API Reference Document for MobileIron WebService

for MobileIron Core 10.4.0.0

July 17, 2019

Contents

Important Note on v1 API Deprecation

MobileIron is deprecating some v1 APIs in favor of their new v2 API counterparts and MobileIron will not concurrently support some of the deprecated v1 versions. Please see this KB article for details.

Document Overview

This document provides development information for customers and partners intending to use MobileIron WebService APIs.

The initial sections provide general guidelines and conventions for reference.

The main part of the document includes API descriptions, which are categorized as follows:

- Exchange ActiveSync – retrieves detailed information for EAS devices and provides a way to act on those devices.
- Security management –helps a user with password protection.
- Alerts – allows for alert retrieval and updates.
- Applications – provides an inventory of installed applications.
- Policies
- Application settings
- AppConnect analytics

Each of the above categories contains one or more API calls. In most cases, there is an API description, a URI, a set of mandatory or optional request parameters, response status codes, and the response (with example data included). The input parameters and output response include definitions where necessary. Please refer to the Administration Guide for additional background and details on how these functions behave in the UI.

The end of the document includes a sample http test client implemented in Java.

General Guidelines and Conventions

Parameter Formats

The HTTP requests use two types of parameters:

- Path parameters
- Query parameters

Path Parameters

Path parameters continue the URI path using a slash (/) for a separator. For example, to get details for a device, you specify its device uuid with a path parameter. The following shows the URI format for this request, and an example:

```
https://{host-name}/api/v1/dm/devices/{deviceuuid}
https://mycore.mobileiron.com/api/v1/dm/devices/4239b999-46e3-423b-b808-54fff69b544c
```

If a request uses path parameters, they are specified in the URI format for the request.

Query parameters

Query parameters are included in the URI path using a question mark or ampersand (? or &). For example, to retire a device, after specifying the device uuid as a path parameter, you specify a reason as a query parameter. The following shows the URI format for this request, and an example:

```
https://{host-name}/api/v1/dm/devices/retire/{deviceuuid}
https://mycore.mobileiron.com/api/v1/dm/devices/retire/c097c9e2-c82e-40f6-9e69-
a0478c4fcee0?reason=AnyReasonTextYouChoose
```

The first query parameter is preceded by a question mark (?), using the following format:

?parameterName=parameterValue

Subsequent query parameters are preceded by an ampersand (&), using the following format:

&parameterName=parameterValue

For example, to get all the Android devices that have a particular application installed, use the following request:

```
https://mycore.mobileiron.com/api/v1/apps/inventory/app?appname=Frog%20Toss!_
017%201.0&platform=A
```

Note: If a parameter is not shown in a request's URI format with a slash, it is a query parameter; the URI format for a request shows only the path parameters. The description that follows each URI format provides information on the query parameters, if any.

Date Formats

Many API calls include start and end dates in the request. In general, the dates are optional. If dates are not included, all available records will be returned. If start and end dates are included in the request, only records within the date range will be returned.

Dates can be in the following formats

- Jan 1 2010
- January 1, 2010
- January 1, 2010, 00:00:00
- UTC format: YYYY-MM-DDThh:mm:ssTZD
  - For example: 2010-03-10T15:04:06+00:00 which is March 10, 2010 3:04:06 PM
- Alternate format: MM-DD-YYYY hh:mm:ss
  - For example: 03-12-2010 13:23:12 which is March 12, 2010 1:23:12 PM

Phone Number Formats

Many API calls require a phone number in the request. The following rules apply to an input phone number:

- Enter numbers only.
- Do not include a country code.
- Do not include parenthesis, dashes, periods, or other special characters.

HTTP request methods

Depending on the HTTP request, use one of the following HTTP request methods:

- Get – Use for requests that retrieve information from MobileIron Core.
- Put – Use for requests that change information on MobileIron Core.
- Post – Use for the bulk requests that perform actions on many devices, or for requests that provide substantial information on MobileIron Core.

Each request description specifies which HTTP request method to use.

Response Formats

Requests to the API can return xml or json, based on the request headers.

- For xml output, set the 'Accept' header in the request to 'application/xml'.
- For json output, set the 'Accept' header in the request to 'application/json'.

HTTP Response Codes

Responses from the API use the codes listed below. In addition, a "Success" message is shown for successful method executions. When method executions fail, a descriptive error message is displayed.

- 200 OK: Success
- 400 Bad request: The request was invalid. The accompanying error message in the output explains the reason.
- 401 Unauthorized: Authentication to the API has failed. Authentication credentials are missing or wrong.
- 404 Not found: The requested resource is not found. The accompanying error message explains the reason.
- 405 Method Not Allowed: The HTTP request method that was specified is not the correct method for the request.
- 500 Internal Server Error: An internal server error has occurred while processing the request.
- 502 Bad Gateway: The MobileIron server is not reachable.

Using offset and limit Parameters to Cycle through Records

Some requests result in responses that contain many records. For example, the request for the list of all devices on which a specific application is installed can match hundreds or thousands of devices.

To return more than the first 100 records, these APIs support the limit and offset query parameters. These parameters allow you to get successive sets of records in successive responses.

Specifically, use these query parameters to do the following:

- Limit the number of records returned in the response to a number you choose, using the query parameter limit. For example, the following request returns the first 50 devices that have the LinkedIn app installed:

  ```
  https://
  mycore.mobileiron.com/api/v1/apps/inventory/app?appname=LinkedIn&limit=50
  ```

- Specify the index of the first record to return in the response, using the query parameter offset. The value is zero-based. For example, the following request returns 50 devices, starting with the 101st device:

  ```
  https://
  mycore
  .mobileiron.com/api/v1/apps/inventory/app?appname=LinkedIn&limit=50&offset=100
  ```

  The offset parameter defaults to 0. Therefore, both of the following requests return 50 devices, starting with the first device:

  ```
  https://
  mycore
  .mobileiron.com/api/v1/apps/inventory/app?appname=LinkedIn&limit=50&offset=0
  https://
  mycore.mobileiron.com/api/v1/apps/inventory/app?appname=LinkedIn&limit=50
  ```

Therefore, to get successive sets of records in successive responses, increase the offset value by the limit value in each request. For example:

```
https://mycore.mobileiron.com/api/v1/apps/inventory/app?appname=LinkedIn&limit=50&offset=0
https://mycore.mobileiron.com/api/v1/apps/inventory/app?appname=LinkedIn&limit=50&offset=50
https://mycore.mobileiron.com/api/v1/apps/inventory/app?appname=LinkedIn&limit=50&offset=100
```

For the API that gets the devices that have a particular app installed (Get Devices by Application Name), you can also set limit to -1 to get all the devices in one response. For example:

```
https://mycore.mobileiron.com/api/v1/apps/inventory/app?appname=LinkedIn&limit=-1
```

The following table summarizes the limit and offset query parameters.

Query parameter	Description	Default value	Special value
limit	Maximum number of records to show in the response.	100 Note: This default value applies only to APIs that support the limit query parameter. Other APIs always return all applicable records in the response.	-1 Shows all records in the response. Note: This special value only applies to the apps/inventory/app API.
offset	Zero-based index of first record to show in the response	0	

Device and User Identifiers

The requests and responses use identifiers for devices and users.

Every time a user or a device is created, MobileIron Core internally generates a unique identifier, called a “uuid”. Each device gets a unique uuid. Because a single user can have multiple devices, each user gets a unique uuid, independent of any devices used.

Other IDs are used for other purposes, such as identifying a network, and are described in the APIs.

Operating System Dependencies

Refer to the Administrator’s Guide for an up-to-date matrix which displays supported features by operating system and platform. If a particular feature is not supported by an OS, the feature API will not return a valid response.

Supported Browsers and Recommended Plugins

FireFox and Chrome are the supported browsers. Internet Explorer is not supported. Plugins like Poster for FireFox and Advanced REST Client for Chrome are recommended for interacting with the http server; simply submitting a URI in the browser does not consistently result in a properly rendered response.

General Practice

All the requests are wrapped in WebServiceRequest, and the responses are wrapped in WebServiceResponse. startDate and endDate from the request are returned in the response.

Authentication

Access to the web service is granted using roles. The ability to grant role access is available to administrators that are assigned the role 'Manage administrators and device spaces'. These Super Administrators can assign the 'API' role to a user with the following steps:

1. From the Admin Portal, select Admin > Admins.
2. Select a user from the list of users.
3. Select Actions > Edit Roles.
4. Select the 'API' role, which is listed under Others.
5. Click Save.

Username/Password

The web service requires authentication via username and password:

- Username: Username of any local or LDAP user who has the 'API' role.
- Password: The same password used to login to MobileIron Core the Admin Portal.

WADL

The WADL (Web Application Description Language) is an xml interface file between client and server. The WADL file is present in the API test client zip file. To view the generated WADL, open the following URL from a browser:

https://{host-name}/api/?_wadl&_type=xml

Device Management

Device Management APIs allow administrators to retrieve a variety of details for devices based on varying search criteria. These APIs can also register, retire, wipe, lock, unlock, locate, and wakeup a device. Lastly, labels, countries, and operators can be retrieved from these APIs.

Status and statusCode values

Some Device Management APIs refer to device "status" and "statusCode" in either requests or responses. Possible values for these variables are listed here:

status:

- ACTIVE – Active devices
- IENROLL_VERIFIED – Enrollment verified devices for iPhone and WebOS
- IENROLL_INPROGRESS – Enrolling devices for iPhone and WebOS
- IENROLL_COMPLETE – Enrolled devices for iPhone and WebOS
- INFECTED – Virus Infected devices
- LOST – Lost devices
- RETIRED – Retired devices
- VERIFIED – Registration Verified devices
- VERIFICATION_PENDING – Verification Pending devices
- EXPIRED – Expired devices
- WIPED – Wiped devices

statusCode:

- ACTIVE - 97
- BLOCKED - 98
- IENROLL_VERIFIED - 100
- IENROLL_INPROGRESS - 101
- IENROLL_COMPLETE - 102
- INFECTED - 105
- LOST - 108
- RETIRED - 114
- VERIFIED - 118
- VERIFICATION_PENDING - 112
- EXPIRED - 120
- WIPED - 119

Compliance, quarantinedStatus, and blockReason values

The APIs which return information about a device include these fields:

- <compliance>
- <quarantinedStatus>
- <blockReason>

The value of each of these fields appears in the response as a decimal number that represents a bitmap value. Each bit in the value represents a reason why the device is not in compliance, has been quarantined, or has been blocked from accessing the ActiveSync server.

The following table shows when each of these fields is non-zero:

Field	Value
<compliance>	The value is non-zero if the device is not in compliance as specified by its security policy.
<quarantinedStatus>	The value is non-zero if the device is not in compliance as specified by its security policy, and the setting that is not in compliance specifies an action that includes quarantining the device.
<blockReason>	The value is non-zero in the following cases: • The device is not in compliance as specified by its security policy, and the setting that is not in compliance specifies an action that includes blocking access to the ActiveSync server. • The administrator has manually blocked the device's access to the ActiveSync server. This action is available in MobileIron Core Admin Portal, at Users & Devices | ActiveSync Associations.

