the token to any API request by adding a bearer token.

Sample Request URL:

`http://localhost:25801/connect/token` or `http://<IP or machine name>:25801/connect/token`

Refer <http://localhost:25801/swagger/index.html> for more details

Sample code (C#):

```
...  
  
class Example  
{  
    private const string PartnerId = "<PartnerId>";  
  
    //This is a simple example. A proper implementation would protect the  
    //credentials by not exposing in source code.  
    private const string PartnerSecret = "<PartnerSecret>";  
  
    private const string IdentityEndPoint =  
        "http://localhost:8082/connect/token";  
  
    static void EntryMethod()  
    {  
        // This should return an access token  
        var tokenResponse = GetToken().GetAwaiter().GetResult();  
  
        // Call API with access token  
        var jsonResult = Get(tokenResponse);  
    }  
  
    private static async Task<TokenResponse> GetToken()  
    {  
        var protocolClient = new HttpClient();  
  
        var tokenResponse =  
            await protocolClient.RequestClientCredentialsTokenAsync(  
                new ClientCredentialsTokenRequest  
                {  
                    Address = IdentityEndPoint,  
                    ClientId = PartnerId,  
                    ClientSecret = PartnerSecret  
                }  
            );  
    }  
}
```

```

        return tokenResponse;
    }

    private static async Task<string> Get(TokenResponse tokenResponse)
    {
        var apiClient = new HttpClient();

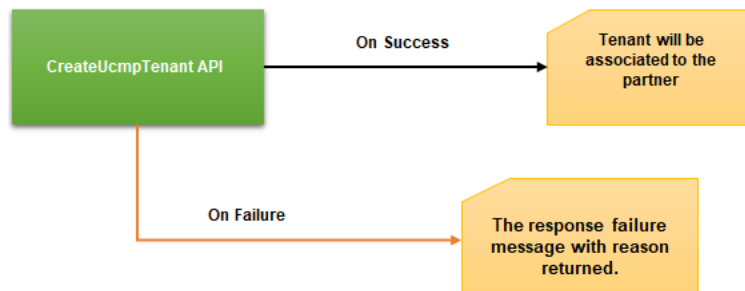
        // Add bearer token
        apiClient.SetBearerToken(tokenResponse.AccessToken);

        var result = await apiClient.GetStringAsync("<ApiEndPoint>");

        return result;
    }
}

```

2.2 Tenant Registration



The purpose of this API is to create a tenant. Once the tenant is successfully created and auto-approved, the same tenant will be associated to the respective partner.

Make a request to the API mentioned below. Send the JWT token in the headers of the request.

Refer to section [Ucmp swagger](#)

Sample Request URL:

<http://localhost:8082/api/v1/Tenants/CreateUcmpTenant>

On successful creation of the tenant, the response will have the below information:

- Tenant Id – This is unique identification in the system for the tenant created. This is required for accessing the data of the tenant through other APIs
- Tenantname – as specified in the request
- Companyname – as specified in the request

For more details, refer swagger.

Existing EPOS Manager customer associating to a partner solution

If the existing EPOS Manager customer wants to associate to a partner solution, they can go to tenant settings page of EPOS Manager UI and select the relevant partner.

A Partner can use the below end point to get the associated tenants.

Sample Request URL:

<https://ucmp.enterprise.eposaudio.com/api/v1/Admin/getTenantsForPartner?partnerId=D7763197-2651-401a-a0ab-d1ec2be50059>

2.3 Update Partner Key

An API to change or update the partner key. It accepts PartnerId and Key as input parameters. On successful execution, generates and return new partner key.

Sample Request URL:

<http://localhost:8082/api/v1/Partner/UpdatePartnerKey>

2.4 Change Default Partner

To change the details of default partner created at the time of docker-compose file execution.

As a part of docker-compose file execution, a default partner will be created automatically. It is recommended to change your default partner details (one time activity)

Refer to below Powershell Script or it can also be found in the downloaded EPOS Manager package.



The script will ask for the below input parameters:

Identityendpoint: Open docker dashboard and check for the epos_manager_identity port number.

For example: If the port number is 25801, then the identity endpoint will be <http://localhost:25801>

ClientID: Current ClientID. If it is the default one, refer to section 4 “Getting Started” for details.

