



Kony Management Intelligent EMM API Guide

Release V8

Document Relevance and Accuracy

This document is considered relevant to the Release stated on this title page and the document version stated on the Revision History page. Remember to always view and download the latest document version relevant to the software release you are using.

Copyright © 2017 Kony, Inc.

All rights reserved.

October, 2017

This document contains information proprietary to Kony, Inc., is bound by the Kony license agreements, and may not be used except in the context of understanding the use and methods of Kony, Inc., software without prior, express, written permission. Kony, Empowering Everywhere, Kony Fabric, Kony Nitro, and Kony Visualizer are trademarks of Kony, Inc. MobileFabric is a registered trademark of Kony, Inc. Microsoft, the Microsoft logo, Internet Explorer, Windows, and Windows Vista are registered trademarks of Microsoft Corporation. Apple, the Apple logo, iTunes, iPhone, iPad, OS X, Objective-C, Safari, Apple Pay, Apple Watch, and Xcode are trademarks or registered trademarks of Apple, Inc. Google, the Google logo, Android, and the Android logo are registered trademarks of Google, Inc. Chrome is a trademark of Google, Inc. BlackBerry, PlayBook, Research in Motion, and RIM are registered trademarks of BlackBerry. SAP® and SAP® Business Suite® are registered trademarks of SAP SE in Germany and in several other countries. All other terms, trademarks, or service marks mentioned in this document have been capitalized and are to be considered the property of their respective owners.

Revision History

Date	Document Version	Description of Modifications/Release
10/09/2017	1.0	Document published for V8

Table of Contents

1. Preface	7
1.1 Intended Audience	7
1.2 Formatting conventions used in this guide	7
1.3 Contact Us	8
2. Introduction	9
3. EMM JS SDK	10
3.1 Configuring Kony Management EMM JS SDK in Kony Studio	10
3.2 Configuring Kony Management EMM JS SDK in Kony Visualizer	16
4. App Management JS-SDK APIs	18
4.1 Camera Policy API	19
4.2 Phone Policy API	21
4.3 Email Policy API	23
4.4 SMS Policy API	26
4.5 AppLock Policy API	28

4.6 IdleTimeOut Policy API	29
4.7 AppExpiry Policy API	32
4.8 ClipBoard Policy API	34
4.9 GeoFence Policy API	36
4.10 Sharing Policy API	39
4.11 Network Policy API	41
4.12 Business Days Policy API	44
4.13 Get User Name API	47
4.14 Get User Details API	48
4.15 Is Device Rooted API	49
5. Device Management JS-SDK APIs	51
5.1 EMM Info API	51
5.2 Device Info API	52
5.3 User Group Information Policy API	54
5.4 Policy Info API	56
5.5 Apply Policy API	65

5.6 Policy Applied Status API	68
5.7 Revert Applied Policy API	70
6. Common EMM JS-SDK APIs	73
6.1 Custom Attributes APIs	73
7. Native SDK APIs	79
7.1 OpenID SSO API	79
7.2 Android APIs	79
7.3 iOS APIs	85

1. Preface

Kony EMM JS-SDK allows developers to build native apps for multiple platforms using Kony Visualizer. Kony EMM JS-SDK APIs currently support both Android and iOS. App developers can use these APIs to create custom experiences for different sets of app users, thereby increasing control and security for these apps.

1.1 Intended Audience

This document is targeted at developers who will use the IDE to write and develop applications.

1.2 Formatting conventions used in this guide

The following formatting conventions are used throughout the document:

Conventions	Explanation
Monospace	■ User input text, system prompts and responses ■ File Path ■ Commands ■ Program Code ■ File Names
<i>Italic</i>	■ Emphasis ■ Names of Books and Documents ■ New Terminology

Conventions	Explanation
Bold	■ Windows ■ Menus ■ Buttons ■ Icons ■ Fields ■ Tabs ■ Folders
<u>URL</u>	Active link to a URL.
<i>Note</i>	Provides helpful hints or additional information.
<i>Important</i>	Highlights actions or information that might cause problems to systems or data.

1.3 Contact Us

We welcome your feedback on our documentation. Write to us at techpubs@kony.com. For technical questions, suggestions, comments, or to report problems on Kony's product line, contact support@kony.com.

2. Introduction

The Kony EMM JS-SDK API reference guide comprises of all the non GUI APIs that you can use on Kony Visualizer. These APIs allow developers to build in security into the apps by invoking several EMM policies and actions at form level. This allows app developers to create custom experiences for different sets of app users, thereby increasing control and security for these apps.

Any app development / design tool can be used to develop native app. In this document, as an example, the JS-SDK integration is shown by using Kony Visualizer.

If you use any other framework to develop mobile apps, when invoking EMM policies and actions, use EMM REST APIs. For more details, refer to [Kony iEMM REST APIs Documentation](#)

The Kony EMM JS-SDK API library consists of the following:

- [App Management JS-SDK APIs](#)
- [Device Management JS-SDK APIs](#)
- [Common JS-SDK APIs](#)
- [Native SDK APIs](#)

3. EMM JS SDK

3.1 Configuring Kony Management EMM JS SDK in Kony Studio

3.1.1 Prerequisites

- JQuery - <http://jquery.com/download/>
- Kony Studio

3.1.2 Downloading Kony JS SDK Files

To download Kony JS SDK, follow these steps:

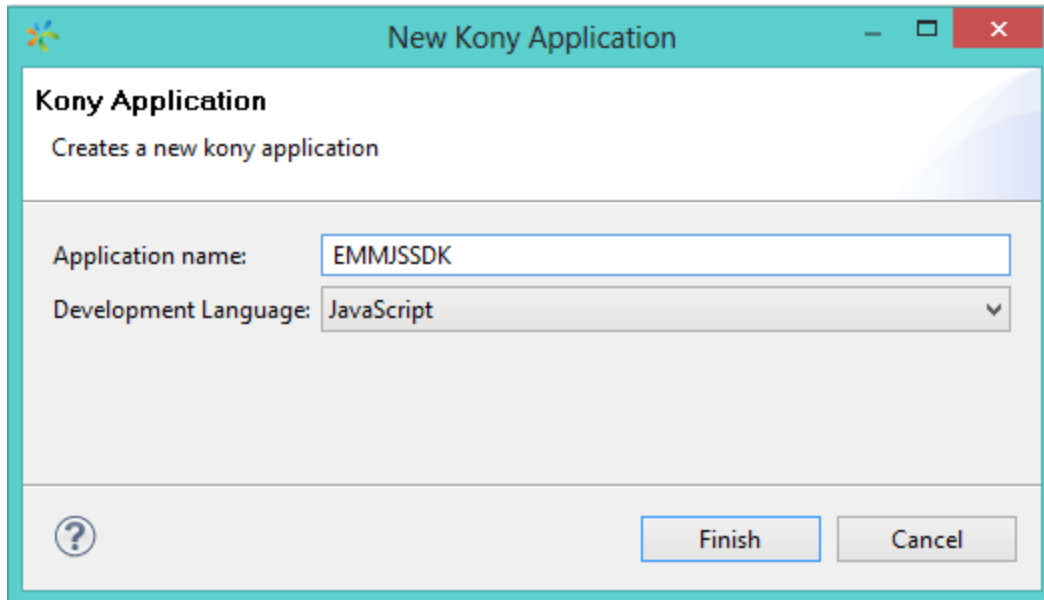
1. Download `emm_js_sdk.zip` from [Kony Portal](#) from the **Kony EMM API Library** section.
2. Save the `emm_js_sdk.zip` file in your local system.