Note: If multiple reasons apply, the corresponding bit values are summed. For example, if the device has been compromised (value 1), and its OS version is less than the required version (value 2), then the field has the value 3.

The following table shows all the possible bitmap values for the <compliance>, <quarantinedStatus>, and <blockReason> fields:

Hexadecimal value	Decimal value	Description
0x000000	0	Device is in compliance. Note: Jailbroken Android devices have the compliance value 0, but the security_ state field in the Android device details has the value "compromised".
0x000001	1	Device is compromised.
0x000002	2	OS version is less than the supported OS version.

Hexadecimal value	Decimal value	Description
0x000004	4	Hardware version is not allowed.
0x000008	8	Data Protection is not enabled.
0x000020	32	Device is out of contact.
0x000040	64	App control policy is out of compliance.
0x000080	128	Device exceeds per mailbox limit.
0x000100	256	Device is not registered.
0x000200	512	Device is manually blocked.
0x000400	1024	Exchange Reported.
0x000800	2048	Device administrator is deactivated. Note: On an Android device, the device administrator is deactivated. On iOS 5.0 and higher, the MDM profile has been removed, which deactivates MDM on the device.
0x001000	4096	Disallowed app control policy is out of compliance.
0x002000	8256	Required app control policy is out of compliance.
0x000010	8272	Policy is out of date.
0x004000	16384	Allowed app control policy is out of compliance.
0x008000	32768	User logged out.
0x10000	65536	Attestation Failed.
0x400000	4194304	Unknown reason.

Get Devices by Status

A device within the MobileIron system travels through a variety of different states, each of which can be retrieved through an API call. States such as enrollment-in-progress, active, retired, lost, or wiped can be retrieved. This API returns all devices that match the requested status type. If the status is not specified, all devices with 'Active' status are returned.

Examples:

Get all devices with the status ACTIVE:

```
https://mycore.mobileiron.com/api/v1/dm/devices
```

Get all devices with the status LOST:

```
https://mycore.mobileiron.com/api/v1/dm/devices?status=LOST
```

URI: https://{host-name}/api/v1/dm/devices	Devices with the input status are returned.
Http Method:	GET
Format:	xml, json
Request:	
Status	Optional. See list of valid values above.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<deviceManagementWebServiceResponse>	
<messages>	
<message>212 Device(s) returned</message>	Status message.
</messages>	
<devices>	
<device id='2'>	
<uuid>8d711cdc-e93c-49b1-88d6-222f54132445</uuid>	Unique identifier for the device.
<principal>jdoe</principal>	User ID for the user of the device. This corresponds to the user ID in the MobileIron Core Admin Portal, as seen in Users & Devices | Users.
<blockReason>0</blockReason>	A bitmap value that lists the reasons, if any, that the device is blocked from accessing the

	ActiveSync server. The possible values are described in Compliance, quarantinedStatus, and blockReason values
<clientId>1073741831</clientId>	For MobileIron Core internal use.
<comment>comment for the device</comment>	Comment entered by the administrator.
<compliance>0</compliance>	A bitmap value that lists the reasons, if any, that the device is out of compliance with its security policy. The possible values are described in Compliance, quarantinedStatus, and blockReason values.
<countryCode>1</countryCode>	Country code for the device.
<countryId>183</countryId>	Country identifier for the device. MOBILEIRON CORE assigns this identifier to the country.
<countryName>United States</countryName>	Country name for the device.
<details>	Device details, which consist of key-value pairs. The set of key-value pairs vary by the make, model, and operator of the device. The set shown is only an example. For more information, see Android Details Key-Value Descriptions and iOS Details Key-Value Descriptions. If device registration is pending, then the details section is empty.
<entry>	
<key>device_model</key>	
<value>DROIDX</value>	
</entry>	
<entry>	

<key>platform_name</key>	
<value>2.2</value>	
</entry>	
<entry>	
<key>Client_version</key>	
<value>4.2.0</value>	
</entry>	
</details>	
<deviceCount>1</deviceCount>	Not used. Always 0.
<easLastSyncAttempt>2012-01-10T20:36:57+00:00</easLastSyncAttempt>	Time of the last attempt the device made to synchronize with Exchange ActiveSync.
<easUuid>4d22d6d7-29dc-4c35-8e67-23dee442cf85</easUuid>	Exchange ActiveSync device id.
<emailAddress>jdoe@mobileiron.com</emailAddress>	The user's email address as entered during registration.
<emailDomain>txt.att.net</emailDomain>	Email domain of the operator for the device.
<employeeOwned>false</employeeOwned>	true - the employee owns the device. false - the enterprise owns the device. The value is set during registration and the administrator can change it.
<homeOperator>Verizon</homeOperator>	The service operator for the device when it is not roaming.
<languageCountryId>0<languageCountryId>	The unique identifier for the country associated with the language used on the device. For example, there would be a different ID for a Canadian French language device when compared to a device from France.

	MobileIron Core assigns this identifier to the country.
<languageId>1</languageId>	The unique identifier for the language used on the device.
<lastConnectedAt>2011-07-08T01:52:33+00:00</lastConnectedAt>	The date and time that the device last made successful contact with the MobileIron server. For iOS devices that have iOS MDM enabled, this value is the time of the last iOS MDM checkin.
<manufacturer>Research In Motion</manufacturer>	The device manufacturer as automatically reported by the device during registration.
<mdmManaged>false</mdmManaged>	Indicates that the MDM profile is enabled on the device. This field applies only to iOS devices. For other devices, the value is always false.
<mdmProfileUrlId></mdmProfileUrlId>	MOBILEIRON CORE internal ID for its iOS MDM profile information.
<model>8130</model>	The model of the device as reported by the device during registration.
<name>jdoe:Android 4.4:PDA 2</name>	The concatenated name used to identify the device/user combination.
<notifyUser>true</notifyUser>	true indicates the user should be notified via email during registration. This does not control whether MobileIron Core sends an SMS message given a valid phone number, which it always does. false indicates the user should not be notified. The notification consists of the principal name, platform, and phone number.
<operator>AT&T</operator>	Service provider for the device. The value PDA indicates no operator is associated with the device.

<operatorId>269</operatorId>	Identifier of the operator for the device. MOBILEIRON CORE assigns this identifier to the operator.
<phoneNumber>4085551212</phoneNumber>	The phone number entered by the user or administrator during registration.
<pin>2732E6DB</pin>	Unique identification number for a BlackBerry device. Not available for other devices.
<platform>Android 4.4</platform>	String indicating the platform installed on the device. The string is specified during registration.
<quarantinedStatus>0</quarantinedStatus>	A bitmap value that lists the reasons, if any, that the device is quarantined. When a device is quarantined, its configurations (that is, profiles) have been removed due to violations with its security policy. The possible values are described in Compliance, quarantinedStatus, and blockReason values.
<regCount>0</regCount>	For Blackberry, after the MobileIron client is downloaded, the VSP sends the provisioning SMS message to the client. If the client fails to connect, then the VSP resends the message at a scheduled interval. This value indicates how many times the VSP sent the provisioning message to the client.
<regType>DEFAULT</regType>	This value applies only to BlackBerry devices, indicating the registration type configured on MobileIron Core. Possible values are: DEFAULT: Register/Deploy via MobileIron BES: Register via MobileIron, Deploy via BES BESAUTO: Register/Deploy via BES 5.x
<status>ACTIVE</status>	String indicating the current status of the device with regard to registration and connection. For valid values, see Status field above.

<statusCode>97</statusCode>	Numeric code defined for the status. See list of valid values above.
<userDisplayName>Joe Doe</userDisplayName>	The concatenation of the user's first name and last name as defined during registration.
<userFirstName>Joe</userFirstName>	
<userLastName>Doe</userLastName>	
<userSource>76</userSource>	Value 76 for a Local user. Value 68 for an LDAP user. Note: 76 is the value of ASCII 'L', which stands for Local. 68 is the value of ASCII 'D', which stands for Directory (LDAP).
<userUUID>de398fcb-a3a4-412c-a1dd-9be8bd46e728</userUUID>	Internal user ID.
<iPhoneVersion>8J2</iPhoneVersion>	Version number of iPhone.
</device>	
</devices>	
</deviceManagementWebServiceResponse>	

Get Device details for a specific device

Example:

```
https://mycore.mobileiron.com/api/v1/dm/devices/12849438-0d74-3c30-6b7d-121a3da8645d
```

URI: https://{host-name}/api/v1/dm/devices/{deviceuuid}	Devices with the input status are returned.
Http Method:	GET

Format:	xml, json
Request:	
Deviceuuid	Unique identifier for the device. Only one uuid can be passed at a time.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<deviceManagementWebServiceResponse>	
</messages>	Empty if the device specified exists. Otherwise, a <message> element is included indicating that no device was found.
<device id='2'>	
<uuid>8d711cdc-e93c-49b1-88d6-222f54132445</uuid>	Unique identifier for the device.
<principal>jdoe</principal>	User ID for the user of the device. This corresponds to the user ID in the MobileIron Core Admin Portal, as seen in Users & Devices | Users.
<blockReason>0</blockReason>	A bitmap value that lists the reasons, if any, that the device is blocked from accessing the ActiveSync server. The possible values are described in The possible values are described in Compliance, quarantinedStatus, and blockReason values.
<clientId>1073741831</clientId>	For MOBILEIRON CORE internal use.
<compliance>0</compliance>	A bitmap value that lists the reasons, if any, that the device is out of compliance with its security policy. The possible values are described in Compliance, quarantinedStatus, and blockReason values.

<countryCode>1</countryCode>	The country code for the device.
<details>	Device details, which consist of key-value pairs. The set of key-value pairs vary by the make, model, and operator of the device. The set shown is only an example. For more information, see Android Details Key-Value Descriptions and iOS Details Key-Value Descriptions. If device registration is pending, then the details section is empty.
<entry>	
<key>device_model</key>	
<value>DROIDX</value>	
</entry>	
<entry>	
<key>platform_name</key>	
<value>2.2</value>	
</entry>	
<entry>	
<key>Client_version</key>	
<value>4.2.0</value>	
</entry>	
</details>	
<currentPhoneNumber>4085551212</currentPhoneNumber>	The phone number entered by the user or administrator during registration.
<deviceCount>1</deviceCount>	Not used. Always 0.