ClientSecret: Current ClientSecret. If it is the default one, refer to section 4 “Getting Started” for details.

ParnerName: Tenant Admin to enter the new Partner name which you wish to put.

Ucmpendpoint: Open docker dashboard and check for the epos_manager_api port number.

For example: If the port number is 8082, then the identity endpoint will be http://localhost:8082

The output of the script will be the updated Partner name, ClientID, ClientSecret. Save these details, as it is required to reset the password for Tenant Admin.

Refer to sample powershell execution screenshot

```
cmdlet ChangeDefaultPartner.ps1 at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
Identityendpoint : http://localhost:25801
ClientID : 78D75782-11DD-42B7-AEAC-F7E60D7DE7DC
ClientSecret : lodkdieekp111dd
PartnerName : TestPartner
Ucmpendpoint : http://localhost:8082
```

```
Windows PowerShell
Updated ClientID : = 836f781c-2f67-4036-8e80-df321a416f71
Updated ClientSecret./ : = 7zJYx5Ivrviz90En4eNTG!lp(QpTDpPNhtVKpJgw
Updated PartnerName : = localtest
-----
Press Enter key to Exit:
```

2.5 Reset Password

An API to generate the reset password URL. It takes user's email id as an input parameter.

Refer to section [Ucmp swagger](#)

Sample Request URL:

http://localhost:8082/api/v1/Partner/getResetPasswordURL?email={{emailId}}

Note: URL expires in 96 hours.

In case the login password has been forgotten, Tenant Admin can use the powershell script below embedded or can also be found in the downloaded EPOS Manager package.

Powershell Script:



The script will ask for the below input parameters:

Identityendpoint: Open docker dashboard and check for the epos_manager_identity port number.

For example: If the port number is 25801, then the identity endpoint will be `http://localhost:25801`

ClientID: Current ClientID. If it is the default one, refer to section 4 “Getting Started” for details.

ClientSecret: Current ClientSecret. If it is the default one, refer to section 4 “Getting Started” for details.

Ucmppoint: : Open docker dashboard and check for the epos_manager_api port number.

For example: If the port number is 8082, then the identity endpoint will be `http://localhost:8082`

Email Id: Email address of the user for whom the password reset is required.

The output of the script will be the Reset Password URL. Tenant Admin can copy this link to the browser and reset the password

Refer to sample powershell execution screenshot

```
cmdlet GetResetPasswordURL.ps1 at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
Identityendpoint : http://localhost:25801
ClientID : 78D75782-11DD-42B7-AEAC-F7E60D7DE7DC
ClientSecret : lodkdieekp111dd
Ucmppoint : http://localhost:8082
```

Administrator: Windows PowerShell

```
ResetPassword URL : http://localhost:80/#/resetpassword?token=db345ecf-fc78-4867-9d9e-7cfb added75588
Message : valid email
IsSuccess : True
-----
Press Enter key to Exit:
```

3. Tenant Management API

3.1 Get Tenants

To fetch Tenants which are associated with a Partner “GetTenants” API to be invoked. As a prerequisite, Partner must be successfully logged in.

Refer to section [Ucmp swagger](#)

Sample Request URL:

`http://localhost:8082/api/v1/tenants/ getTenants?Take=30&Skip=0&SortColumn&SortDescending`

4. Asset Management API

4.1 GetDevices

This API is used to get devices list for a tenant.

Refer to section [Ucmp swagger](#)

Sample Request URL:

`http://localhost:8082/api/v1/Assets/getDevices?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A&Take=10&Skip=0&SortColumn&SortDescending`

The response has the below information:

- Id – Unique ID
- deviceId – uniquely identifies a device
- vendor – currently supported vendors like EPOS/Jabra/Plantronics
- name – name of the device
- firstseen – The time the device first contacted with the server
- lastSeen- The time the device last contacted the server
- firstContactFwVersion- The device version when it first contacted EPOS connect
- currentContactFwVersion- The device FW version when it last contacted EPOS connect
- firstContactIpAddress- The IP address of the machine from where the device first contacted
- lastContactIpAddress- The IP address of the machine from where the device last contacted
- currentUsername- The Firstname+Lastname of the user currently using the device.
- deviceStatus- the status of the device i.e. Active/Inactive based on the time when it last contacted

5. User Management API

User API can provide the registered user information.