3.1.3 Configuring EMM-JS_SDK in Kony Studio

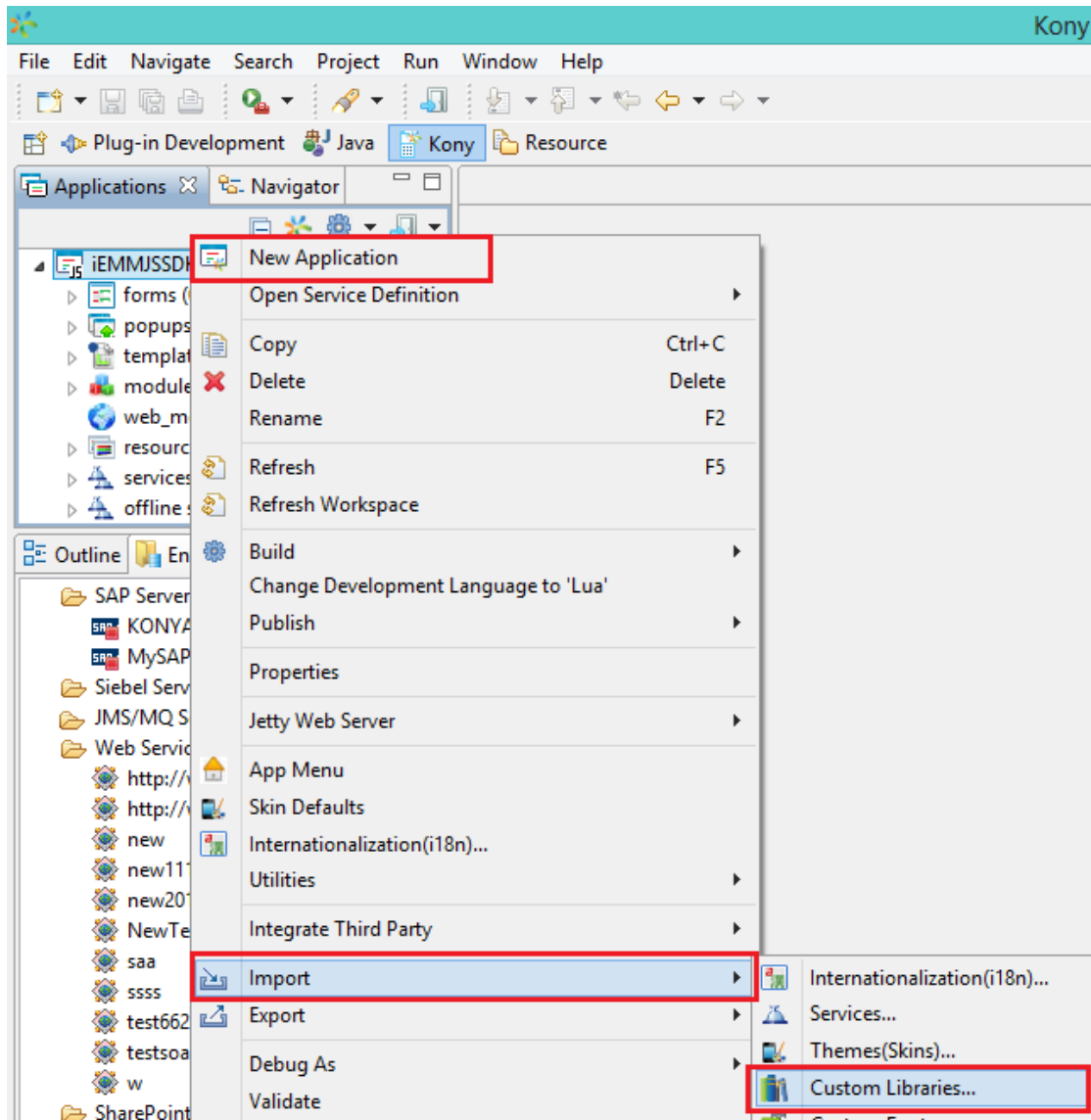
Before using EMM JS libraries, you must configure `emm-js-sdk.zip` into your IDEs, such as Kony Visualizer.

To configure `konysdk.js` file, follow these steps:

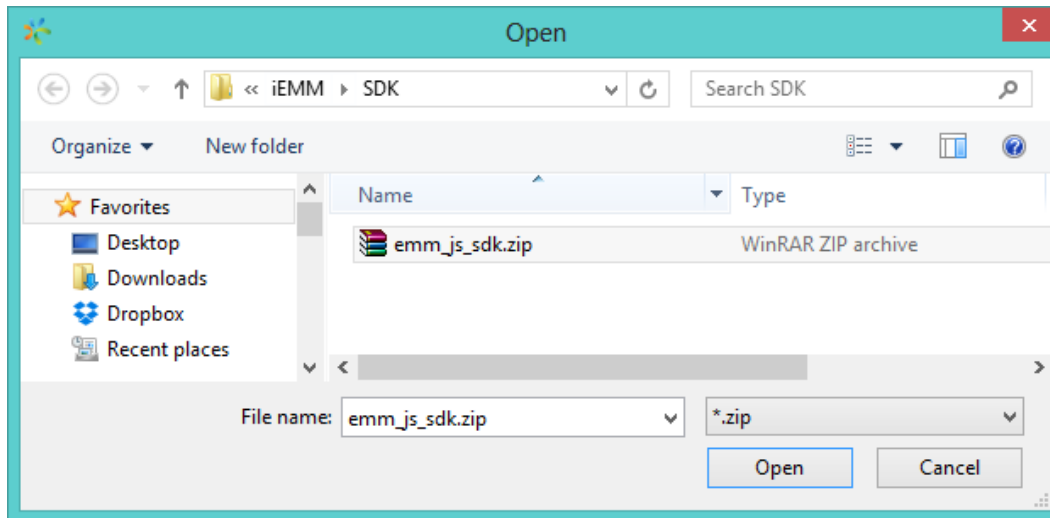
1. In Kony Visualizer, click **File > New > Kony Application**. The **New Kony Application** screen appears.



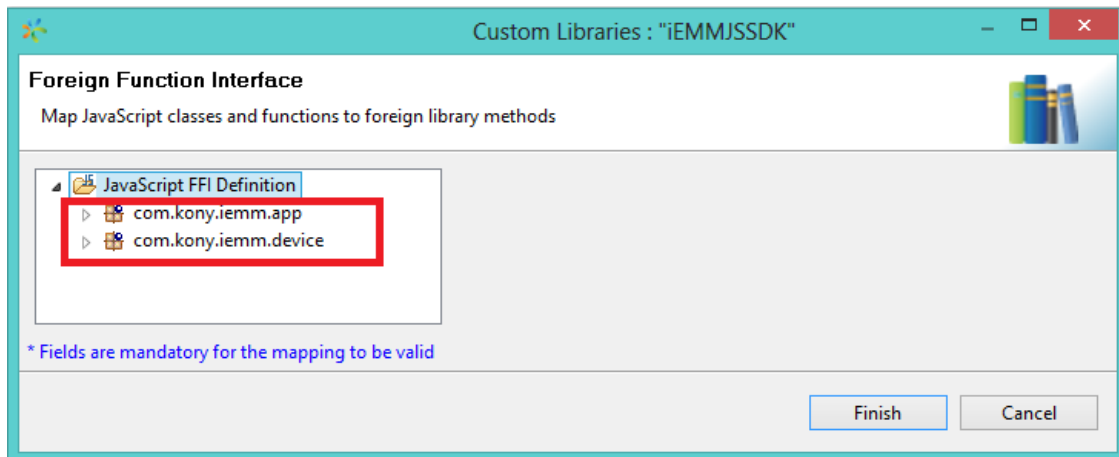
2. In the **New Kony Application** section enter the following details.
 - a. **Application Name:** Enter the application name.
 - b. **Development Language:** Select language as **JavaScript** from the drop-down list.
3. Click **Next**. The new application is created.
4. In the **Application** panel, navigate to your **application > Modules > JS** , and right-click **JS**, then select **Import > Custom Libraries**.



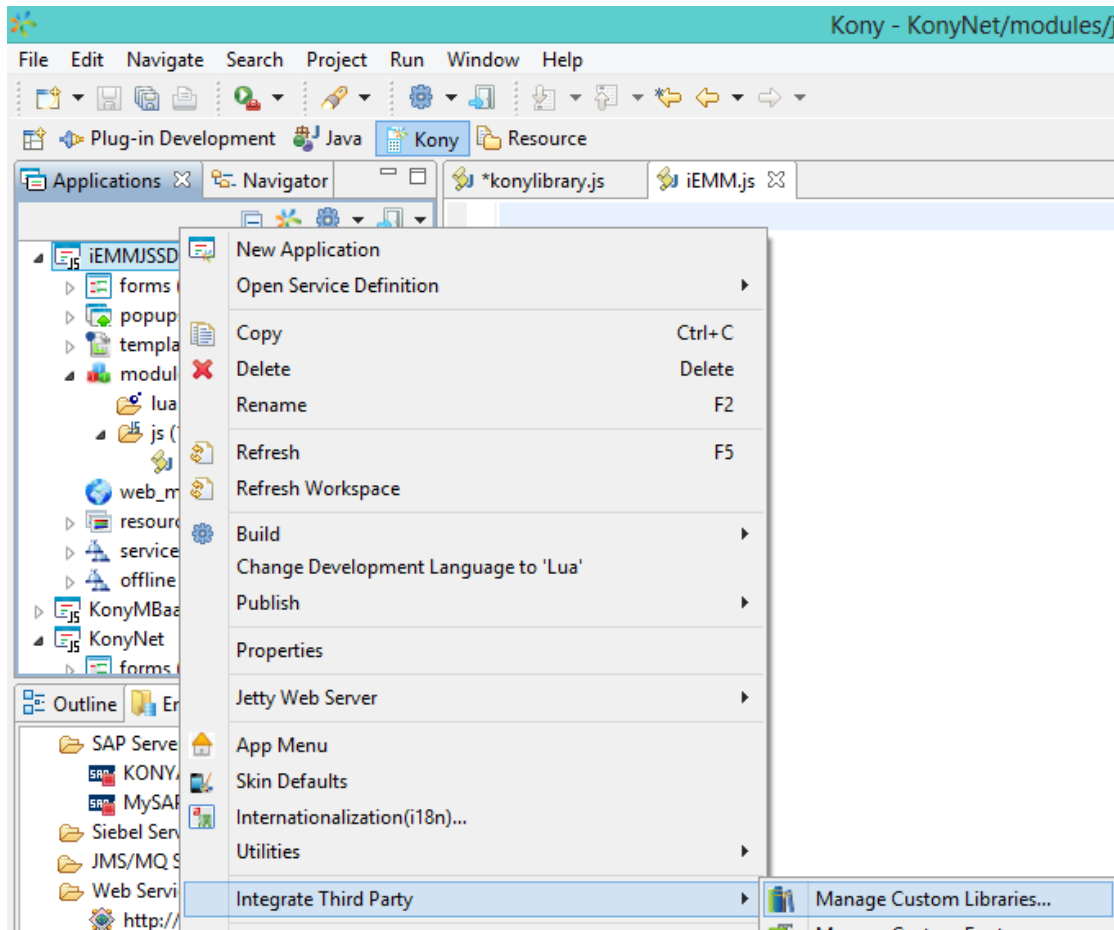
The **Open** dialog appears.