<emailAddress>jdoe@mobileiron.com</emailAddress>	The user's email address as entered during registration.
<employeeOwned>false</employeeOwned>	true - the employee owns the device. false - the enterprise owns the device. The value is set during registration and the administrator can change it.
<languageCountryId>0<languageCountryId>	The unique identifier for the country associated with the language used on the device. For example, there would be a different ID for a Canadian French language device when compared to a device from France. MobileIron Core assigns this identifier to the country.
<languageId>1</languageId>	The unique identifier for the language used on the device.
<lastConnectedAt>2011-07-08T01:52:33+00:00</lastConnectedAt>	The date and time that the device last made successful contact with the MobileIron server. For iOS devices that have iOS MDM enabled, this value is the time of the last iOS MDM checkin.
<manufacturer>Research In Motion</manufacturer>	The device manufacturer as automatically reported by the device during registration.
<mdmManaged>false</mdmManaged>	Indicates that the MDM profile is enabled on the device. This field applies only to iOS devices. For other devices, the value is always false.

<mdmProfileUrlId></mdmProfileUrlId>	MOBILEIRON CORE internal ID for its iOS MDM profile information.
<model>8130</model>	The model of the device as reported by the device during registration.
<namejdoe:Android 4.4:PDA 2</name>	The concatenated name used to identify the device/user combination.
<notifyUser>true</notifyUser>	true indicates the user should be notified via email during registration. This does not control whether MobileIron Core sends an SMS message given a valid phone number, which it always does. false indicates the user should not be notified. The notification consists of the principal name, platform, and phone number.
<operator>PDA</operator>	Service provider for the device. The value PDA indicates no operator is associated with the device.
<operatorId>269</operatorId>	Identifier of the operator for the device. MOBILEIRON CORE assigns this identifier to the operator.
<platform>Android 4.4</platform>	String indicating the platform installed on the device. The string is specified during registration.
<quarantinedStatus>0<quarantinedStatus>	A bitmap value that lists the reasons, if any, that the device is quarantined. When a device is quarantined, its configurations (that is, profiles) have been removed due to violations with its security policy.

	The possible values are described in Compliance, quarantinedStatus, and blockReason values.
<regCount>0</regCount>	For Blackberry, after the MobileIron client is downloaded, the VSP sends the provisioning SMS message to the client. If the client fails to connect, then the VSP resends the message at a scheduled interval. This value indicates how many times the VSP sent the provisioning message to the client.
<regType>DEFAULT</regType>	This value applies only to BlackBerry devices, indicating the registration type configured on the VSP. Possible values are: DEFAULT: Register/Deploy via MobileIron BES: Register via MobileIron, Deploy via BES BESAUTO: Register/Deploy via BES 5.x.
<status>ACTIVE</status>	String indicating the current status of the device with regard to registration and connection. For valid values, see Status field above.
<statusCode>97</statusCode>	Numeric code defined for the status. See list of valid values above.
<userDisplayName>Joe Doe</userDisplayName>	The concatenation of the user's first name and last name as defined during registration.
<userFirstName>Joe</userFirstName>	
<userLastName>Doe</userLastName>	
<userSource>76</userSource>	Value 76 for a local user. Value 68 for an LDAP user.

	Note: 76 is the value of ASCII 'L', which stands for Local. 68 is the value of ASCII 'D', which stands for Directory (LDAP).
<userUUID>de398fcb-a3a4-412c-a1dd-9be8bd46e728</userUUID>	Internal user ID.
</device>	
</deviceManagementWebServiceResponse>	

Android Details Key-Value Descriptions

The following table shows the key-value pairs in the <details> element for Android devices. The set of key-value pairs and the order they appear in the response can vary according to the type of device. Therefore, the table presents the pairs in alphabetical order by the key name.

If a key-value pair is not applicable for a device, typically the HTTP response does not include the pair.

The MobileIron Core Admin Guide has more information about fields that are available in the Admin Portal.

Key Name	Key Description	Value
admin_ activated	Whether device administrator privilege is activated for the MobileIron client on the device.	true false
battery_life	Power remaining in the battery life.	The percentage of power remaining in the battery. Example: 100
board	The name of the underlying board on the Android device.	A name that the Android OS provides. Example: venus2
brand	The brand (e.g., carrier) the Android software is customized for, if any.	A string that the Android OS provides.

Key Name	Key Description	Value
		Example: verizon
c2dmToken	Android C2DM registration ID for the device.	A string of characters
client_name	Name of MobileIron client application on the device.	Example: com.mobileiron
client_ version	MobileIron client version number running on the device.	Example: 4.5.0
codename	Android platform's current development codename, or the string "REL" if this is a release build.	Example: REL
country_ code	The device's Mobile Country Codes (MCCs). MCCs are defined in ITU E.212.	Example for United States: 310
current_ mobile_ number	Phone number of the device	Example: 6505551212
current_ operator_ name	Name of current registered operator.	Example: Verizon Wireless
current_ SIM_ module_ number	International Mobile Subscriber Identity number for the device.	Example: 262014530204577
device	The name of the industrial design of the device.	A string that the Android OS provides. Examples: cdma_droid2 cdma_shadow
device_id	Unique identifier for the device	Example: ddc865b69c13eeb4
device_ manufacturer	Manufacturer of the device.	Example: motorola
device_ model	Model of the Android device	Example: DROID2

Key Name	Key Description	Value
device_ roaming_flag	Whether the device is roaming.	on – The device is roaming. off -- The device is not roaming.
device_type	Whether the device uses CDMA or GSM technology to transmit voice calls. If the device does not transmit voice calls, this fields whether the device uses CDMA or GSM technology is transmit data.	CDMA or GSM
display_size	Size of the device's display	Dimensions in pixels, in the format: <height>X<width> Example: 854X480
free_media_ card_size	Amount of unused storage on the media card on the device.	Number in bytes Example: 2.36M
free_media_ card_size_ bytes	Amount of unused storage on the media card on the device.	Number in bytes Example: 104857000
free_ram_ size	Amount of RAM available on the device.	Number of megabytes, shown with M suffix. Example: 5.84M
free_ram_ size_bytes	Amount of unused RAM memory on the device.	Number in bytes. Example: 104857000
free_ storage_size	Amount of unused storage on the device	Number in bytes Example: 6489.68M
free_ storage_ size_bytes	Size of unused storage on the device.	Number in bytes. Example: 104857000
home_ operator	Home service provider for the device	The service provider name, mobile country code and mobile network code of the provider in the following

Key Name	Key Description	Value
		format: <name>::<MCC+MNC> Example: Verizon::310004
imei	International Mobile Equipment Identity of the device.	Example: A00000226EBF9F
imsi	International Mobile Subscriber Identity number for the device.	Example: 262018410218015
incremental	Android platform version's build number.	Example: 110719
lat_long_ last_ captured_at	The last time the location of the device was recorded.	Specified as seconds since January 1, 1970. Example: 1324421860972
latitude	Latitude of the device's location.	Degrees latitude. Example: 37.396074
locale	Locale for the device	Examples: en-US en
longitude	Longitude of the device's location.	Degrees longitude Example: -122.056339
mdm_ enabled	Whether the MobileIron client is fully configured on the device. Note: The MobileIron client can be installed and running, but still unable to manage the device if it is not fully configured.	true – The MobileIron client is fully configured. false – The MobileIron client is not fully configured.
multi_mdm	Whether multiple Device Admin applications are active on the device.	true – More than one Device Admin application are active. False – One or zero Device Admin Applications are active.

Key Name	Key Description	Value
network_id	CDMA network identification number.	Example: 6
os_version	The Android SDK version code	Example: 10 The value 10 corresponds to Android 2.3.3. Values are defined on http://developer.android.com.
platform_ name	Android platform version number on the device.	Example: 2.3.3
processor_ architecture	Processor architecture of the device.	armeabi-v7a
prv_ bluetooth	Whether the lockdown policy for the device has disabled access to Bluetooth.	ON – Access to Bluetooth is enabled for both audio and data. AUDIO – Access to Bluetooth is enabled for audio only. OFF – Access to Bluetooth is disabled. unsupported – The MobileIron client does not support enabling or disabling Bluetooth on the device.
prv_camera	Whether the lockdown policy for the device has disabled access to the camera.	ON – Access to the camera is enabled. OFF – Access to the camera is disabled. unsupported – The MobileIron client does not support enabling or disabling the camera on the device.
prv_device_ encryption	Whether the security policy for the device has enabled data encryption on the device.	on – Device encryption is enabled. off -- Device encryption is not enabled. unsupported – The MobileIron client

Key Name	Key Description	Value
		does not support enabling or disabling data encryption on the device.
prv_ exchange_ Domain	Domain of the email server of the device's user.	Email server domain. For example: MOBILEIRON If the email client is not yet configured, the value is na. If the email client is not supported by MobileIron, then the response does not include this key-value pair.
prv_ exchange_ Server	Email server for the device's user.	Email server address. For example: mail.mobileiron.com If the email client is not yet configured, the value is na. If the email client is not supported by MobileIron, then the response does not include this key-value pair.
prv_ exchange_ UserName	Email user name of the device's user.	Email user name. For example: jdoe@mobileiron.com If the email client is not yet configured, the value is na. If the email client is not supported by MobileIron, then the response does not include this key-value pair.
prv_ exchange_ UseSSL	Whether email transport uses Secure Socket Layer.	ON – Email uses the Secure Socket Layer. The value is ON if MobileIron supports the email client and the email client is configured. If the email client is not yet

Key Name	Key Description	Value
		configured, the value is na. If the email client is not supported by MobileIron, then the response does not include this key-value pair.
prv_max_ failed_ attempts	Maximum number of times the user can enter an incorrect password before the device is wiped.	The maximum number, or the value 0 if no maximum exists. This value is applicable only if prv_ password_type indicates that a password is mandatory.
prv_max_ idle_time	Maximum time the device can be inactive before the user must re-enter the password.	Number of minutes Example: 30 This value is applicable only if prv_ password_type indicates that a password is mandatory.
prv_ password	Whether both of the following conditions are true: 1. A password is mandatory for the user to access the device, as specified in the device's security policy. 2. The device is compliant with the security policy.	ON – Both conditions are true. OFF – One or both of the conditions are not true.
prv_ password_ expiration_ timeout	Numbers of days after which the device's password will expire.	The number of days, or the value unsupported if a password is optional. Example: 30 This value is applicable only if prv_ password_type indicates that a password is mandatory.
prv_ password_ history_ length	Number of passwords remembered to ensure that the device's user define a different password.	A number, or the value unsupported if a password is optional. This value is applicable only if prv_ password_type indicates that a

Key Name	Key Description	Value
	For example, the value 4 prevents the user from repeating a password for the next four password changes.	password is mandatory.
prv_ password_ length	Minimum length for the device's password.	Number between 1 and 10, or -1 which indicates the password has no minimum length. This value is applicable only if prv_ password_type indicates that a password is mandatory.
prv_ password_ minimum_ symbols	Minimum number of special characters that must be included in a password. Applicable only to Android 3.0 and higher.	A number or the value unsupported if no minimum is required. This value is applicable only if prv_ password_type indicates that a password is mandatory.
prv_ password_ type	Whether the device's password is mandatory, and whether it must be restricted to simple numeric input, alphanumeric characters, or has no restrictions. The security policy assigned to the device specifies the password type.	0 – password is mandatory and is restricted to alphanumeric characters. 1 – password is mandatory and is restricted to simple numeric characters. 2 – password is mandatory and has no character restrictions. -1 – password is optional.
prv_sd_ encryption	Whether the security policy for the device has enabled encrypting the contents of the SD (Secure Data card) on the device.	on – SD encryption is enabled. off -- SD encryption is not enabled. unsupported – The MobileIron client does not support enabling or disabling SD encryption on the device.
prv_sdcard	Whether the lockdown policy for the device has disabled access to the SD card.	ON – Access to the SD card is enabled.