5.1 GetUsers

This API request retrieves all the users of a tenant.

Refer to section [Ucmp swagger](#)

Sample Request URL:

`http://localhost:8082/api/v1/user/getUsers?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A&Take=10&Skip=0&SortColumn&SortDescending`

Response has the below information:

- `userName`- Firstname + lastname of the user
- `softwareVersion`: the EPOS connect version the user has installed on his/her machine
- `deviceCount`- the no. of devices the user is using
- `userId`- uniquely identifies the user in the system
- `email`- the email address of the user

5.2 GetManualGroups

This API request retrieves Manual Groups created for a tenant. It includes subgroups also.

Refer to section [Ucmp swagger](#)

Sample Request URL:

`http://localhost:8082/api/v1/User/getManualGroups?TenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A`

5.3 GetAdGroups

This API request retrieves AD Group List for a tenant.

Refer to section [Ucmp swagger](#)

Sample Request URL:

`http://localhost:8082/api/v1/User/getAdGroups?TenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A`

The response contains information such as groupname, groupid, the users in the group and subgroups (if any).

5.4 Create Manual UserGroup

This API is used to create a new manual user group for a tenant.

Refer to section [Ucmp swagger](#)

Sample Request URL

`http://localhost:8082/api/v1/User/createManualUserGroup`

5.5 Create Active Directory UserGroup

This API is used to create new active directory user group for a tenant.

Refer to section [Ucmp swagger](#)

Sample Request URL

`http://localhost:8082/api/v1/User/createActiveDirectoryUserGroup`

5.6 Delete UserGroup

This API is used to delete user group.

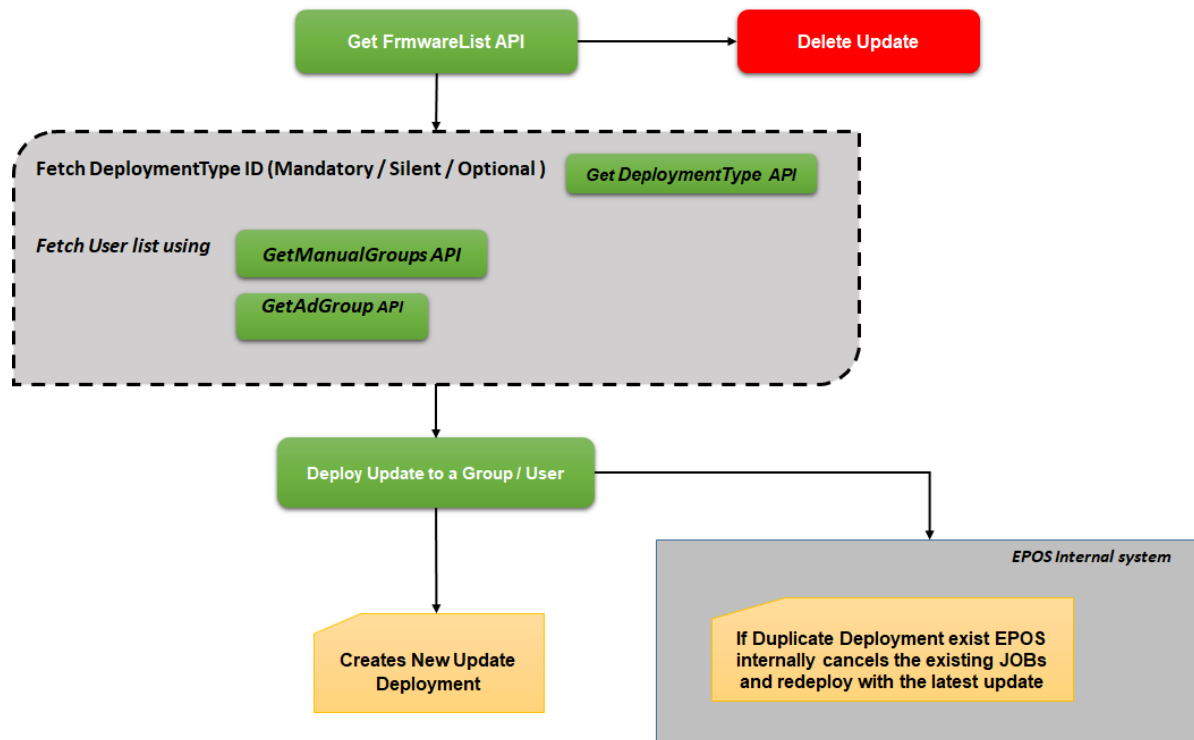
Refer to section [Ucmp swagger](#)

Sample Request URL

`http://localhost:8082/api/v1/User/deleteUserGroup`

6. Update Management API

Manages all the update related functionalities like getting firmware updates, deployment types and deploying updates to user groups or single user and getting deployment status.