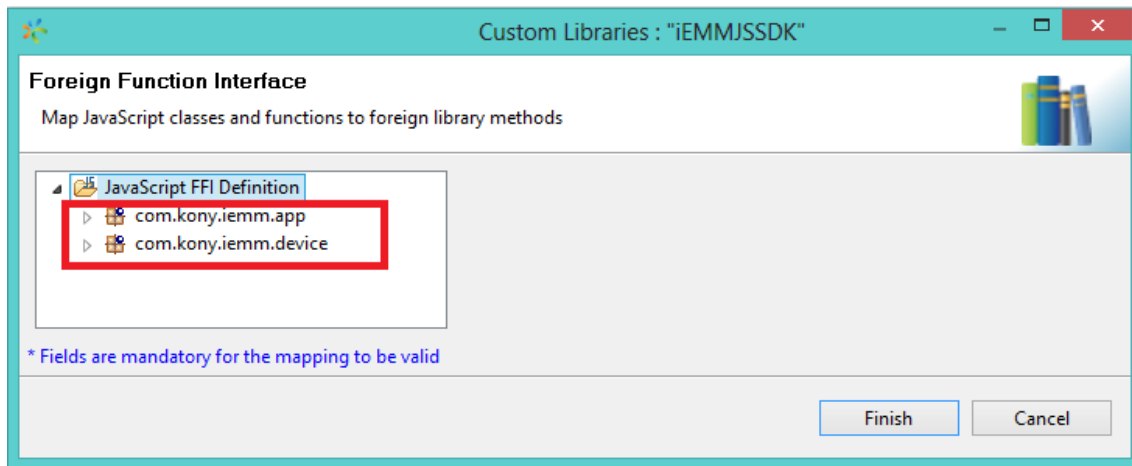
5. Navigate to your local system and then select the `emm_js_sdk.zip` file.



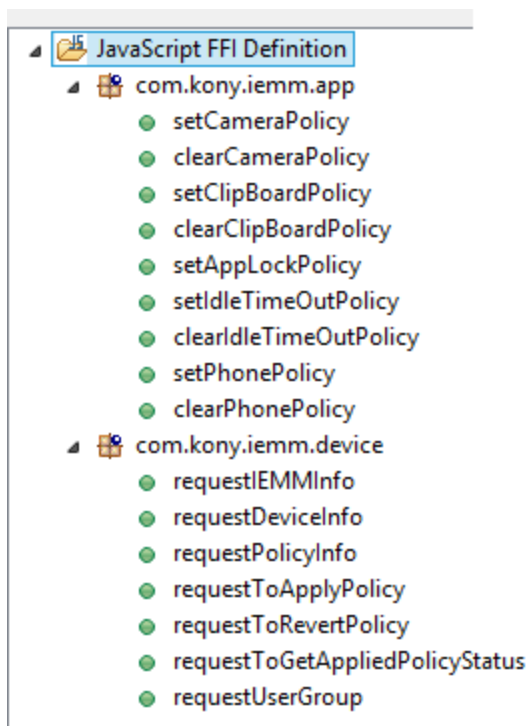
6. Right-click your **application**, and then select **Integrate Third Party > Manage Custom Libraries**.



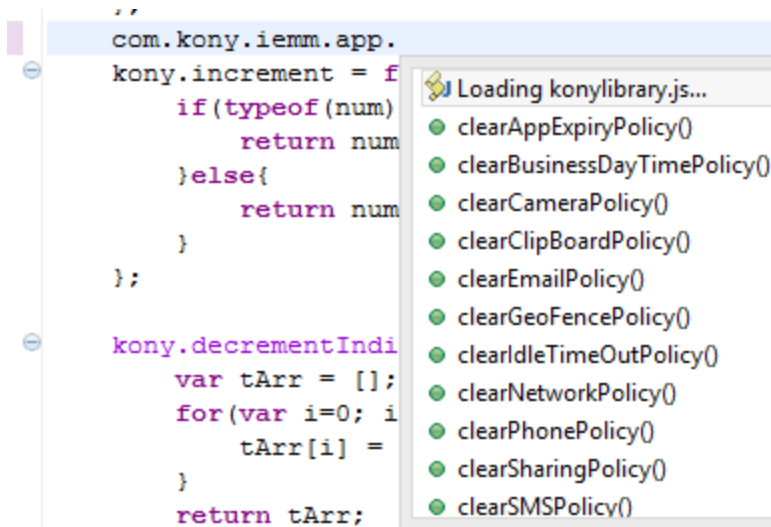
The **Custom Libraries** dialog appears. Now you can see the `com.kony.iemm.app` and the `com.kony.iemm.device` are under the **JavaScript FFI Definition**.



The `EMM JS SDK libraries` are loaded in your application.



You can call iemm functions in your program.



3.2 Configuring Kony Management EMM JS SDK in Kony Visualizer

3.2.1 Prerequisites

- JQuery - <http://jquery.com/download/>
- Kony Visualizer
- Visualizer Sample App. You can download it from [here](#).

3.2.2 Configuring EMM-JS_SDK in Kony Visualizer

You can configure EMM JS SDK in Kony Visualizer in two ways, for a new application, and for an existing application.

To configure EMM JS SDK for a new application in Kony Visualizer Enterprise (7.2.1.2),

1. Download the [Visualizer Sample App](#) from Kony portal.
2. Extract the zip file contents

3. Import the application into Visualizer. For more information on importing a sample app into Visualizer, click [here](#).

To configure EMM JS SDK for an existing application in Kony Visualizer Enterprise (7.2.1.2),

1. Navigate to <WorkspacePath>\Visualizer Sample App\resources folder and copy the **customlibs** folder.
2. Navigate to <WorkspacePath>\Your App\resources folder and paste the **customlibs** folder.
3. Navigate to <WorkspacePath>\Visualizer Sample App folder and copy the **jsFfiMappings** file.
4. Navigate to <WorkspacePath>\Your App folder and paste the **jsFfiMappings** file.

4. App Management JS-SDK APIs

The following APIs can be used by developers to control application functionality. These APIs are policies that take effect on the app specified:

Important: For iOS, these APIs return JSON tables, while for Android, these APIs will return JSON strings. If you want to convert a JSON string to a JSON table, use `JSON.parse` API.

- [Camera Policy API](#)
- [Phone Policy API](#)
- [Email Policy API](#)
- [SMS Policy API](#)
- [AppLock Policy API](#)
- [IdleTimeOut Policy API](#)
- [AppExpiry Policy API](#)
- [Clipboard Policy API](#)
- [GeoFence Policy API](#)
- [Sharing Policy API](#)
- [Network Policy API](#)
- [Business Days Policy API](#)
- [Get User name API](#)
- [Get User Details API](#)
- [Is Device Rooted API](#)

4.1 Camera Policy API

The 'Camera Policy API' helps a developer to enable or disable camera from within an app.

4.1.1 com.kony.iemm.app.setCameraPolicy

This method sets camera policy defined by a developer to an app. This will overwrite camera policy if that is already set from the server.

4.1.1.1 Signature

```
JavaScript:com.kony.iemm.app.setCameraPolicy(status);
```

4.1.1.2 Input Parameters

Variable Name	status
Presence	Mandatory
Type	Boolean
Values	True - Camera should open False - Camera should not open

4.1.1.3 Result Values

Int

4.1.1.4 Result Codes

Code	Description
200	SUCCESS

Code	Description
500	FAILURE
400	FEATURE_NOT_SUPPORTED
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.1.2 com.kony.iemm.app.clearCameraPolicy

This method clears camera policy set by a developer to app. Once the policy is cleared, the server defined camera policy will be applied to an app if any.

4.1.2.1 Signature

```
JavaScript:com.kony.iemm.app.clearCameraPolicy();
```

4.1.2.2 Input Parameters

- none

4.1.2.3 Result Values

Int

4.1.2.4 Result Codes

Code	Description
200	SUCCESS
500	FAILURE

Code	Description
400	FEATURE_NOT_SUPPORTED
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.2 Phone Policy API

The 'Phone Policy API ' helps a developer to enable or disable Phone from within an app.

4.2.1 com.kony.iemm.app.setPhonePolicy

This method sets Phone policy defined by a developer to an app. This will overwrite Phone policy if that is already set from the server.