Key Name	Key Description	Value
		OFF – Access to the SD card is disabled. unsupported – A lockdown policy is not applied to this device.
prv_vpn_ servers	A list of VPN servers that the device can access.	List of semi-colon-separated VPN servers, each given as an IP address, a host name, or a URL. The value is na if the list is empty.
prv_wifi	Whether the lockdown policy for the device has disabled access to wireless LANs.	ON – Access to wireless LANs is enabled. OFF – Access to wireless LANs is disabled. unsupported – The MobileIron client does not support enabling or disabling access to wireless LANs on the device.
prv_wlan_ ssids	Wireless local area network Service Set Identifiers for all wireless LANs configures on the device.	List of identifiers, separated by semi-colons. If none, then the value is na. Example: MobileIron-Guest;MobileIron-Test
registration_ imsi	International Mobile Subscriber Identity number for the device.	Example: 262073991646313
registration_ operator_ name	The name of the service provider for the device.	Example: Verizon
regUuid	Device's unique ID.	Example: ddc865b69c13eeb4
Samsung_ DM	Samsung device information for Samsung devices that support Samsung MDM APIs.	Example: FW: Key2,1 SW:1.0

Key Name	Key Description	Value
security_ state	Indicates whether the device has been compromised. A compromised Android device means that the device has been rooted, which means that an application has root access to the device's file system.	Ok – The device has not been compromised. Compromised – The device has been compromised.
SIM_ module_ number	International Mobile Subscriber Identity number for the device.	Example: IMSI:3104105000000000
system_id	CDMA System Identification number	Example: 40
total_media_ card_size	Amount of storage on the media card on the device.	Number of megabytes, shown with M suffix. Example: 7574.19M
total_media_ card_size_ bytes	Amount of storage on the media card on the device.	Number in bytes Example: 785037745
total_ram_ size	Amount of RAM memory on the device.	Number of megabytes, shown with M suffix. Example: 475.93M
total_ram_ size_bytes	Amount of RAM memory on the device.	Number in bytes Example: 504857000
total_ storage_size	Amount of storage on the device.	Number of megabytes, shown with M suffix. Example: 6700.98M
total_ storage_ size_bytes	Amount of storage on the device.	Number in bytes Example: 104857000
usb_ debugging	Indicates whether USB debugging is enabled on the device.	ON – USB debugging is enabled. OFF – USB debugging is disabled.

Key Name	Key Description	Value
wifi_mac_ addr	Wi-Fi MAC address of the device.	Example: f87b7a29838f

iOS Details Key-Value Descriptions

The following table shows the key-value pairs in the <details> element for iOS devices. The set of key-value pairs and the order they appear in the response vary according to the type of device, such as iPhone or iPad. Therefore, the table presents the pairs in alphabetical order by the key name.

Note: In most cases, key names that have an underscore, such as security_state or Client_build_date, contain information that the device's MobileIron client provides. Key names without underscores, such as allowUntrustedTLSPrompt or maxGracePeriod, contain information that the device's operating system provides.

Key Name	Key Description	Value
allowAppInstallation	Whether installation of applications is allowed.	Example: false
allowCloudBackup	Whether backing up the device to iCloud is allowed. Availability: iOS 5.0 and later.	true – Backing up to iCloud is allowed. false – Backing up to iCloud is not allowed.
allowCloudDocumentSync	When false, document and key-value syncing to iCloud is disabled.	Example: false
allowExplicitContent	Whether explicit music or video content	true – Explicit content is not hidden. false – Explicit content is hidden.

Key Name	Key Description	Value
	purchased from the iTunes Store is hidden. Content is marked as explicit by content providers when sold through the iTunes Store.	
allowInAppPurchases	Whether In-App purchases are allowed.	true – In-App Purchases are allowed. false - In-App Purchases are not allowed.
allowiTunes	Whether the iTunes Music Store is allowed on the device.	true – iTunes is allowed. false - iTunes is not allowed.
allowMultiPlayerGaming	Whether multiplayer gaming is allowed.	true – Multiplayer gaming is allowed. false - Multiplayer gaming is not allowed.
allowPhotoStream	Indicates whether the device's Photo Stream is allowed on the device. Availability: iOS 5.0 and later.	true – Photo Stream is allowed. false – Photo Stream is not allowed.
allowUntrustedTLSPrompt	When false, automatically rejects untrusted HTTPS certificates	true or false

Key Name	Key Description	Value
	without prompting the user. Availability: iOS 5.0 and later.	
allowVideoConferencing	Whether videoconferencing is allowed on the device.	true - Videoconferencing is allowed. false – Videoconferencing is not allowed.
allowVoiceDialing	Whether voice dialing is allowed when the device is locked.	true – Voice dialing is allowed when the device is locked. false - Voice dialing is not allowed when the device is locked.
allowYouTube	Whether the YouTube application is allowed on the device.	true - YouTube is allowed. false – YouTube is not allowed.
apnsToken	The device's APNs (Apple Push Notification service) token.	Example: 5c7b0866d6d068f8b4015690b83a6d1c00fb9484bdb00ea40d926bbade28de5f
AvailableDeviceCapacity	Floating-point gibibytes (base-1024 gigabytes).	Example: 13.765106201171875
Battery Level	Floating-point percentage expressed as a value between 0.0 and 1.0, or -	Example: 0.10000000149011612

Key Name	Key Description	Value
	1.0 if battery level cannot be determined. Availability: iOS 5.0 and later.	
battery_life	Power remaining in the battery life.	The percentage of power remaining in the battery. Example: 30
BluetoothMAC	Bluetooth MAC address.	Example: B8FF617F7927
BuildVersion	The iOS build number (8A260b, for example).	Example: 8J3
CarrierSettingsVersion	Version of the currently-installed carrier settings file.	Example: 11.0
CellularTechnology	Returns the type of cellular technology. Availability: iOS 4.2.6 and later.	Example: GSM
CheckOut Received	MobileIron Core has received a checkout message from the device. This message indicates that the MDM profile was removed	true – MobileIron Core has received a checkout message. false – MobileIron Core has not received a checkout message.

Key Name	Key Description	Value
	from the device.	
Client_build_date	Build date of the MobileIron client.	Example: Apr 8 2011 12:02:24
client_name	Name of MobileIron client application on the device.	Example: com.mobileiron.phoneatwork
Client_version	MobileIron client version number running on the device.	Example: 4.5.12.33698
country_code	The device's Mobile Country Codes (MCCs). MCCs are defined in ITU E.212 .	Example for United States: 310
Current MCC	The device's Mobile Country Codes (MCCs). MCCs are defined in ITU E.212 .	Example for United States: 310
Current MNC	Current Mobile Network Code. If the device is not roaming, this is the same as the SIM MNC.	Example: 00

Key Name	Key Description	Value
DataRoamingEnabled	Whether Data Roaming is enabled.	Example: false
device_id	The International Mobile Equipment Number for an iPhone.	Example: IMEI:012537000804721
device_manufacturer	Device manufacturer. For iOS devices, the value is always Apple.	Example: Apple
device_model	Model of the iOS device.	Examples: iPad iPhone 4
device_type	Whether the device uses CDMA or GSM technology to transmit voice calls. If the device does not transmit voice calls, this field indicates whether the device uses CDMA or GSM technology to transmit data.	CDMA or GSM Example: GSM
DeviceCapacity	Floating-point gibibytes (base-	Example: 14.020126342773438

Key Name	Key Description	Value
	1024 gigabytes).	
DeviceCompromised	Whether the device is compromised.	true – The device is compromised. false – The device is not compromised.
DeviceName	The name given to the device via iTunes.	Example: Joe B's iPad
forceEncryptedBackup	Whether the device forces encrypted backups.	true or false
free_storage_size_byte	Size of unused storage on the device.	Number in bytes. Example: 14780170240.000000000000000
HardwareEncryptionCaps	Describes the underlying hardware encryption capabilities of the device.	The value represents a bit field with following meanings: 1 – block-level encryption 2 – file-level encryption Therefore, because these are bit field values, the value 3 means both block-level and file-level encryption.
imei	The device's IMEI number. Ignored if the device does not support GSM.	Example: 011981001429081
ImeiOrMeid	The device's MEID number. Ignored if the device does not support CDMA.	Example: 01 198100 142908 1

Key Name	Key Description	Value
imsi	International Mobile Subscriber Identity number for the device.	The IMSI or the value na if the device has no IMSI. Example: 262073947704030
iOSBackgroundStatus	The status of background location multitasking on the device.	0 – The device supports background location multitasking, and the user has enabled location services. 1 – The device supports background location multitasking, but the user has disabled location services. 2 – Background multitasking has been disabled by the privacy policy applied to the device. 3 – The device hardware does not support background multitasking. 4 – The iOS version is earlier than 4.0, and therefore does not support background multitasking. Example: 3
ip_address	IP address of the device.	Example: 192.168.1.174 The response includes this field only if the device had connected to a WIFI network. However, this field does not indicate whether the device is currently connected to a WIFI network.
iPhone ICCID	The ICC identifier for the installed SIM card.	Example: 8949 2260 7349 2040 105
iPhone IMEI	International Mobile Equipment Identity of the device.	Example: 01 253700 080472 1
iPhone MAC_ADDRESS_EN0	WIFI MAC address of	Example: b8:ff:61:7f:79:26

Key Name	Key Description	Value
	device.	
iPhone PRODUCT	The model code for the device.	Examples: iPad iPhone 4
iPhone UDID	The unique device identifier (UDID) of the iOS device.	Example: 81a3379d884f1bd9f1b0ce9b340358288081f7a1
iPhone VERSION	The iOS build number of the iOS version that the device is running.	Example: 8J3
it_policy_result	Not used.	Not used.
lat_long_last_captured_at	The last time the location of the device was recorded.	Specified as seconds since January 1, 1970. Example: 1325108114776
latitude	Latitude of the device's location.	Degrees latitude. Example: 50.645397
locale	Locale for the device	Examples: en-US en
longitude	Longitude of the device's location.	Degrees longitude. Example: 7.943374
maxGracePeriod	Maximum grace period, in minutes, to	Example: 900