Below are steps of APIs to be called for deploying update to a group:

Step1: Call the getDeploymentType API to get the available deployment types

Step2: Call the getManualGroups or getAdGroups API to get the groups in the tenant

Step3: Call the getFirmwareList to get the firmware updates available

Step4: Finally call the deployUpdateToGroup with all the required information to deploy the update to a group

6.1 Deployment Type

This API is used to get the list of deployment options that is used while deploying updates.

- Mandatory
- Optional
- Silent

Refer section [Ucmp swagger](#) for more details

Below fields are part of response:

- Id – this uniquely identifies the deployment type. This value has to be passed as input for deployUpdateToGroup API.
- Name : is the name of deployment type

6.2 Deployment Status

This API is used to get the Deployment Status of the updates that are deployed to the users.

Refer to section [Ucmp swagger](#)

Sample Request URL

`http://localhost:8082/api/v1/Update/getDeploymentStatus?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A&Take=10`

The response has some of the below key data:

- JobId - this ID uniquely identifies an update deployment that is done using deployUpdateToGroup API
- deviceName - the name of the product for which update deployment was done
- type - deployment status is of type “update” or “configuration”.
- updateStatus
 - Failed: the no of users on whose machines the deployment failed
 - Pending: the no of users to whom the update has not reached
 - Success: the no of users for whom the update was successful

6.3 Get FirmwareList

API is used to get the list of all Firmware available for respective tenants which can be used for deploying update to groups/user.

Refer to section [Ucmp swagger](#)

Sample Request URL

`http://localhost:8082/api/v1/Update/getFirmwareList?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A&showLatest=true&Take=10&Skip=0&SortColumn=Build&SortDescending=false`

6.4 Deploy UpdateToGroup

API allows to deploy updates to user groups.

Note:

1. For Mandatory/Optional deploy (DeploymentTypeId 6/7): ScheduleDateTime is mandatory and it should be in 'MM/dd/yyyy HH:mm:ss' format. ScheduleEndTime, UpdateStartTime and UpdateEndTime are not required.
2. For **Silent** deploy (DeploymentTypeId 8): ScheduleDateTime is mandatory and ScheduleEndTime is optional. Both should be 'MM/dd/yyyy' format. UpdateStartTime

- and UpdateEndTime are optional and it should be in 'HH:mm:ss' format. Default values for UpdateStartTime and UpdateEndTime are '00:00:00' and '23:59:59' respectively.
3. UpdateMessage is optional for Mandatory deploy (Not required for Optional/Silent deploy). If it is null or empty then default message will be considered.

Refer to section [Ucmp swagger](#)

Sample Request URL

<http://localhost:8082/api/v1/update/deployUpdateToGroup>

6.5 Deploy UpdateToUser

API is used to deploy firmware updates to single user.

Note:

1. For Mandatory/Optional deploy (DeploymentTypeId 6/7): ScheduleDateTime is mandatory and it should be in 'MM/dd/yyyy HH:mm:ss' format. ScheduleEndTime, UpdateStartTime and UpdateEndTime are not required.
2. For **Silent** deploy (DeploymentTypeId 8): ScheduleDateTime is mandatory and ScheduleEndTime is optional. Both should be 'MM/dd/yyyy' format. UpdateStartTime and UpdateEndTime are optional and it should be in 'HH:mm:ss' format. Default values for UpdateStartTime and UpdateEndTime are '00:00:00' and '23:59:59' respectively.
3. UpdateMessage is optional for Mandatory deploy (Not required for Optional/Silent deploy). If it is null or empty then default message will be considered.