4.2.1.1 Signature

```
JavaScript:com.kony.iemm.app.setPhonePolicy(allow,numbers);
```

4.2.1.2 Input Parameters

Variable Name	allow
Presence	Mandatory
Type	Boolean
Values	True - Allow phone call to specified numbers False - Restrict phone call to specified numbers

Variable Name	numbers
Presence	Mandatory
Type	String Array
Values	Set of phone numbers in string array

4.2.1.3 Result Values

Int

4.2.1.4 Result Codes

Code	Description
200	SUCCESS
300	NULL_OR_EMPTY_DATA
400	FEATURE_NOT_SUPPORTED
500	FAILURE
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.2.2 com.kony.iemm.app.clearPhonePolicy

This method clears Phone policy set by a developer. Once the dev defined policy is cleared, the server defined Phone policy will be applied to an app if any.

4.2.2.1 Signature

```
JavaScript:com.kony.iemm.app.clearPhonePolicy();
```

4.2.2.2 Input Parameters

- none

4.2.2.3 Result Values

Int

4.2.2.4 Result Codes

Code	Description
200	SUCCESS
500	FAILURE
400	FEATURE_NOT_SUPPORTED
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.3 Email Policy API

The 'Email Policy API ' helps a developer to enable or disable email from within an app.

4.3.1 com.kony.iemm.app.setEmailPolicy

This method sets Email policy defined by a developer to an app. This will overwrite Email policy if that is already set from the server.

4.3.1.1 Signature

```
JavaScript:com.kony.iemm.app.setEmailPolicy(allow,emailsordomains);
```

4.3.1.2 Input Parameters

Variable Name	allow
Presence	Mandatory
Type	Boolean
Values	True - allow app to send emails to specified email id or domain. False - restrict app from sending emails to specified email id or domain.

Variable Name	emailsordomains
Presence	Mandatory
Type	String Array
Values	Set of email or domains in string array

4.3.1.3 Result Values

Int

4.3.1.4 Result Codes

Code	Description
200	SUCCESS
300	NULL_OR_EMPTY_DATA
400	FEATURE_NOT_SUPPORTED
500	FAILURE
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.3.2 com.kony.iemm.app.clearEmailPolicy

This method clears Email policy set by a developer. Once the policy is cleared, the server defined Email policy will be applied to an app if any.

4.3.2.1 Signature

```
JavaScript:com.kony.iemm.app.clearEmailPolicy();
```

4.3.2.2 Input Parameters

- none

4.3.2.3 Result Values

Int

4.3.2.4 Result Codes

Code	Description
200	SUCCESS
500	FAILURE
400	FEATURE_NOT_SUPPORTED
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.4 SMS Policy API

The 'SMS Policy API ' helps a developer to enable or disable SMS from within an app.

4.4.1 com.kony.iemm.app.setSMSPolicy

This method sets SMS policy defined by a developer to an app. This will overwrite SMS policy if that is already set from the server.

4.4.1.1 Signature

```
JavaScript:com.kony.iemm.app.setSMSPolicy(allow,numbers);
```

4.4.1.2 Input Parameters

Variable Name	allow
Presence	Mandatory
Type	Boolean

Values	True - allow app to send sms the specified numbers False - restrict app from adding sms to specified numbers
--------	---

Variable Name	numbers
Presence	Mandatory
Type	String Array
Values	Set of numbers in string array

4.4.1.3 Result Values

Int

4.4.1.4 Result Codes

Code	Description
200	SUCCESS
300	NULL_OR_EMPTY_DATA
400	FEATURE_NOT_SUPPORTED
500	FAILURE
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.4.2 com.kony.iemm.app.clearSMSPolicy

This method clears SMS policy set by a developer. Once the dev defined policy is cleared, the server defined SMS policy will be applied to an app if any.

4.4.2.1 Signature

```
JavaScript:com.kony.iemm.app.clearSMSPolicy();
```

4.4.2.2 Input Parameters

- none

4.4.2.3 Result Values

Int

4.4.2.4 Result Codes

Code	Description
200	SUCCESS
500	FAILURE
400	FEATURE_NOT_SUPPORTED
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.5 AppLock Policy API

The 'AppLock Policy API' helps a developer to lock an app.

4.5.1 com.kony.iemm.app.setAppLockPolicy

This method locks an app. This will overwrite Applock policy if that is already set from the server.

4.5.1.1 Signature

```
JavaScript:com.kony.iemm.app.setAppLockPolicy();
```

4.5.1.2 Input Parameters

- none

4.5.1.3 Result Values

Int

4.5.1.4 Result Codes

Code	Description
200	SUCCESS
400	FEATURE_NOT_SUPPORTED
500	FAILURE
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.6 IdleTimeOut Policy API

The IdleTimeOutPolicy API' helps a developer to set idle time out for an app.

4.6.1 com.kony.iemm.app.setIdleTimeoutPolicy

This method sets IdleTimeout policy defined by a developer to an app. This will overwrite IdleTimeout policy if that is already set from the server.

4.6.1.1 Signature

```
JavaScript:com.kony.iemm.app.setIdleTimeoutPolicy(timeinsecs);
```

4.6.1.2 Input Parameters

Variable Name	timeinsecs
Presence	Mandatory
Type	int
Values	Number of seconds

4.6.1.3 Result Values

Int

4.6.1.4 Result Codes

Code	Description
100	INVALID_DATA
200	SUCCESS
400	FEATURE_NOT_SUPPORTED

Code	Description
500	FAILURE
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.6.2 com.kony.iemm.app.clearIdleTimeoutPolicy

This method clears IdleTimeout policy set by a developer. Once the policy defined, the server defined IdleTimeout policy will be applied to an app if any.

4.6.2.1 Signature

```
JavaScript:com.kony.iemm.app.clearIdleTimeoutPolicy();
```

4.6.2.2 Input Parameters

- none

4.6.2.3 Result Values

Int

4.6.2.4 Result Codes

Code	Description
200	SUCCESS
500	FAILURE
400	FEATURE_NOT_SUPPORTED
600	APP_CONTEXT_NULL_ERRORCODE (android)

Code	Description
700	PACKAGE_NULL_ERRORCODE(android)

4.7 AppExpiry Policy API

The 'AppExpiry Policy API' helps a developer set or clear appexpiry from within an app.

4.7.1 com.kony.iemm.app.setAppExpiryPolicy

This method sets appexpiry policy defined by a developer to an app. This will overwrite appexpiry policy if that is already set from the server.

4.7.1.1 Signature

```
JavaScript:com.kony.iemm.app.setAppExpiryPolicy(datetime);
```

4.7.1.2 Input Parameters

Variable Name	datetime
Presence	Mandatory
Type	string
Values	Format– "MM/dd/yyyy HH:mm@GMT+HH:MM"

4.7.1.3 Result Values

Int

4.7.1.4 Result Codes

Code	Description
100	INVALID_DATA
200	SUCCESS
300	NULL_OR_EMPTY_DATA
400	FEATURE_NOT_SUPPORTED
500	FAILURE
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.7.2 com.kony.iemm.app.clearAppExpiryPolicy

This method clears appexpiry policy set by a developer to app. Once the policy is cleared, the server defined appexpiry policy will be applied to an app if any.

4.7.2.1 Signature

```
JavaScript:com.kony.iemm.app.clearAppExpiryPolicy();
```

4.7.2.2 Input Parameters

- none

4.7.2.3 Result Values

Int

4.7.2.4 Result Codes

Code	Description
200	SUCCESS
500	FAILURE
400	FEATURE_NOT_SUPPORTED
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.8 ClipBoard Policy API

The 'ClipBoard Policy API ' helps a developer to enable or disable clipboard from within an app.

4.8.1 com.kony.iemm.app.setClipboardPolicy

This method sets clipboard policy defined by a developer to an app. This will overwrite clipboard policy if that is already set from the server.

4.8.1.1 Signature

```
JavaScript:com.kony.iemm.app.setClipboardPolicy(allow);
```

4.8.1.2 Input Parameters

Variable Name	allow
Presence	Mandatory
Type	Boolean

Values	True It will allow cut/copy/paste False It will restrict cut/copy/paste
--------	--

4.8.1.3 Result Values

Int

4.8.1.4 Result Codes

Code	Description
200	SUCCESS
400	FEATURE_NOT_SUPPORTED
500	FAILURE
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.8.2 com.kony.iemm.app.clearClipboardPolicy

This method clears clipboard policy set by a developer. Once the developer defined policy is cleared, the server defined clipboard policy will be applied to an app if any.

4.8.2.1 Signature

```
JavaScript:com.kony.iemm.app.clearClipboardPolicy();
```

4.8.2.2 Input Parameters

- none

4.8.2.3 Result Values

Int

4.8.2.4 Result Codes

Code	Description
200	SUCCESS
500	FAILURE
400	FEATURE_NOT_SUPPORTED
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.9 GeoFence Policy API

The 'GeoFence Policy API ' helps a developer to enable or disable Geofence from within an app.

4.9.1 com.kony.iemm.app.setGeoFencePolicy

This method sets Geofence policy defined by a developer to an app. This will overwrite Geofence policy if that is already set from the server.