Key Name	Key Description	Value
	unlock the phone without entering a passcode. The value 0 means no grace period is allowed; a passcode is required immediately.	
maxInactivity	Number of minutes for which the device can be idle (without being unlocked by the user) before it gets locked by the system. Once this limit is reached, the device is locked and the passcode must be entered.	Example: 300
minLength	Minimum overall length of the passcode.	Example: 4
mobile_number	Phone number of the device.	The mobile number, or the value (null) if the device has no mobile number. Example: +491718169911
Model	The device's model number.	Examples:

Key Name	Key Description	Value
		MC820LL MC603DN
ModelName	Name of the device model.	Examples: iPad iPhone
ModemFirmwareVersion	The baseband firmware version.	Example: 05.16.05
os_version	The version of iOS that the device is running.	Example: iPhone OS 4.3.3 (8J3) iPhone OS 5.0.1 (9A405)
OSVersion	The version of iOS that the device is running.	Example: 4.3.3
PasscodeIsCompliant	Set to true if the user's passcode is compliant with all requirements on the device, including Exchange and other accounts.	true or false
PasscodeIsCompliantWithProfiles	Set to true if the user's passcode is compliant with requirements from profiles.	Example: true

Key Name	Key Description	Value
PasscodePresent	Set to true if the device is protected by a passcode.	true or false
platform_name	For all iOS devices, this field has the value iPhone.	Example: iPhone
platform_type	Either iPad or iPhone.	Examples: iPad iPhone
processor_architecture	For iOS devices, the value is always ARM.	Example: ARM
ProductName	The model code for the device.	Examples: iPad1,1 iPhone3,1
ratingApps	Maximum rating for apps on the device, according to Apple's ranking of apps.	Example: 1000
ratingMovies	Maximum rating for movies on the device, according to Apple's ranking of movies.	Example: 1000
ratingTVShows	Maximum rating for TV shows	Example: 1000

Key Name	Key Description	Value
	on the device, according to Apple's ranking of TV shows.	
registration_imsi	International Mobile Subscriber Identity number for the device.	Example: (null)
registration_operator_ name	The name of the service provider for the device.	The name of the service provider, or (null) if not applicable. Example: AT&T
safariAcceptCookies	Indicates Safari's setting to accept cookies.	0 - Never 1 - From visited 2 - Always
safariAllowPopups	Indicates whether Safari is set to allow pop-ups.	true – popups are allowed. false – popups are not allowed.
safariForceFraudWarning	Indicates whether Safari is set to enable fraud warning.	true – Fraud warning is enabled. false – Fraud warning is not enabled.
security_reason_code	Not used.	Not used.
security_state	Indicates whether the device has been compromised.	0 – The device has been compromised. 1 – The device has not been compromised.

Key Name	Key Description	Value
SerialNumber	The device's serial number.	Example: V5046DGHZ38
signal_strength	The signal strength on the device.	A number representing the signal strength, given in dBm.
SIM MCC	Home Mobile Country Code (numeric string). MCCs are defined in ITU E.212 .	Example for United States: 310
SIM MNC	The Mobile Network Code of the SIM card on the device. Note: This field contains a numeric MNC only if the network is GSM. For CDMA networks, this field contains an abbreviation of the carrier name, such as VZW or SPR, for Verizon and Sprint.	Example: 01 07
SIMCarrierNetwork	Name of the home carrier network.	Example: Telekom.de
Subscriber Carrier Network	Name of the home carrier	Example: o2-de

Key Name	Key Description	Value
	network. (Replaces SIMCarrierNetwork.) Availability: iOS 5.0 and later.	
total_storage_size_bytes	Amount of storage on the device.	Number in bytes. Example: 15053996032.000000536870912
Voice Roaming Enabled	Whether Voice Roaming is enabled.	Example: true
WiFiMAC	Wi-Fi MAC address.	Example: B8FF617F7926

Exporting Device Information to a CSV

To export device information to a CSV, use the following URI:

https://{host-name}/api/v1/dm/devices.csv

Note: No support is available for exporting device information for Exchange ActiveSync (EAS) devices to a CSV. The request https://{host-name}/api/v1/eas/devices.csv is not supported.

The following fields are exported:

- Operator
- Country
- Device UUID
- Phone Number
- Principal
- Name
- Platform
- Manufacturer
- Model
- Email Address
- Status Code
- Employee Owned
- Compliance

- Quarantine Status
- IMSI
- IMEI
- UDID
- Client Version
- MDM Enabled
- Serial Number
- Last Connected At
- Active Sync UUID
- Active Sync Last Sync Attempt
- Wi-Fi MAC Address
- Device Encryption
- Last MDM Check-In
- Last Security State Changed On
- Registered On

Get Device Details for a Phone Number/User/Label/Wi-Fi MAC Address

Device details such as manufacturer, model, OS, status, and registered email address can be retrieved in multiple ways using an API. Search requests can be made by phone number, user ID, or label. A single user may be assigned multiple devices, in which case a list of devices could be returned for a matching user ID. Given a phone number in the request, the API returns the device details for the pairing of user and phone number. Given a label in the request, the API returns the device details for all devices assigned to that label. The details returned depend on what the device reports; different devices may return different information. This API applies only to registered devices.

Examples:

Get the device details for the device that has a specified phone number:

```
https://mycore.mobileiron.com/api/v1/dm/phones/4155551212
```

Get the device details for the devices that have the specified phone numbers:

```
https://mycore.mobileiron.com/api/v1/dm/phones/6505551212,4155551212
```

Get the device details for the devices belonging to the specified user:

```
https://mycore.mobileiron.com/api/v1/dm/users/jdoe
```

Get the device details for the devices assigned to a specific label:

```
https://mycore.mobileiron.com/api/v1/dm/labels/android
```

Get the device details for the device having a specific Wi-Fi MAC address:

```
https://mycore.mobileiron.com/api/v1/dm/devices/mac/38AA3C62BFAD
```

1. URI:	Device details of the input phone number is

https://{host-name}/api/v1/dm/phones/{phone#}	returned.
Http Method:	GET
Format:	xml, json
Request:	
phoneNumber	Required. This can be multiple, comma-separated phone numbers. Example: 4085551212,6505551212
2. URI: https://{host-name}/api/v1/dm/users/{username}	Device details of all devices registered to the input username will be returned.
Http Method:	GET
Format:	xml, json
Request:	
userName	Required. Device unique login user name.
3. URI: https://{host-name}/api/v1/dm/labels/{labelname}	Device details of all devices assigned to the input labelname will be returned.
Http Method:	GET
Format:	xml, json
Request:	
labelName	Required. Unique label name.
4. URI: https://{host-name}/api/v1/dm/devices/mac/{macaddress}	Device details of the device associated with the input Wi-Fi MAC address.
Http Method:	GET
Format:	xml, json

Request:		
macAddress	Required.	
Response Status Code:		
'404 – No Data Found'	There is no data.	
'200 – OK'	Data is present and the response is returned.	
Response:		
<deviceManagementWebServiceResponse>		
<phoneNumber>14085551212</phoneNumber>	Phone numbers from the request. Included in response in these two cases: - only one phone number was in the request. - the request specified more than one phone number, and a problem occurred so that no device details are included in the response. Note: If the request specified a user name or a label name, they are not repeated at the beginning of the response.	
<messages>		
<message>212 Device(s) returned</message>	Success is shown if the method execution is successful.	

	A descriptive error message is shown if the method execution failed. Note: If the request specified one valid phone number, the <messages> element is empty.
</messages>	
<devices>	
<device id='2'>	Device identifier.
<uuid>8d711cdc-e93c-49b1-88d6-222f54132445</uuid>	Unique identifier for the device.
<principal>jdoe</principal>	User ID for the user of the device. This corresponds to the user ID in the Admin Portal, as seen in Users & Devices | Users.
<blockReason>0</blockReason>	A bitmap value that lists the reasons, if any, that the device is blocked from accessing the ActiveSync server. The possible values are described in Compliance, quarantinedStatus, and blockReason values.
<clientId>1073741831</clientId>	For MOBILEIRON CORE internal use.
<compliance>0</compliance>	A bitmap value that lists the reasons, if any, that the device is out of compliance with its security policy. The possible values are described in Compliance, quarantinedStatus, and blockReason values.
<countryCode>1</countryCode>	The country code for the device.
<countryId>183</countryId>	Country identifier for the device. MOBILEIRON CORE assigns this identifier to the country.
<countryName>United States</countryName>	
<details>	Device details, which consist of key-value pairs. The set of key-value pairs vary by the make, model, and operator of the device. The set shown is only an example.

	For more information, see Android Details Key-Value Descriptions and iOS Details Key-Value Descriptions. If device registration is pending, then the details section is empty.
<entry>	
<key>total_ram_size</key>	
<value>109.74M</value>	
</entry>	
<entry>	
<entry>	
<key>device_model</key>	
<value>SGH-i617</value>	
</entry>	
<entry>	
<key>platform_name</key>	
<value>Windows Mobile 6.1 Standard</value>	
</entry>	
</details>	
<deviceCount>0</deviceCount>	Not used. Always 0.
<emailAddress>jdoe@mobileiron.com</emailAddress>	The user's email address as entered during registration.
<emailDomain>mydomain.com</emailDomain>	Not used at this time.
<employeeOwned>false</employeeOwned>	true - the employee owns the device. false - the enterprise owns the device.