Refer to section [Ucmp swagger](#)

Sample Request URL

<http://localhost:8082/api/v1/update/DeployUpdateToUser>

6.6 DeleteUpdate

API allows to delete tenant's Firmware Update or Configuration.

Refer to section [Ucmp swagger](#)

Sample Request URL

<http://localhost:8082/api/v1/update/deleteUpdate?TenantID=3fa85f64-5717-4562-b3fc-2c963f66afa6&updateId=1121>

6.7 CancelJob

API is used to cancel the update (Firmware/Configuration) deployed to respective Tenant.

Refer to section [Ucmp swagger](#)

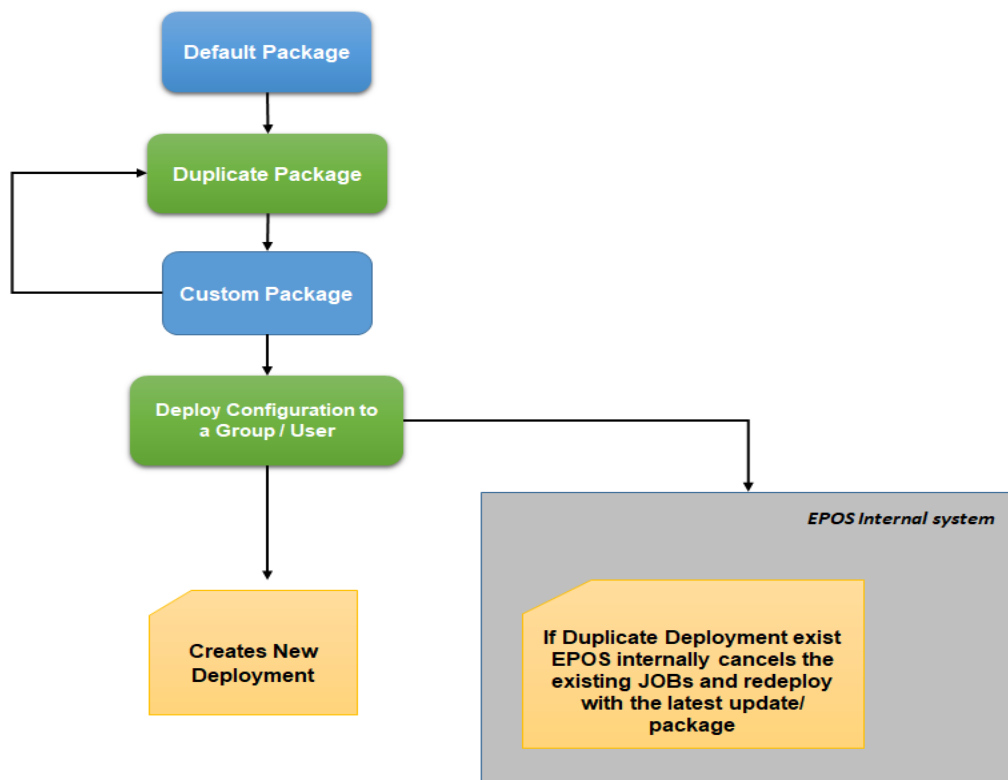
Sample Request URL

http://localhost:8082/api/v1/update/CancelJob?TenantID=76994A92-2E4C-4791-BABC-D1F7879AB23A&schedulerId=1211

7. Configuration Management

Configuration management APIs provides the mechanism to handle all the configuration related scenarios. It allows the end user to perform all the operations like getting configuration packages list, deploying configuration to particular user group etc.

Pictorial sequence to integrate configuration API



7.1 DeviceType List

This API is used to get the list of device and software type (Wireless DECT, Wireless BT, Wired USB, Speakerphone, EPOS Connect).

Refer to section [Ucmp swagger](#)

7.2 DeviceConfiguration Custom List

This API is used to get the customized configuration packages by IT managers in your organization. Fetches list of all raw device packages received and approved.

Result can be filtered using the ShowLatest as TRUE or FALSE (ShowLatest as FALSE will return all the records)

Refer to section [Ucmp swagger](#)

Sample Request URL

```
http://localhost:8082/api/v1/Configuration/getDeviceConfigurationCustomList?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A&showLatest=true&Take=10&Skip=0&SortColumn=Devicename&SortDescending=false
```

7.3 DeviceConfiguration Default List

This API is used to get the default configuration packages by EPOS Admin. Fetches list of all raw device packages received and approved.