4.9.1.1 Signature

```
JavaScript:com.kony.iemm.app.setGeoFencePolicy(allow,regions);
```

4.9.1.2 Input Parameters

Variable Name	allow
---------------	-------

Presence	Mandatory
Type	Boolean
Values	True - app usage is allowed in specified regions only False - app usage is restricted in specified regions only

Variable Name	regions
Presence	Mandatory
Type	String Array
Values	Set of geo-fences defined by (lat:long:radius) in string array

4.9.1.3 Result Values

Int

4.9.1.4 Result Codes

Code	Description
100	INVALID_DATA
200	SUCCESS
300	NULL_OR_EMPTY_DATA
400	FEATURE_NOT_SUPPORTED
500	FAILURE

Code	Description
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.9.2 com.kony.iemm.app.clearGeoFencePolicy

This method clears Geofence policy set by a developer. Once the policy is cleared, the server defined Geofence policy will be applied to an app if any.

4.9.2.1 Signature

```
JavaScript:com.kony.iemm.app.clearGeoFencePolicy();
```

4.9.2.2 Input Parameters

- none

4.9.2.3 Result Values

Int

4.9.2.4 Result Codes

Code	Description
200	SUCCESS
500	FAILURE
400	FEATURE_NOT_SUPPORTED
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.10 Sharing Policy API

The 'Sharing Policy API ' helps a developer to enable or disable data sharing from within an app.

4.10.1 com.kony.iemm.app.setSharingPolicy

This method sets Sharing policy defined by a developer to an app. This will overwrite Sharing policy if that is already set from the server.

4.10.1.1 Signature

```
JavaScript:com.kony.iemm.app.setSharingPolicy(allow);
```

4.10.1.2 Input Parameters

Variable Name	allow
Presence	Mandatory
Type	Boolean
Values	True - app is allowed to share documents False - app is not allowed to share documents

4.10.1.3 Result Values

Int

4.10.1.4 Result Codes

Code	Description
200	SUCCESS

Code	Description
400	FEATURE_NOT_SUPPORTED
500	FAILURE
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.10.2 com.kony.iemm.app.clearSharingPolicy

This method clears Sharing policy set by a developer. Once the policy defined, the server defined Sharing policy will be applied to an app if any.

4.10.2.1 Signature

```
JavaScript:com.kony.iemm.app.clearSharingPolicy();
```

4.10.2.2 Input Parameters

- none

4.10.2.3 Result Values

Int

4.10.2.4 Result Codes

Code	Description
200	SUCCESS
500	FAILURE

Code	Description
400	FEATURE_NOT_SUPPORTED
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.11 Network Policy API

The 'Network Policy API' helps a developer to enable or disable Network from within an app.

4.11.1 com.kony.iemm.app.setNetworkPolicy

This method sets Network policy defined by a developer to an app. This will overwrite Network policy if that is already set from the server.

4.11.1.1 Signature

```
JavaScript:com.kony.iemm.app.setNetworkPolicy
(isNwAllowed,NwDomain,forceHTTPS,isAllowWifi,ssid);
```

4.11.1.2 Input Parameters

Variable Name	isNwAllowed
Presence	Mandatory
Type	Boolean
Values	True - network usage is allowed False - network usage is not allowed

Variable Name	NwDomain
Presence	Mandatory
Type	String Array
Values	Set of network domains in string array

Variable Name	forceHTTPS
Presence	Mandatory
Type	Boolean
Values	True - Establish secure http connection(https) False - https not required

Variable Name	isAllowWifi
Presence	Mandatory
Type	Boolean
Values	True - network communication is allowed for mentioned/provided WiFi SSID False - network communication is not allowed in mentioned/provided WiFi SSID

Variable Name	ssid
Presence	Mandatory

Type	Array
Values	Set of ssid in string array

4.11.1.3 Result Values

Int

4.11.1.4 Result Codes

Code	Description
200	SUCCESS
300	NULL_OR_EMPTY_DATA
400	FEATURE_NOT_SUPPORTED
500	FAILURE
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.11.2 com.kony.iemm.app.clearNetworkPolicy

This method clears Network policy set by a developer. Once developer defined policy is cleared, the server defined Network policy will be applied to an app if any.

4.11.2.1 Signature

```
JavaScript:com.kony.iemm.app.clearNetworkPolicy();
```

4.11.2.2 Input Parameters

- none

4.11.2.3 Result Values

Int

4.11.2.4 Result Codes

Code	Description
200	SUCCESS
500	FAILURE
400	FEATURE_NOT_SUPPORTED
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.12 Business Days Policy API

The 'Business Days Policy API' helps a developer set or clear businessday and time from within an app.

4.12.1 com.kony.iemm.app.setBusinessDayTimePolicy

This method sets business day and time policy to an app. This will overwrite businessdaystime policy if that is already set from the server.

4.12.1.1 Signature

```
JavaScript:com.kony.iemm.app.setBusinessDayTimePolicy  
(isAllowed,bdays,time);
```

4.12.1.2 Input Parameters

Variable Name	isAllowed
Presence	Mandatory
Type	Boolean
Values	True - allowed to use False - restricted to use

Variable Name	bdays
Presence	Mandatory
Type	Array
Values	String Array of fully qualified name of day

Variable Name	time
Presence	Optional
Type	String
Values	starthh:startmm-endhh:endmm/GMT+hh:mm

4.12.1.3 Result Values

Int

4.12.1.4 Result Codes

Code	Description
100	INVALID_DATA
200	SUCCESS
300	NULL_OR_EMPTY_DATA
400	FEATURE_NOT_SUPPORTED
500	FAILURE
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.12.2 com.kony.iemm.app.clearBusinessDayTimePolicy

This method clears businessdaystime policy set by a developer to app. Once the policy is cleared, the server defined businessdaystime policy will be applied to an app if any.

4.12.2.1 Signature

```
JavaScript:com.kony.iemm.app.clearBusinessDayTimePolicy();
```

4.12.2.2 Input Parameters

- none

4.12.2.3 Result Values

Int

4.12.2.4 Result Codes

Code	Description
200	SUCCESS
500	FAILURE
400	FEATURE_NOT_SUPPORTED
600	APP_CONTEXT_NULL_ERRORCODE (android)
700	PACKAGE_NULL_ERRORCODE(android)

4.13 Get User Name API

The 'Get User Name API' helps a developer to get display name corresponding to the EMM user id within an app.

4.13.1 com.kony.iemm.app.getUserName

This method gets display name corresponding to the EMM user id

4.13.1.1 Signature

```
JavaScript: com.kony.iemm.app.getUserName();
```

4.13.1.2 Input Parameters

None

4.13.1.3 Result Values

String

4.14 Get User Details API

The Get User Details API helps a developer to get user ID and email ID from the policy sent from the EMM server.

For Android, this API returns a JSON string. For iOS, this API returns a dictionary.

Important: This API is available from the release 3.0.3 onwards.

4.14.1 com.kony.iemm.app.getUserDetails()

This method returns a dictionary that contains the following:

- user name
- user ID
- email ID

4.14.1.1 Signature

```
JavaScript: com.kony.iemm.app.getUserDetails();
```

4.14.1.2 Input Parameters

None

4.14.1.3 Result Values

```
{
  "username": "admin",
  "userid" : "admin",
  "emailid" : "admin@kony.com"
}
```

4.14.1.4 Examples

```
var userDetailsJsonString = com.kony.iemm.app.getUserDetails();
```

The output will be a JSON string

For example,

```
{ "username": "admin",
  "userid" : "admin",
  "emailid" : "admin@kony.com"
}
```

To read the particular key, JSON string is converted in to a JSON table using JSON.parse API.

```
var userDetailsJsonTable=JSON.parse(userDetailsJsonString);
```

Individual keys can be accessed using the following.

- userDetailsJsonTable.username
- userDetailsJsonTable.userid
- userDetailsJsonTable.emailid

4.15 Is Device Rooted API

The 'Is Device Rooted API' helps a developer to get information on a device's rooted or jailbroken status.

4.15.1 com.kony.iemm.app.isDeviceRooted

This method returns a boolean with information on rooted (Android devices) or jailbroken (iOS devices) status of a device.

4.15.1.1 Signature

```
JavaScript: com.kony.iemm.app.isDeviceRooted();
```

4.15.1.2 Input Parameters

None

4.15.1.3 Result Values

Boolean

- True: Device is jailbroken or rooted
- False: Device is not jailbroken or rooted

5. Device Management JS-SDK APIs

The following APIs can be used by developers to control device functionality from within apps.

Important: For iOS, these APIs return JSON tables, while for Android, these APIs will return JSON strings. If you want to convert a JSON string to a JSON table, use `JSON.parse` API.

- [EMM Information](#)
- [Device Information](#)
- [User Group Information](#)
- [Policy Information](#)
- [Apply Policy](#)
- [Policy Applied Status](#)
- [Revert Applied Policy](#)

5.1 EMM Info API

This API provides the user information on whether IEMM services are available on the given tenant or not. Possible values are True and False.