	The value is set during registration and the administrator can change it.
<homeOperator>Verizon</homeOperator>	The service operator for the device when it is not roaming.
<languageCountryId>183</languageCountryId>	The unique identifier for the country associated with the language used on the device. For example, there would be a different ID for a Canadian French language device when compared to a device from France. MobileIron Core assigns this identifier to the country.
<languageId>1</languageId>	The unique identifier for the language used on the device.
<lastConnectedAt>2011-07-08T01:52:33+00:00</lastConnectedAt>	The date and time that the device last made successful contact with the MobileIron server. For iOS devices that have iOS MDM enabled, this value is the time of the last iOS MDM checkin.
<manufacturer>Research In Motion</manufacturer>	The device manufacturer as automatically reported by the device during registration.
<mdmManaged>false</mdmManaged>	Indicates that the MDM profile is enabled on the device. This field applies only to iOS devices. For other devices, the value is always false.
<mdmProfileUrlId></mdmProfileUrlId>	MOBILEIRON CORE internal ID for its iOS MDM profile information.
<model>8130</model>	The model of the device as automatically reported by the device during registration.
<name>jdoe:Android 4.4:PDA 2</name>	The concatenated name used to identify the device/user combination.
<notifyUser>true</notifyUser>	true indicates the user should be notified via email

	during registration. This does not control whether MobileIron Core sends an SMS message given a valid phone number, which it always does. false indicates the user should not be notified during registration. The notification consists of the principal name, platform, and phone number.
<operator>Verizon</operator>	Service provider for the device. The value PDA indicates no operator is associated with the device.
<operatorId>4195</operatorId>	Identifier of the operator for the device. MOBILEIRON CORE assigns this identifier to the operator.
<phoneNumber>4085551212</phoneNumber>	The phone number entered by the user during registration.
<platform>Android 4.4</platform>	String indicating the platform installed on the device. The string is specified during registration.
<quarantinedStatus>0</quarantinedStatus>	A bitmap value that lists the reasons, if any, that the device is quarantined. When a device is quarantined, its configurations (that is, profiles) have been removed due to violations with its security policy. The possible values are described in Compliance, quarantinedStatus, and blockReason values.
<regCount>0</regCount>	For Blackberry, after the MobileIron client is downloaded, the VSP sends the provisioning SMS message to the client. If the client fails to connect, then the VSP resends the message at a scheduled interval. This value indicates how many times the

	VSP sent the provisioning message to the client.
<regType>DEFAULT</regType>	This value applies only to BlackBerry devices, indicating the registration type configured on the VSP. Possible values are: DEFAULT: Register/Deploy via MobileIron BES: Register via MobileIron, Deploy via BES BESAUTO: Register/Deploy via BES 5.x.
registeredAt	Lists the date and time of device registration.
<status>ACTIVE</status>	String indicating the current status of the device with regard to registration and connection. See list of valid values above.
<statusCode>97</statusCode>	Numeric code defined for the status. See list of valid values above.
<userDisplayName>Joe Doe</userDisplayName>	The concatenation of the user's first name and last name as defined during registration.
<userFirstName>Joe</userFirstName>	
<userLastName>Doe</userLastName>	
<userUUID>de398fcb-a3a4-412c-a1dd-9be8bd46e728</userUUID>	Internal user ID.
</device>	
</devices>	
</deviceManagementWebServiceResponse>	

Register a Device

This API registers a device with MobileIron Core. Registering or enrolling a device designates it for management in MobileIron Core. The action of registering a device accomplishes the following:

- Activates a user associated with the device.
- Makes the device known to the MobileIron system.

- Downloads the MobileIron Client to the device
- Completes an initial scan of the device and synchronizes it to MobileIron Core.

NOTE: You can use this API with ServiceNow.

Examples:

https://mycore.mobileiron.com/api/v1/dm/register?phoneNumber=4155551212&userId=jdoe&platform=A&userFirstName=Joe&userLastName=Doe&userEmailAddress=jdoe@mobileiron.com&notifyUser=True&notifyuserbysms=True&countrycode=1&operator=Verizon

Notice in the following example that if the deviceType is PDA, then the phoneNumber value can be PDA:

```
https://mycore.mobileiron.com/api/v1/dm/register?phoneNumber=PDA&deviceType=PDA
&userId=miadmin&userFirstName=Jane&userLastName=Doe&userEmailAddress=jdoe@mi.com&countrycode=
33&importUserFromLdap=False&notifyUser=True&notifyuserbysms=True&platform=I
```

URI: https://{host-name}/api/v1/dm/register/	Register a device.
Http Method:	PUT
Format:	xml, json
Request:	
phoneNumber	Required for all device types except for PDA. If the device type is PDA, then you can enter PDA as the phone number value. This may improve performance over supplying a bogus phone number.
userId	Required.
operator	String indicating operator. This field will be updated after registration if MobileIron Core can find the operator based on the phoneNumber entry.
isEmployeeOwned	True indicates the device is owned by the employee. False indicates it is owned by

	the company. Default is false.
platform	Required. Platform or operating system of the device. Valid values: A- Android I – iOS E – Windows M – Windows Phone devices (WP8, WP8.1) L- Mac OS X
deviceType	Device type can be a phone or PDA. Valid values : Phone, PDA If device is a PDA, then phone number is optional and you can enter PDA as the phone number value, too.
importUserFromLdap	True – import the matching user from LDAP. False –create a local user. If a local user does not exist with the input userid, then a new local user is created. For local users, first name, last name, and email address are required.

	MobileIron Core sets the password for a new local user to the userid.
userFirstName	Required for local user. User's first name.
userLastName	Required for local user. User's last name.
userEmailAddress	Required for local user. User's email address.
notifyUser	True indicates user should be notified of registration by email. False indicates user should not be notified.
notifyuserbysms	True indicates user should be notified of registration by SMS. False indicates user should not be notified by SMS.
countryCode	Required. Country code of the operator.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<deviceManagementWebServiceResponse>	
<phoneNumber>14085551212</phoneNumber>	Phone number registered.
<registration>	
<messages/>	
<deviceUuid>caba40e7-f56b-44aa-ac70-79e32e91adf8</deviceUuid>	Alpha-numeric string that uniquely identifies the device.

<messages/>	Status Message. Success is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
<status>SUCCESS</status>	See "Status and statusCode values".
<passcode>63460</passcode>	5-digit numeric passcode needed during registration validation. If the passcode is not applicable for the operating system, it will be empty.
<passcodeTTL>120</passcodeTTL>	Number of hours the passcode is valid.
<registrationUrl>http://app16.mobileiron.com:8080/v/75b13</registrationUrl>	URL provided to the user. User enters a passcode to verify the device registration and the client begins to download.
</registration>	
</deviceManagementWebServiceResponse>	

Retire a Device

This API retires a device. Devices are retired based on a unique device ID (uuid).

NOTE: You can use this API with ServiceNow.

Examples:

```
https://mycore.mobileiron.com/api/v1/dm/devices/retire/ee8198d9-5d79-4961-94c4-
e21bf04b2467?Reason=User%20replaced%20device
```

```
https://mycore.mobileiron.com/api/v1/dm/devices/retire/mac/38AA3C62BFAD?Reason=User%20replaced%20device
```

1. URI: https://{host-name}/api/v1/dm/devices/retire/{deviceuuid}	Device with the input device uuid is retired
Http Method:	PUT
Format:	xml, json
Request:	
Device uuid	Required. Unique ID of the device. This ID can be retrieved in the response of other API calls, such as Device Registration or Get Device Details.
Reason	Free form text field (512 character limit) to display reason why the device is being retired.
2. URI: https://{host-name}/api/v1/dm/devices/retire/mac/{macaddress}	Device with the input Wi-Fi MAC address is retired.
Http Method:	PUT
Format:	xml, json
Request:	
macAddress	Required.
Reason	Free form text field (512 character limit) to display reason why the device is being retired.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<deviceManagementWebServiceResponse>	

<messages>	
<message> Device is retired successfully.</message>	Status Message. Success is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<deviceUuid>caba40e7-f56b-44aa-ac70-79e32e91adf8</deviceUuid>	Unique device ID.
< /deviceManagementWebServiceResponse>	

Lock a Device

This API locks a device, which typically forces the user to enter a passcode (either a user-generated or MobileIron-generated password) to access the device and prevents the user from reversing this restriction. Devices are locked based on unique device ID (uuid). As all mobile operating systems behave differently, refer to the Administration Guide for details on lock support.

NOTE: You can use this API with ServiceNow.

Examples:

```
https://mycore.mobileiron.com/api/v1/dm/devices/lock/ee8198d9-5d79-4961-94c4-e21bf04b2467?Reason=User%20lost%20device

https://mycore.mobileiron.com/api/v1/dm/devices/lock/mac/38AA3C62BFAD?Reason=User%20lost%20device
```

1. URI: https://{host-name}/api/v1/dm/devices/lock/{deviceuuid}	Device with the input device uuid is locked.
Http Method:	PUT
Format:	xml, json
Request:	
Device UUID	Required. Unique ID of the device. This ID is sent in the response of the Registration API.

Reason	Required. Free form text field (512 character limit) to display reason why the device is being locked.
1. URI: https://{host-name}/api/v1/dm/devices/lock/mac/{macaddress}	Device with the Wi-Fi MAC address is locked.
Http Method:	PUT
Format:	xml, json
Request:	
macAddress	Required.
Reason	Required. Free form text field (512 character limit) to display reason why the device is being locked.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<deviceManagementWebServiceResponse>	
<messages>	
<message>Device is locked successfully.</message>	Status Message. Success is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<deviceUuid>caba40e7-f56b-44aa-ac70-79e32e91adf8</deviceUuid>	Unique device ID.

	This field is not included for Android and iOS devices.
<unlockpasscode>12345</unlockpasscode>	Passcode to unlock the device. If this is missing, then the device was most likely locked with the user-set passcode. This field is not included for Android and iOS devices.
< /deviceManagementWebServiceResponse >	

Unlock a Device

This API unlocks a device.

On Android and iOS devices, unlocking the device clears its passcode.

On Blackberry devices, when a device without a user-generated passcode is locked, a special MobileIron passcode must be generated and shared with the user to unlock the device. A special passcode may be generated based on unique device ID (uuid). For those device, this API returns the unlock passcode.

Refer to the MobileIron Core Administration Guide for details on unlock support.

NOTE: You can use this API with ServiceNow.

Examples:

```
https://mycore.mobileiron.com/api/v1/dm/devices/unlock/ee8198d9-5d79-4961-94c4-e21bf04b2467?Reason=User%20forgot%20password
https://mycore.mobileiron.com/api/v1/dm/devices/unlock/mac/38AA3C62BFAD?Reason=User%20forgot%20password
```

1.URI: https://{host-name}/api/v1/dm/devices/unlock/{deviceuuid}	Unlock passcode for the device with the input device uuid is returned.
Http Method:	GET
Format:	xml, json
Request:	
Device UUID	Required. Unique ID of the device. This ID is sent in the

	response of the Registration API.
Reason	Required. Free form text field (512 character limit) to display reason why the device is being unlocked.
2.URI: https://{host-name}/api/v1/dm/devices/unlock/mac/{macaddress}	Unlock passcode for the device with the input device Wi-Fi MAC address is returned.
Http Method:	GET
Format:	xml, json
Request:	
macAddress	Required.
Reason	Required. Free form text field (512 character limit) to display reason why the device is being unlocked.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<deviceManagementWebServiceResponse>	
<messages>	
<message>1 passcode(s) sent.</message>	Status Message. Passcode count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	

<passcodes>	
<passcode>	
<uuid>cf667a65-1e1a-4121-af63-398b11540d2f</uuid>	Unique device ID.
<name> jdoe:Android:6505551212</name>	Username:Platform: phonenumber string to help distinguish between a user's multiple devices.
<value>6910</value>	For iOS devices: Q – The device is MDM managed and has a passcode. F – The device is MDM managed but has no passcode. NA – The device is not MDM managed. Unlock is not possible. For Android devices: Q – The device has a passcode. F – The device has no passcode. NA – Unlock failed. For Blackberry devices: Passcode to unlock the device. If value is empty, then the device was most likely locked with the user-set passcode.
</passcode>	
</passcodes>	
< /deviceManagementWebServiceResponse >	

Wipe a Device

This API wipes a device, which returns its settings to the factory defaults. Once wiped, device status changes to "Wiped," and the only valid action to apply is Retire. A wipe call is based on a unique device ID (uuid).