Result can be filtered using the ShowLatest as TRUE or FALSE (ShowLatest as FALSE will return all the records)

Refer to section [Ucmp swagger](#)

Sample Request URL

```
http://localhost:8082/api/v1/Configuration/ getDeviceConfigurationDefaultList?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A&showLatest=true&Take=10&Skip=0&SortColumn=Devicename&SortDescending=false
```

7.4 Duplicate configuration package

This API is used to duplicate the default configuration packages, afterwards IT manager can edit the configuration and make a custom configuration package.

Refer to section [Ucmp swagger](#)

Sample Request URL

```
http://localhost:8082/api/v1/configuration/duplicateConfigPackage?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A&configurationId=1121&packageName=DuplicateEPOSPacakgeTest
```

7.5 Edit configuration package

Edit configuration on the duplicated package for the device to comply your organization policy

Refer to section [Ucmp swagger](#)

Sample Request URL

<http://localhost:8082/api/v1/Configuration/EditConfigurationPackage>

7.6 Package Details

This API is used to get the specific configuration package details with its respective feature key value pair, afterwards IT manager can provide the key value while editing the package.

Refer to section [Ucmp swagger](#)

Sample Request URL

<http://localhost:8082/api/v1/Configuration/getPackageDetails?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A&configurationId=1121>

7.7 ConfigurationOverview

This API is used to get the list of all devices that are applicable for a configuration and gather all the settings in one place. It also provides the information of user using the device and display the available configuration settings for those connected devices. Version column gives the information about the release versions of the uploaded pack. 'Device Type' filter supports below types.

- Wireless Dect
- Wired USB
- Wireless BT
- EPOS Connect
- Speakerphone

Refer to section [Ucmp swagger](#)

Sample Request URL

<http://localhost:8082/api/v1/Configuration/getConfigurationOverview?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A&deviceType=1&Take=10&Skip=0&SortColumn=&SortDescending=false>

NOTE: Refer to below "DeviceType" types

- 1 = WirelessDect
- 2 = WiredUsb
- 3 = WirelessBt
- 4 = SoftwareConfig
- 5 = Speakerphone

6.8 Deploy Configuration to Group

This API Deploys Device/Software configurations to user groups.

Refer to section [Ucmp swagger](#)

Sample Request URL

<http://localhost:8082/api/v1/configuration/deployConfigurationToGroup>

6.9 Deploy Configuration to User

This API Deploys Device/Software configurations to single user.

Refer to section [Ucmp swagger](#)

Sample Request URL

<http://localhost:8082/api/v1/configuration/deployConfigurationToUser>

6.10 DeleteConfiguration

Refer to section "[DeleteUpdate](#)"

6.11 Cancel Configuration JOB

Refer to section "[CancelJOB](#)"

7 Data insights API

EPOS API has a set of report API's which provides insights on firmware compliance of devices, device status, device distribution over a period of time and call activity data.

7.1 DeviceAdoptionReport

This report provides data on the state of the device i.e. active/inactive on a daily basis.
NOTE: Data is shown for last 7 days only.

Refer to section [Ucmp swagger](#)

Sample Request URL

<http://localhost:8082/api/v1/report/getDeviceAdoptionReport?tenantid=76994A92-2E4C-4791-BABC-D1F7879AB23A&startDate=2019-12-01&endDate=2019-12-31&Take=2&Skip=0>

7.2 UpdateStatusOverview Report

API request is used to give the percentage of the devices which are having the latest version by each device model. This provides the firmware compliance of the devices based on device

family. Version comparison is done based on the FW on the device and the latest version deployed by the Tenant Admin.

Refer to section [Ucmp swagger](#)

Sample Request URL

`http://localhost:8082/api/v1/report/getUpdateStatusOverviewReport?tenantid=76994A92-2E4C-4791-BABC-D1F7879AB23A`

7.3 FirmwareOverview Report

API provides the number of the Devices by different FW version. It provides the latest four FW versions data and the remaining versions (if any) are put under older category.