5.1.1 `com.kony.iemm.device.requestIEMMInfo(callback)`

5.1.1.1 Signature

```
JavaScript:com.kony.iemm.device.requestIEMMInfo(callback);
```

5.1.1.2 Input Parameters

Variable Name	callback
---------------	----------

Presence	Mandatory
Type	Your Function

5.1.1.3 Result Values

```
{  
  "description": "success",  
  "responseCode": 1000,  
  "iemmAvailable": true  
}
```

5.1.1.4 Result Codes

Code	Description
1000	SUCCESS
3001	Unexpected error occurred
800	Unknown Error::Http status code::401 Your app JSSDK is trying to make a connection to the server with enhanced security. You must upgrade your JSSDK to the latest version. For more information, click here .

5.2 Device Info API

The 'Device Info API' provides the information about a device enrolled with EMM.

5.2.1 com.kony.iemm.device.requestDeviceInfo(callback)

5.2.1.1 Signature

```
JavaScript:com.kony.iemm.device.requestDeviceInfo(callback);
```

5.2.1.2 Input Parameters

Variable Name	callback
Presence	Mandatory
Type	Your Function

5.2.1.3 Result Values

```
{
  "description": "success",
  "responseCode": 1000,
  "deviceStatus": "ENROLLED",
  "devicePlatform": "IPHONE"
}
```

5.2.1.4 Result Codes

Code	Description
1000	SUCCESS
3001	Unexpected error occurred

Code	Description
3004	IEmm Services are not available
3005	Error - Device is not enrolled with EMM
800	Unknown Error::Http status code::401 Your app JSSDK is trying to make a connection to the server with enhanced security. You must upgrade your JSSDK to the latest version. For more information, click here .

5.3 User Group Information Policy API

This API returns a list of groups to which device user is mapped on the EMM server.

5.3.1 com.kony.iemm.device.requestUserGroup(callback)

5.3.1.1 Signature

```
JavaScript:com.kony.iemm.device.requestUserGroup(callback);
```

5.3.1.2 Input Parameters

Variable Name	callback
Presence	Mandatory
Type	Your Function

5.3.1.3 Result Values

Scenario 1 - User not mapped to any group

```
{"description":"Success",
```

```
"responseCode":1000,
"userGroup":null }
```

Scenario 2 - User mapped to more than one EMM local groups

```
{
"description":"Success",
"responseCode":1000,
"userGroup":
[
{"groupName":"Group2","domain":null},
{"groupName":"Group1","domain":null}
]
}
```

Scenario 3 - User mapped to one group from an Microsoft Active Directory and another EMM local group

```
{ "description": "Success",
"responseCode": 1000,
"userGroup":
[
{ "groupName": "Group1", "domain": "AD-Domain1" },
{ "groupName": "Group2", "domain": null }
]
}
```

5.3.1.4 Result Codes

Code	Description
1000	SUCCESS
3001	Unexpected error occurred

Code	Description
3004	IEMM Services are not available
3005	Error - Device is not enrolled with EMM
800	Unknown Error::Http status code::401 Your app JSSDK is trying to make a connection to the server with enhanced security. You must upgrade your JSSDK to the latest version. For more information, click here .

5.4 Policy Info API

This API is used to query device policy types, device policies and policy applied on the user device.

5.4.1 com.kony.iemm.device.requestPolicyInfo(params,callback)

5.4.1.1 Signature

```
JavaScript:com.kony.iemm.device.requestPolicyInfo(params,callback);
```

5.4.1.2 Input Parameters

Variable Name	callback
Presence	Mandatory
Type	Your Function

Variable Name	params
---------------	--------

Presence	Mandatory
Type	Object
Query	This is mandatory parameter and its possible values are, • policytype • policy • devicepolicy

Query expression	This is optional parameter and it helps user in filtering the output result using simple java expression. Refer the examples for more detail. Possible values: • policytype • id [int] • name [String] • desc [String] • policy • id [int] • name [String] • desc [String] • policyID [int] • state [String] • status [String] • updatorID [String] • creatorID [String] • syncID [String] • devicepolicy • id [int] • name [String] • desc [String] • policyID [int] • state [String] • status [String] • updatorID [String] • creatorID [String] • syncID [String]
------------------	--

5.4.1.3 Result Values

Sample1 - Querying policy catalogue on emm server with queryexpression.

query=policy&queryexpression=name=='samplePasscodePolicy'

Note: queryexpression for string parameter must be enclosed in single quote like 'samplePasscodePolicy'

```
{
  "description": "Success",
  "responseCode": 1000,
  "policyInstances":
  [
    {
      "policyID": "7578646820899786523",
      "id": 1,
      "name": "samplePasscodePolicy",
      "desc": "",
      "state": "ACTIVE",
      "status": "PUBLISHED",
      "syncID": null,
      "creatorID": null,
      "createdTimestamp": "2014-10-31 12:07:18.0",
      "priority": 2,
      "updaterID": null,
      "updatedTimestamp": "2014-10-31 12:07:25.0",
      "publishTimestamp": "2014-10-31 12:07:25.0",
      "policyDescription":
      [
        {
          "constraintKey": "AND_MIN_PASSCODE_LENGTH",
          "constraintName": "Minimum Passcode Length",
```

```
"constraintValue": "0"
},
{
  "constraintKey": "PASSCODE_CONTENT",
  "constraintName": "Passcode Content",
  "constraintValue": "ANY"
},
{
  "constraintKey": "AND_UNIQUE_PASSCODES_BEFORE_REUSE",
  "constraintName": "Unique Passcodes required Before Reuse",
  "constraintValue": "3"
},
{
  "constraintKey": "FAILED_ATTEMPTS_LIMIT",
  "constraintName": "Failed Attempts Limit",
  "constraintValue": "0"
},
{
  "constraintKey": "REQUIRE_STORAGE_ENCRYPTION",
  "constraintName": "Require Storage Encryption",
  "constraintValue": "NO"
},
{
  "constraintKey": "MAX_PASSCODE_AGE_DAYS",
  "constraintName": "Maximum Passcode Age (in days)",
  "constraintValue": "30"
},
{
  "constraintKey": "ALLOW_IDLE_TIME_BEFORE_UNLOCK",
  "constraintName": "Allowed Idle Time (in Minutes)Before (Auto-lock)",
  "constraintValue": "0"
},
{
```

```

"constraintKey": "AND_REQUIRE_PASSCODE",
"constraintName": "Require Passcode",
"constraintValue": "true"
}
],
}

```

Sample 2- Querying policy catalogue on emm server without queryexpression.

query=policy

```

{
description: "Success"
responseCode: 1000
policyInstances: [2]
0: {
policyID: "7583003918865550861"
id: 4
name: "nwPolicy 17"
desc: ""
state: "ACTIVE"
status: "PUBLISHED"
syncID: null
creatorID: null
createdTimestamp: "2014-11-24 13:13:03.903"
priority: 1
policyDescription: [0]
updaterID: null
updatedTimestamp: "2014-11-24 13:59:01.617"
publishTimestamp: "2014-11-24 13:59:01.617"
}
1: {
policyID: "7583003918865550861"
id: 4

```

```
name: "nwPolicy 19"
desc: ""
state: "ACTIVE"
status: "PUBLISHED"
syncID: null
creatorID: null
createdTimestamp: "2014-11-24 13:13:03.903"
priority: 1
policyDescription: [0]
updaterID: null
updatedTimestamp: "2014-11-24 13:59:01.617"
publishTimestamp: "2014-11-24 13:59:01.617"
}
```

Sample3 - Query=devicepolicy

The policy applied on the device and acknowledged.

query=devicepolicy&queryexpression=policyID==7578546384951855393

```
{
  "description": "Success",
  "responseCode": 1000,
  "policyInstances":
  [
    {
      "policyID": "7578546384951855393",
      "id": 4,
      "name": "AutoVirtualNetworkPolicyForWifi",
      "desc": "network policy for virtual policy push testing for wifi",
      "state": "ACTIVE",
      "status": "PUBLISHED",
      "syncID": null,
      "creatorID": null,

```

```

"createdTimestamp": "2014-10-31 01:29:26",
"priority": 4,
"policyDescription":
[
],
"updaterID": "EMM EMM",
"updatedTimestamp": "2014-10-31 09:36:19",
"publishTimestamp": "2014-10-31 01:30:24"
}
]
}

```

Sample 4 - query=policytype querying supported policy types on emmserver.

Note: queryexpression is not supported with query=policytype

query=policytype

```

{
"description": "Success",
"responseCode": 1000,
"policies":
[
{
"id": 1,
"name": "PASSCODE_POLICY",
"desc": "Passcode Policy"
},
{
"id": 2,
"name": "DEVICE_RESTRICTIONS",
"desc": "Device Restrictions"
},

```

```
{
  "id": 3,
  "name": "EMAIL_CALEDAR",
  "desc": "Email and Calendar"
},
{
  "id": 4,
  "name": "NETWORK",
  "desc": "Network"
},
{
  "id": 5,
  "name": "CERTIFICATE_DISTRIBUTION",
  "desc": "Certificate Distribution"
},
{
  "id": 6,
  "name": "WEB_CLIPS",
  "desc": "Web Clips"
},
{
  "id": 8,
  "name": "COMPLAINCE_ACTIONS",
  "desc": "Compliance Actions"
},
{
  "id": 11,
  "name": "MANDATORY_APPS",
  "desc": "App Policy"
},
{
  "id": 12,
  "name": "OTHERS",
```

```
"desc": "Others"
}
]
}
```

5.4.1.4 Result Codes

Code	Description
1000	SUCCESS
3001	Unexpected error occurred
3002	Error - Improper Query: Query provided is not supported
3004	IEmm Services are not available
3005	Error - Device is not enrolled with EMM
800	Unknown Error::Http status code::401 Your app JSSDK is trying to make a connection to the server with enhanced security. You must upgrade your JSSDK to the latest version. For more information, click here .