You can use this API with ServiceNow.

Example:

```
https://mycore.mobileiron.com/api/v1/dm/devices/wipe/ee8198d9-5d79-4961-94c4-
e21bf04b2467?Reason=Device%stolen
https://mycore.mobileiron.com/api/v1/dm/devices/wipe/38AA3C62BFAD?Reason= Device%stolen
```

1.URI: https://{host-name}/api/v1/dm/devices/wipe/{deviceuuid}	Device with the input device uuid is wiped.
Http Method:	PUT
Format:	xml, json
Request:	
Device UUID	Required. Unique ID of the device. This ID is sent in the response of the Registration API.
Reason	Free form text field (512 character limit) to display reason why the device is being wiped.
2.URI: https://{host-name}/api/v1/dm/devices/wipe/mac/{macaddress}	Device with the input device Wi-Fi MAC address is wiped.
Http Method:	PUT
Format:	xml, json
Request:	
macAddress	Required.
Reason	Free form text field (512 character limit) to display reason why the device is being wiped.

Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
< deviceManagementWebServiceResponse>	
<messages>	
<message></message>	Status Message. Success is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<deviceUuid>caba40e7-f56b-44aa-ac70-79e32e91adf8</deviceUuid>	Unique device ID.
< /deviceManagementWebServiceResponse>	

Wakeup Client

This API forces a device to connect to MobileIron Core, waking up the MobileIron Client. A wakeup call is based on a unique device ID (uuid).

NOTE: You can use this API with ServiceNow.

Examples:

```
https://mycore.mobileiron.com/api/v1/dm/devices/wakeup/ee8198d9-5d79-4961-94c4-e21bf04b2467
```

```
https://mycore.mobileiron.com/api/v1/dm/devices/wakeup/mac/38AA3C62BFAD
```

1.URI: https://{host-name}/api/v1/dm/devices/wakeup/{deviceuuid}	Request to wake up is sent to device with the input device uuid.
Http Method:	GET

Format:	xml, json
2.URI: https://{host-name}/api/v1/dm/devices/wakeup/mac/{macaddress}	Request to wake up is sent to device with the input Wi-Fi MAC address.
Http Method:	GET
Format:	xml, json
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
< deviceManagementWebServiceResponse>	
<messages>	
<message>Wake up request sent to device with uuid:cf667a65-1e1a-4121-af63-398b11540d2f</message>	Status Message. Success is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<deviceUuid>caba40e7-f56b-44aa-ac70-79e32e91adf8</deviceUuid>	Unique device ID.
< /deviceManagementWebServiceResponse>	

Locate a Device

The MobileIron Client periodically records cell tower location information. When this API is used, the last known location of the device is returned based on requested unique device ID (uuid). If needed, this API will remotely turn on a device's GPS. To find the current location of the device:

1. Call this API with locatenow=true. This will send a request to the device to determine the current location. This process might take between a few seconds and 1 minute.
2. Call the locate API again after a few minutes without the locatenow parameter . This will return the current location found from step 1. If the current location could not be determined it will return the last known location.

NOTE: You can use this API with ServiceNow.

Examples:

```
https://mycore.mobileiron.com/api/v1/dm/devices/locate/ee8198d9-5d79-4961-94c4-e21bf04b2467

https://mycore.mobileiron.com/api/v1/dm/devices/locate/mac/38AA3C62BFAD

https://mycore.mobileiron.com/api/v1/dm/devices/locate/ee8198d9-5d79-4961-94c4-
e21bf04b2467?locatenow=true
```

1.URI: https://{host-name}/api/v1/dm/devices/locate/{deviceuuid}	Location of the device with the input device uuid is returned.
Http Method:	GET
Format:	xml, json
Request:	
Device UUID	Required. Unique ID of the device. This ID is sent in the response of the Registration API.
locatenow	Optional. True or false. Defaults to false. See step 1 in the explanation above. This parameter does not apply to iOS.
2.URI: https://{host-name}/api/v1/dm/devices/locate/mac/{macaddress}	Location of the device with the input Wi-Fi MAC address is returned.
Http Method:	GET
Format:	xml, json
Request:	

macAddress	Required.
locatenow	Optional. True or false. Defaults to false. See step 1 in the explanation above. This parameter does not apply to iOS.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
< deviceManagementWebServiceResponse >	
<deviceUuid>cf667a65-1e1a-4121-af63-398b11540d2f</deviceUuid>	Unique Device ID.
<messages>	
<message></message>	Status Message. Success is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<locations>	
<location>	
<uuid>cf667a65-1e1a-4121-af63-398b11540d2f</uuid>	Unique Device ID.
<lookupResult>Cell</lookupResult>	Value returned describes how location was retrieved. Valid values: LookupFailure: unable to retrieve the location of the device. GPS: location retrieved using the GPS of the device. Cell: location retrieved using cell towers.

<latitude>37.386433</latitude>	Latitude of the location of the device.
<longitude>-122.053902</longitude>	Longitude of the location of the device.
<radius>1500</radius>	
<capturedAt>1285802663000</capturedAt>	Time when the location of the device was captured in epoch format.
</location>	
</locations>	
<locateNow>false</locateNow>	
< /deviceManagementWebServiceResponse>	

Enable Roaming

This API enables or disables voice and data roaming on an iOS 5 device. However, note the following:

- Voice roaming is available only on certain carriers. If you use this API to enable voice roaming on a device, the API returns success regardless of whether voice roaming is available on that device's carrier.
- If you disable voice roaming, you are also disabling data roaming, even if you specify true (enable) for the data roaming query parameter.
- The API returns success regardless of whether the device supports the setting. To see whether a device has data or voice roaming enabled, see the Voice Roaming Enabled and DataRoamingEnabled fields in the response to a Get Device API. See iOS Details Key-Value Descriptions.

Example:

```
https://mycore.mobileiron.com/api/v1/dm/devices/enableroaming/ee8198d9-5d79-4961-94c4-e21bf04b2467?voice=true&data=false
```

URI: https://{host-name}/api/v1/dm/devices/enableroaming/{deviceuuid}	The specified deviceuuid indicates the device on which to change roaming settings.
Http Method:	PUT
Format:	xml, json
Request:	
deviceuuid	Required. Unique ID of the iOS device. This ID can be retrieved in the response of other

	API calls, such as Get Devices by Status.
voice	Required. This parameter is a query parameter. Set to true to enable voice roaming. Set to false to disable voice roaming.
data	Required. This parameter is a query parameter. Set to true to enable data roaming. Set to false to disable data roaming. If you set the voice parameter to false, data roaming is disabled even if you set the data parameter to true.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<deviceManagementWebServiceResponse>	
<deviceUuid> 190eb32e-32e1-4fe2-baa1-06a4488aaa4c </deviceUuid>	
<messages>	
<message> Device voice roaming settings updated successfully. The voice roaming setting is available only on certain carriers.	Status message for voice roaming.

Disabling voice roaming also disables data roaming. </message>	Success is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
<message> Device data roaming settings updated successfully. </message>	Status message for data roaming. Success is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
< /deviceManagementWebServiceResponse>	

Get all Labels

Using labels is the method by which devices are grouped in the MobileIron database. Labels can be used for applying policies or performing other management tasks on multiple devices. An administrator can create labels in addition to a default set supplied in MobileIron Core. This API lists all labels, whether or not they are in use.

Example:

```
https://mycore.mobileiron.com/api/v1/dm/labels
```

URI: https://{host-name}/api/v1/dm/labels	All labels in the database are returned.
Http Method:	GET
Format:	xml, json
Response Status Code:	

'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
< deviceManagementWebServiceResponse >	
<messages>	
<message>1 Label (s) returned</message>	Status message. A label count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<labels>	
<label id="-3">	Internal database ID. Negative numbers correspond to the default labels. Positive numbers correspond to labels that MobileIron Core administrator added.
<name>iOS</name>	Label name.
<description>Label for all iOS devices.</description>	Label description.
<staticLabel>false</staticLabel>	Static labels are system created labels. False indicates a dynamic label. Devices which satisfy the criteria specified in <searchCriteria> are automatically added to this label. True indicates a static label, which has no <searchCriteria> Devices are manually assigned to static labels.

<query> "common.platform"="iOS" AND "common.retired"=false </query>	
<deviceCount>3</deviceCount>	The number of devices currently assigned to the label.
<isESSearch>116</isESSearch>	
<label>	
</labels>	
< /deviceManagementWebServiceResponse>	

List of Labels for a Device

A device may be applied to one or more labels. This API gets the list of all labels to which a unique device ID is assigned.

Example:

```
https://mycore.mobileiron.com/api/v1/dm/labels/devices/12849438-0d74-3c30-6b7d-121a3da8645d
```

URI: https://{host-name}/api/v1/dm/labels/devices/ {deviceuuid}	All labels assigned to uuid are returned.
Http Method:	GET
Format:	xml, json
Request:	
Device UUID	Required. Unique ID of the device. This ID is sent in the response of the Registration API.

Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
< deviceManagementWebServiceResponse >	
<messages>	
<message>1 Label (s) returned</message>	Status message. Label count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<labels>	
<label id="-3">	
<name>Android</name>	
<description>Label for all Android Phones.</description>	
<staticLabel>false</staticLabel>	
<deviceCount>0</deviceCount>	
<isESSearch>116</isESSearch>	
<label>	
</labels>	
< /deviceManagementWebServiceResponse>	

Apply Labels to a Device

Using labels is the method by which devices are grouped in the MobileIron database. Labels can be used for applying policies or performing other management tasks on multiple devices. MobileIron provides a set of default labels that you can apply to devices. You can also create your own labels using the Admin Portal (refer to the MobileIron Administration guide for instructions). Using this API, you can:

- apply a label to a device
- apply a label to multiple devices
- apply multiple labels to one device
- apply multiple labels to multiple devices

The API response contains error messages in these situations:

- The request contains an invalid device uuid.
- The request contains an invalid label.
- A label in the request is already applied to the device.