Refer to section [Ucmp swagger](#)

Sample Request URL

`http://localhost:8082/api/v1/report/getFirmwareOverviewReport?tenantid=76994A92-2E4C-4791-BABC-D1F7879AB23A&devicetype=1`

Available values : 0, 1, 2, 3

- 0 = WirelessBT
- 1 = WirelessDECT
- 2 = WiredUSB
- 3 = Speakerphone

7.4 CallActivity Data

This API provides call statistics i.e. name of the caller/receiver, the start and end time of a call, the device and softphone used for the call and whether the call was an incoming and outgoing call. The call data is provided for the requested time period for a tenant.

Refer to section [Ucmp swagger](#)

Sample Request URL

`http://localhost:8082/api/v1/report/getCallActivityData?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A&startDate=2019-05-05&endDate=2019-12-31&Take=10&Skip=0`

7.5 Dashboard data

This report provides the count of active devices, inactive devices and firmware compliant devices.

7.5.1 ActiveDevices

API request is used to get the information about the total count of active devices to the total number of devices.

Refer to section [Ucmp swagger](#)

Sample Request URL

`http://localhost:8082/api/v1/dashboard/getActiveDevices?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A`

7.5.2 FirmwareCompliance

API request is used to get the information about the total count of firmware versions up to date to the total number of firmware versions.

Refer to section [Ucmp swagger](#)

Sample Request URL

`http://localhost:8082/api/v1/Dashboard/GetFirmwareCompliance?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A`

7.5.3 SoftwareCompliance

API request is used to get the information about the total count of EPOS Connect software versions up to date to the total number of software versions.

Refer to section [Ucmp swagger](#)

Sample Request URL

`http://localhost:8082/api/v1/dashboard/getSoftwareCompliance?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A`

7.5.4 CallActivity

API request is used to get the total number of calls and Total call time which are generated yesterday.

Refer to section [Ucmp swagger](#)

Sample Request URL

`http://localhost:8082/api/v1/dashboard/getCallActivity?tenantId=76994A92-2E4C-4791-BABC-D1F7879AB23A`

8 Troubleshooting

The REST API attempts to use appropriate HTTP status codes to indicate any errors that occur while processing requests. These error responses provide detailed information on the underlying fault that can be used for debugging the error as well as providing user friendly feedback in the app.

8.1 Response Types

HTTP response codes to indicate the success or failure of an API request. In general: Codes in the 2xx range indicate success. Codes in the 4xx range indicate an error that failed given the information provided (e.g., a required parameter was omitted, validation failed, etc.). Codes in the 5xx range indicate an error.

HTTP Code	Error Code	Description
200		Success
400		Generic malformed request.
422		One or more of the required parameters not found in the request URI.
401		Unauthorized
500		Internal server error.
403		Unauthorized token error.
409		Conflict

8.2 Check for token generation

Token-based authentication is a process where the user sends his credential to the server, server will validate the user details and generate a token which is sent as response to the users. Refer to below postman screenshot for successful generation of token.

POST ▼ {{identity.api}}/connect/token Send ▼

Params **Authorization** ● Headers (10) ● Body ● Pre-request Script ● Tests ● Settings

TYPE

Basic Auth ▼

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Username

Password

☐ Show Password

Response

Refer to below successful authenticated response

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjBNjEyODY4NTcxMDBBMjk2NDUxRkI5NjZCMThFODc",
  "expires_in": 3600,
  "token_type": "Bearer",
  "scope": "Ucmp.API.Read Ucmp.API.Write"
}
```

8.3 Failed login

The 401 Unauthorized status code is returned when the API request is missing authentication credentials or the credentials provided are invalid.

Refer to below Postman response header sample

KEY	VALUE
Server ⓘ	Microsoft-IIS/10.0
WWW-Authenticate ⓘ	Bearer error="invalid_token", error_description="The token is expired"
Request-Context ⓘ	appId=cid-v1:e5a8359a-7f7e-40c5-83a2-04b9c3b2487e
Strict-Transport-Security ⓘ	max-age=2592000
X-Powered-By ⓘ	ASP.NET
Set-Cookie ⓘ	ARRAffinity=5d6b9c86d5e41b9e7f57be0bb238693f30751264fd8808dc0ea120.
Date ⓘ	Sat, 21 Mar 2020 17:12:06 GMT
Connection ⓘ	close
Content-Length ⓘ	0

Contact EPOS support
software-help@eposaudio.com