5.5 Apply Policy API

This service is used to apply a policy on the device. This takes precedence over the policy applied using device set configuration in EMM. This API accepts a comma separated list of policy IDs defined in the EMM server.

5.5.1 com.kony.iemm.device.requestToApplyPolicy(params,callback)

5.5.1.1 Signature

```
JavaScript:com.kony.iemm.device.requestToApplyPolicy  
(params,callback);
```

5.5.1.2 Input Parameters

Variable Name	callback
Presence	Mandatory
Type	Your Function

Variable Name	params
Presence	Mandatory
Type	Object
Query	This is mandatory parameter and its possible value are, • applypolicy
devicesetId	This is an optional parameter and it has comma-separated values of deviceset id which should be applied on device. If this optional parameter is a pass, then it will verify if the device belongs to the devicesetid or not. If it belongs to any of devicesets then policy will be applied on device, otherwise it will be ignored.
policyid	This is a mandatory parameter and its comma-separated values of policy id should be applied on device, please refer examples for more detail.

5.5.1.3 Sample Input

```
{
  "query": "applypolicy",
  "policyid": "7581726638168086964,7502560957311238180"
}
```

5.5.1.4 Result Values

```
{
  "description": "success",
  "responseCode": 1000,
  "requestID": 1041,
  "policyApplied": "true"
}
```

5.5.1.5 Result Codes

Code	Description
1000	SUCCESS
3001	Unexpected error occurred
3004	iEmm Services are not available
3005	Error - Device is not enrolled with EMM
3006	deviceid not enrolled with EMM

Code	Description
3007	Error - Invalid Policy passed
4001	Policy already exists on device
4002	Error - Improper Policy: PolicyID not exist.PolicyID not exist.
4003	Error - Unpublished Policy provided PolicyID state/status should be in APPROVED/PUBLISHED
800	Unknown Error::Http status code::401 Your app JSSDK is trying to make a connection to the server with enhanced security. You must upgrade your JSSDK to the latest version. For more information, click here .

5.6 Policy Applied Status API

This API is used to get the status of the applied policy.

5.6.1 com.kony.iemm.device.requestToGetAppliedPolicyStatus(params, callback)

5.6.1.1 Signature

```
JavaScript:com.kony.iemm.device.requestToGetAppliedPolicyStatus  
(params, callback);
```

5.6.1.2 Input Parameters

Variable Name	callback
Presence	Mandatory
Type	Your Function

Variable Name	params
Presence	Mandatory
Type	Object
Query	This is a mandatory parameter and the required value is • policystatus
requestid	This comes part of response from Apply Policy API call. It needs to be passed here to fetch the status of that API call. This call returns true when EMM get acknowledgment from the device.

5.6.1.3 Sample Input

```
{
  "query": "policystatus",
  "requestId": 779
}
```

5.6.1.4 Result Values

```
{
  "description": "success",
  "responseCode": 1000,
  "isAcknowledged": true
}
```

5.6.1.5 Result Codes

Code	Description
1000	SUCCESS
3001	Unexpected error occurred
3004	IEMM Services are not available
3005	Error - Device is not enrolled with EMM
3008	Error - Invalid/Missing RequestId
800	Unknown Error::Http status code::401 Your app JSSDK is trying to make a connection to the server with enhanced security. You must upgrade your JSSDK to the latest version. For more information, click here .

5.7 Revert Applied Policy API

This API reverts an applied policy on a device.

5.7.1 com.kony.iemm.device.requestToRevertPolicy(params, callback)**5.7.1.1 Signature**

```
JavaScript:com.kony.iemm.device.requestToRevertPolicy(params,  
callback);
```

5.7.1.2 Input Parameters

Variable Name	callback
Presence	Mandatory
Type	Your Function

Variable Name	params
Presence	Mandatory
Type	Object
Query	This is a mandatory parameter and the required value is • revertpolicy
requestid	This is an optional parameter. With every applied policy service, it returns one requestid. Passing that requestid in this service, the device will return to the state it was before the policy was applied. By default will revert all the policies applied through services and maintain the state as at the time of enrollment.

5.7.1.3 Sample Input

```
{  
  "query": "revertpolicy",  
  "requestId": 779  
}
```

5.7.1.4 Result Values

```
{
  "description": "success",
  "responseCode": 1000,
  "policiesRevert": true
}
```

5.7.1.5 Result Codes

Code	Description
1000	SUCCESS
3001	Unexpected error occurred
3004	IEmm Services are not available
3005	Error - Device is not enrolled with EMM
4004	Error - Existing device policies are empty .Existing device policies returning null
4005	Policies already reverted for given requestid
800	Unknown Error::Http status code::401 Your app JSSDK is trying to make a connection to the server with enhanced security. You must upgrade your JSSDK to the latest version. For more information, click here .

6. Common EMM JS-SDK APIs

The following are the common APIs a developer can use in EMM:

- [Custom Attribute APIs](#)

6.1 Custom Attributes APIs

The following are various APIs used for Custom Attribute Sets feature of EMM.

- [User Custom Attributes API](#)
- [User Group Custom Attributes API](#)
- [App Custom Attributes API](#)
- [Device Custom Attributes API](#)

6.1.1 User Custom Attributes API

The User Custom Attributes API returns a list of custom attributes targeted to a user.

Note: Cloud user cannot be assigned Custom Attributes. When a cloud user is a member of a group that has Custom Attributes, the cloud user can use group custom attributes, but cannot use user level custom attributes.

6.1.1.1 `com.kony.iemm.device.requestToUserCustomAttributes(callback)`

Signature

```
JavaScript:com.kony.iemm.device.requestToUserCustomAttributes  
(callback);
```

Input Parameters

Variable Name	callback
Presence	Mandatory
Type	Your function

Result Values

```
{  
  "key1": "value1",  
  "key2": "value2",  
  "key3": "value3"  
}
```

Result Codes

Code	Description
1000	SUCCESS
1003	Error - Missing Device ID: Device ID should not be empty.
1009	Error - User Not Enrolled with EMM

6.1.2 User Group Custom Attributes API

The User Group Custom Attributes API returns a list of custom attributes targeted to a user group.

6.1.2.1 com.kony.iemm.device.requestToUserGroupCustomAttributes(grpid,callback)**Signature**

```
JavaScript:com.kony.iemm.device.requestToUserGroupCustomAttributes  
(params,callback);
```

Input Parameters

Variable Name	callback
Presence	Mandatory
Type	Your function

Variable Name	grpid
Presence	Mandatory
Type	JSON Object {"grpid" : group id (from user group service)}

Result Values

```
{  
  "key1": "value1",  
  "key2": "value2",  
  "key3": "value3"  
}
```

Result Codes

Code	Description
1000	SUCCESS
1010	Group ID cannot be empty
1011	Group Not Found
3005	Error - Device is not enrolled with EMM.

6.1.3 App Custom Attributes

The App Custom Attributes API returns a list of custom attributes targeted to a child application.

6.1.3.1 com.kony.iemm.device.requestToAppCustomAttributes(callback)**Signature**

```
JavaScript:com.kony.iemm.device.requestToAppCustomAttributes  
(callback);
```

Input Parameters

Variable Name	callback
Presence	Mandatory
Type	Your function

Result Values

```
{
  "key1": "value1",
  "key2": "value2",
  "key3": "value3"
}
```

Result Codes

Code	Description
1000	SUCCESS
1003	Error - Missing Device ID: Device ID should not be empty.

6.1.4 Device Custom Attributes API

The Device Custom Attributes API returns a list of custom attributes targeted to a device.

6.1.4.1 com.kony.iemm.device.requestToDeviceCustomAttributes(callback)

Signature

```
JavaScript:com.kony.iemm.device.requestToDeviceCustomAttributes
(callback);
```

Input Parameters

Variable Name	callback
---------------	----------

Presence	Mandatory
Type	your function

Result Values

```
{  
  "key1": "value1",  
  "key2": "value2",  
  "key3": "value3"  
}
```

Result Codes

Code	Description
1000	SUCCESS
1003	Error - Missing Device ID: Device ID should not be empty

7. Native SDK APIs

Native SDK APIs in EMM are the APIs which must be imported into native projects of iOS or Android for a feature to work. You must download all required native sdk APIs from the kony portal.

- For Android APIs, import Android SDK APIs into your Android app project.
- For iOS APIs, import iOS SDK APIs into your iOS Xcode app project.

The following feature uses native SDK APIs

- [Open ID SSO](#)

7.1 OpenID SSO API

OpenID SSO APIs help a developer to use the Single Sign On feature for an app submitted to the EMM.

Use OpenID SSO APIs only if you have selected the **Only Sign** option in the **Signing Rule** feature. When you select **Only Sign**, the **Allow SSO** feature is enabled in the EMM management console. You must select **Yes** for the **Allow SSO** feature for these APIs to work.

7.2 Android APIs

Important: You must import the `sso_sdk.jar` file into your Android project to use SSO APIs.

7.2.1 `int: com.kony.signonly.ssohelper.SSOHelper.storeCookie (java.util.Hashtable <java.lang.String,java.lang.Object> params);`

The storeCookie API stores a single cookie entry in the common storage for the given data.

7.2.1.1 Signature

```
int:com.kony.signonly.ssohelper.SSOHelper.storeCookie
(java.util.Hashtable <java.lang.String,java.lang.Object> params);
```

7.2.1.2 Input Parameters

Variable Name	params
Presence	Mandatory
Type	Hashtable
Values (Keys in the Hashtable)	name - The name of the cookie (java.lang.String type and mandatory). value - The value of the cookie (java.lang.String type and mandatory). domain - The domain against which the cookie is stored (java.lang.String type and mandatory). path - The path against which the cookie is stored. If the path key is missing or if the value is null, then the path is ignored (java.lang.String type and optional). context - The application context for the app (android.content.Context type and mandatory). Note: Use the getApplicationContext() API to get the application context.

7.2.1.3 Result Values

Int

7.2.1.4 Result Codes

Code	Description
200	SUCCESS
400	FAILURE
201	NULL input
202	INSUFFICIENT number of arguments
203	Missing domain key
204	NULL or EMPTY domain
205	Missing name key
206	NULL or EMPTY name
207	Missing value key
208	NULL or EMPTY value
209	Missing context
210	Null context

**7.2.2 `java.util.HashMap:com.kony.signonly.ssohelper.SSOHelper.getCookie`
(`java.util.HashMap<java.lang.String,java.lang.Object>` params)**

The `getCookie` API retrieves all cookies from the common storage (cookies that match the combination of given domain and path).

If the path provided is null or empty, then all cookies that match the given domain are retrieved from the common storage.

7.2.2.1 Signature

```
java.util.Hashtable:com.kony.signonly.ssohelper.SSOHelper.getCookie  
(java.util.Hashtable  
<java.lang.String,java.lang.Object> params)
```

7.2.2.2 Input Parameters

Variable Name	params
Presence	Mandatory
Type	Hashtable
Values (Keys in the Hashtable)	domain - The domain for which you need to retrieve the cookies (java.lang.String type and mandatory). path - The path in which the cookie is stored. If the path key is missing or if the value is null, then the path is ignored (java.lang.String type and mandatory). context - The application context for the app (android.content.Context type and mandatory). Note: Use the <code>getApplicationContext()</code> API to get the application context.

7.2.2.3 Result Values

Hashtable

For success, a hashtable with resultcode and cke keys (cookies in the json format) are returned. For example,

```
{ "cookies":
  [{"secure":true,"path":"\/","domain":"appdirect.com","value":"yy","name":"AWSELB"}
  ,
  {"secure":true,"path":"\/","domain":"appdirect.com","value":"1","name":"__utmt_rollup"}]}
```

7.2.2.4 Result Codes

Code	Description
200	SUCCESS
400	FAILURE
201	NULL input
202	INSUFFICIENT number of arguments
203	Missing domain key
204	NULL or EMPTY domain
209	Missing context
210	Null context

7.2.3 `int:com.kony.signonly.ssohelper.SSOHelper.deleteCookie(java.util.Hashtable<java.lang.String,java.lang.Object> params)`

The deleteCookie API deletes all cookies that match the combination of provided domain and path from the common storage.

If the path provided is null or empty, then all cookies that match the provided domain are deleted from the common storage.

7.2.3.1 Signature

```
int:com.kony.signonly.ssohelper.SSOHelper.deleteCookie
(java.util.Hashtable
<java.lang.String,java.lang.Object> params)
```

7.2.3.2 Input Parameters

Variable Name	params
Presence	Mandatory
Type	Hashtable
Values (Keys in a hashtable)	domain - The domain for which you need to delete the cookies (java.lang.String type and mandatory) path - The path in which the cookie is stored. If the path key is missing or if the value is null, then the path is ignored (java.lang.String type and optional). context - The application context for the app (android.content.Context type and mandatory). Note: Use the getApplicationContext() API to get the application context.

7.2.3.3 Result Values

Int

This API returns the number of cookies deleted (if any cookies were stored) based on the combination given in the domain and the path values.

7.2.3.4 Result Codes

Code	Description
400	FAILURE
201	NULL input
202	INSUFFICIENT number of arguments
203	Missing domain key
204	NULL or EMPTY domain
209	Missing context
210	Null context

7.3 iOS APIs

Important: You must import **SSOHelper.h** and **libSSOHelper.a** files into your XCode project to use SSO APIs.

Note: NSString and NSDictionary are part of the Foundation framework under the iOS SDK.

7.3.1 [SSOhelper storeCookie:params];

The storeCookie API stores a single cookie entry in the common storage for the given data.

You must declare a function before using this API.

Function Declaration: `+(int) storeCookie:(NSDictionary *)params;`

7.3.1.1 Signature

```
[SSOhelper storeCookie:params];
```

Important: Params in the signature above is the NSDictionary declared in the function earlier.

7.3.1.2 Input Parameters

Variable Name	params
Presence	Mandatory
Type	NSDictionary
Values	name - The name of the cookie (NSString type and mandatory). value - The network usage is allowed (NSString type and mandatory). domain - The domain against which the cookie must be set (NSString type and mandatory). path - The path against which the cookie must be set (NSString type and optional). If the path key is missing or if the value is null, then the path is ignored.

7.3.1.3 Result Values

Int

7.3.1.4 Result Codes

Code	Description
200	SUCCESS
400	FAILURE

Code	Description
201	NULL input
202	INSUFFICIENT number of arguments
203	Missing domain key
204	NULL or EMPTY domain
205	Missing name key
206	NULL or EMPTY name
207	Missing value key
208	NULL or EMPTY value

7.3.2 [SSOHelper getCookie : params];

The getCookie API retrieves all cookies from the common storage that match the combination of given domain and path.

If the path given is null or empty, then all cookies that match the provided domain are retrieved from the common storage.

You must declare a function before using this API.

Function Declaration: +(NSDictionary *) getCookie: (NSDictionary *) params;

```
[SSOHelper getCookie : params];
```

Important: Params in the signature above is the NSDictionary declared in the function earlier.

7.3.2.1 Input Parameters

Variable Name	params
Presence	Mandatory
Type	NSDictionary
Values	domain - The domain for which cookies need to be retrieved (NSString type and mandatory). path - The path in which the cookie is stored (NSString type and optional) . If the path key is missing or if the value is null, then the path is ignored.

7.3.2.2 Result Values

NSDictionary (with **resultcode** and **cke** keys)

7.3.2.3 Result Codes

Code	Description
200	SUCCESS For success, along with a result code a cke (cookies in the json format) key returns. For example, <pre>{ "cookies": [{<NSHTTPCookie version:0 name:\"ajs_group_id\" value:\"null\" expiresDate:2016-04-22 11:47:46 +0000 created:2015-04-23 11:47:46 +0000 (4.51482e+08) sessionOnly:FALSE domain:\".appdirect.com\" path:\"/\" isSecure:FALSE}> {<NSHTTPCookie version:0 name:\"_gauges_unique\" value:\"1\" expiresDate:2025-04-20 11:41:47 +0000 created:2015-04-23 11:41:47 +0000 (4.51482e+08) sessionOnly:FALSE domain:\"www.appdirect.com\" path:\"/\" isSecure:FALSE}> }] }</pre>

Code	Description
400	FAILURE
201	NULL input
202	INSUFFICIENT number of arguments
203	Missing domain key
204	NULL or EMPTY domain
211	No cookies found that match the combination of the given domain and path.

7.3.3 [SSOHelper deleteCookie : params];

The deleteCookie API deletes all cookies that match the combination of given domain and path from the common storage.

If the path is given as null or empty then all cookies that match the given domain are deleted from the common storage.

You must declare a function before using this API.

Function Declaration: +(int)deleteCookie :(NSDictionary *)params;

7.3.3.1 Signature

```
[SSOHelper deleteCookie : params];
```

Important: Params in the signature above is the NSDictionary declared in the function earlier.

7.3.3.2 Input Parameters

Variable Name	params
Presence	Mandatory
Type	NSDictionary
Values	domain - The domain for which cookies need to be deleted (NSString and mandatory). path - The path in which the cookie is stored (NSString and optional). If the path key is missing or if the value is null, then the path is ignored.

7.3.3.3 Result Values

Int

If success, this API returns the number of cookies deleted (based on the combination given in domain and path values).

7.3.3.4 Result Codes

Code	Description
200	SUCCESS
400	FAILURE
201	NULL input
202	INSUFFICIENT number of arguments
203	Missing domain key
204	NULL or EMPTY domain