Examples:

Apply one label named TestLabel to one device:

```
https://mycore.mobileiron.com/api/v1/dm/labels/TestLabel/bdcbdf2e-a64f-41ac-800c-f834eb8869e2?action=apply
```

Apply two labels, named TestLabel1 and TestLabel2, to one device:

```
https://mycore.mobileiron.com/api/v1/dm/labels/TestLabel1,TestLabel2/bdcbdf2e-a64f-41ac-800c-f834eb8869e2?action=apply
```

Apply two labels, named TestLabel1 and TestLabel2, to two devices.

```
https://mycore.mobileiron.com/api/v1/dm/labels/TestLabel1,TestLabel2/bdcbdf2e-a64f-41ac-800c-f834eb8869e2,3eaab11d-0437-4528-a0db-0713f75a701b?action=apply
```

URI: https://{host-name}/api/v1/dm/labels/{label}/{deviceUuid}	All labels assigned to uuid are returned.
Http Method:	PUT
Format:	xml, json
Request:	
label	Required. The name of the label to be applied. When applying multiple labels to a device, separate each label with a comma, e.g., LabelOne,LabelTwo,LabelThree.

deviceUuid	Required. The device Uuid to which the label is to be applied. When a label is applied to multiple devices, separate each Uuid with a comma, e.g., b0bbcd5c-09ed-4de0-97b5-5bb18056b177,893e11e4-2281-43af-85e7-33dde660316d Note: Do not put spaces between commas.
action=	Required. This parameter is a query parameter: ?action=apply
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<deviceManagementWebServiceResponse>	
<messages>	
<message>Label(s) applied to device (s).</message>	
<SucceededLabels>	
<succeededLabel>	
<label>Executive</label>	
<device>	
<uuid> a7f4ce8e-e6de-4a0a-b487-a32a63840e32</uuid>	
</device>	
</succeededLabel>	
</SucceededLabels>	
<deviceManagementWebServiceResponse>	

Remove Labels from a Device

Use this API to remove:

- one label from one device
- multiple labels from one device
- one label from multiple devices

The API response contains error messages in these situations:

- The request contains an invalid device uuid.
- The request contains an invalid label.
- A label in the request is not applied to one or more of the specified devices.

Examples:

Remove one label named TestLabel from one device:

```
https://mycore.mobileiron.com/api/v1/dm/labels/TestLabel/bdcbdf2e-a64f-41ac-800c-f834eb8869e2?action=remove
```

Remove two labels, named TestLabel1 and TestLabel2, from one device:

```
https://mycore.mobileiron.com/api/v1/dm/labels/TestLabel1,TestLabel2/bdcbdf2e-a64f-41ac-800c-f834eb8869e2?action=remove
```

Remove two labels, named TestLabel1 and TestLabel2, from two devices.

```
https://mycore.mobileiron.com/api/v1/dm/labels/TestLabel1,TestLabel2/bdcbdf2e-a64f-41ac-800c-f834eb8869e2,3eaab11d-0437-4528-a0db-0713f75a701b?action=remove
```

URI: https://{host-name}/api/v1/dm/labels/{label}/{deviceUuid}	All labels assigned to uuid are returned.
Http Method:	PUT
Format:	xml, json
Request:	
label	Required. The name of the label to be removed. When removing multiple labels to a device, separate each label with a comma, e.g., LabelOne,LabelTwo,LabelThree.
deviceUuid	Required. The device Uuid from which the label is to be removed. When a label is removed from multiple devices, separate each Uuid

	with a comma, e.g., b0bbcd5c-09ed-4de0-97b5-5bb18056b177,893e11e4-2281-43af-85e7-33dde660316d Note: Do not put spaces between commas.
action=	Required. This parameter is a query parameter: ?action=remove
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<deviceManagementWebServiceResponse>	
<messages>	
<message>Label(s) removed from device (s).</message>	
<SucceededLabels>	
<succeededLabel>	
<label>Executive</label>	
<device>	
<uuid> a7f4ce8e-e6de-4a0a-b487-a32a63840e32</uuid>	
</device>	
</succeededLabel>	
</SucceededLabels>	
<deviceManagementWebServiceResponse>	

List of Operators

MobileIron retains a default list of operators for use during device registration. Operators may be enabled or disabled by an administrator. This API returns a complete list of all operators, regardless of whether they are used.

Example:

```
https://mycore.mobileiron.com/api/v1/dm/operators
```

URI: https://{host-name}/api/v1/dm/operators	All operators defined in database are returned.
Http Method:	GET
Format:	xml, json
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
< deviceManagementWebServiceResponse >	
<messages>	
<message>1 Operator (s) returned</message>	Status message. Operator count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<operators>	
<operator>	
<carrierShortName>AT&T</carrierShortName>	Operator name.
<carrierType>Mobile</carrierType>	Mobile: Operator provides mobile services.

	Fixed: Operator provides fixed telecom services.
<countryCode>1</countryCode>	Numeric country code.
<countryId>183</countryId>	Country identifier for the device. MOBILEIRON CORE assigns this identifier to the country.
<countryName>United States</countryName>	Country name.
<enabled>true</enabled>	True indicates the operator is enabled (configured for display) in the registration screen. False indicates the operator is disabled.
<id>269</id>	Unique operator identifier in the database.
</operator>	
</operators>	
< /deviceManagementWebServiceResponse>	

List of Countries

MobileIron retains a default list of countries in the database. A country selection populates the country code field. This API provides a complete list of all defined countries, regardless of whether they are used. This list of countries is used during device registration.

Example:

```
https://mycore.mobileiron.com/api/v1/dm/countries
```

URI: https://{host-name}/api/v1/dm/countries	All countries defined in the database are returned.
Http Method:	GET
Format:	xml, json
Response Status Code:	

'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
< deviceManagementWebServiceResponse >	
<messages>	
<message>2 Countries returned</message>	Status Message. Country count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<countries>	
<country>	
<countryName>United States</countryName>	Country name.
<countryCode>1</countryCode>	Numeric country code.
<isoAlpha2Code>US</isoAlpha2Code>	ISO Alpha 2 country code.
<enabledForRegistration>102</enabledForRegistration>	Whether the MobileIron Core administrator enabled the country for registration. 102 means disabled. 116 means enabled. Note: 116 is the ASCII value for 't', which stands for true, and 102 is the ASCII value for 'f', which stands for false.
</country>	

<country>	
<countryName>India</countryName>	
<countryCode>91</countryCode>	
<isoAplha2Code>IN</isoAplha2Code>	
<enabledForRegistration>116</enabledForRegistration>	
</country>	
</countries>	
< /deviceManagementWebServiceResponse>	

Send Action to bulk devices

This request sends an action to multiple devices. The possible actions are:

- Lock or unlock one or more devices.
- Retire one or more devices.
- Wipe one or more devices.
- Wake up the MobileIron client on one or more devices, to force the clients to check in with MobileIron Core.

MobileIron Core validates that the request has a valid action and valid devices, and then sends the response. MobileIron Core performs the actions after sending the response. You can view the actions taken by looking at Logs & Events | All Logs in the Admin Portal.

If the requested action is invalid, MobileIron Core sends a response so indicating. If some devices are invalid, the response lists them, but MobileIron Core will still take the action on the valid devices.

Note: The UNLOCK bulk request is the exception. In this case, MobileIron Core performs the action before sending the response.

Example:

A LOCK request on two valid devices:

```
https://mycore.mobileiron.com/api/v1/dm/bulk/devices/LOCK?deviceUuids=1ac8bd81-4ab9-4e3e-
b3a8-0c4f70521d23&deviceUuids=ab7e93f4-90e2-485b-82b9-7a030ef7d985
```

The resulting response:

```
<deviceManagementWebServiceResponse>
<messages/>
</deviceManagementWebServiceResponse>
```

URI: https://{host-name}/api/v1/dm/bulk/devices/{actiontype}	
Http Method:	POST
Format:	xml, json
Request:	
actiontype	Required. Specify one of these action types: LOCK UNLOCK WAKEUP_DEVICE RETIRE WIPE Note: These values are all capital letters.
deviceUuids	Required. List each device uuid as a query parameter that has the name deviceUuids. By default, the maximum number of device uuids you can list is 20,000. You can configure this value by setting the variable bulk.api.maxdeviceuuids in the file mifs.properties. This file is located in the directory /mi/tomcat-properties in the Linux system in which MobileIron Core is running. Warning: The name of the parameter is deviceUuids, with an “s” at the end.

Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
< deviceManagementWebServiceResponse >	
<messages>	Status information, if any. Also specifies errors in the request, if any. If MobileIron Core has no status or errors to report, this field is empty.
<message>Invalid action</message>	If you do not capitalize the action or misspell it, this field contains the value "Invalid action".
<invalidDevices>	If the request contains one or more invalid device uuids, this field lists them.
<uuid> 1ac8bd81-4ab9-4e3e-b3a8-0c4f70521d23</uuid>	
<uuid> 623094f9-645b-4ecf-8840-78597cc1254b</uuid>	
</invalidDevices>	
<passcodes>	Only for the UNLOCK action.
<passcode>	
<uuid>1ac8bd81-4ab9-4e3e-b3a8-0c4f70521d23</uuid>	The device uuid.
<name>jdoe:Android:6505551212</name>	The concatenation of the device's user's name, device platform, and phone number.
<value>Q</value>	For iOS devices: Q – The device is MDM managed and has a passcode.

	F – The device is MDM managed but has no passcode. For Android devices: Q – The device has a passcode. F – The device has no passcode. For Blackberry devices: Passcode to unlock the device. If value is empty, then the device was most likely locked with the user-set passcode.
</passcode>	
</passcodes>	
<failedDevices>	Only for the UNLOCK action.
<message>	
	Message contents indicates the device for which the unlock failed.
</message>	
</failedDevices>	
< /deviceManagementWebServiceResponse>	

Send message to devices

This request sends a message to one or more devices using email, SMS or push notification (e.g., APNS).

MobileIron Core validates that the request has valid devices, and then sends the response. MobileIron Core sends the messages to the devices after sending the response. You can view the results of sending the messages by looking at Logs & Events | All Logs in the Admin Portal.

If some devices are invalid, the response lists them, but MobileIron Core will still send the message to the valid devices.

Examples:

Send an SMS to two devices based on UDID.

```
https://app027.auto.mobileiron.com/api/v1/dm/bulk/sendmessage?mode=sms&message=Hello
World&deviceUuid=e6d4f5f0-d883-41d2-8e87-c76fb4ef4cde&deviceUuid=54bc5eb5-592c-472e-98d2-
e859bd037fef
```

The resulting response:

```
<deviceManagementWebServiceResponse>
<messages/>
<message> Message sent successfully for all devices.</message>
</deviceManagementWebServiceResponse>
```

Send an SMS to a device based on Wi-Fi MAC address:

```
https://app027.auto.mobileiron.com/api/v1/dm/bulk/mac/sendmessage?mode=sms&message=Hello
World&deviceWiFiMacAddress=00237696635F
```

1.URI: https://{host-name}/api/v1/dm/bulk/sendmessage	
Http Method:	POST
Format:	xml, json
Request:	
deviceUuid	Required. List each device uuid as a query parameter that has the name deviceUuid. By default, the maximum number of device uuids you can list is 20,000. You can configure this value by setting the variable bulk.api.maxdeviceuuids in the file mifs.properties. This file is located in the directory /mi/tomcat-properties in the Linux system in which MobileIron Core is running. Warning: The name of this parameter is deviceUuid, with no “s” at the end.

message	Required.
subject	Valid only when the mode is email.
mode	Required. Possible values: sms email pns (indicates push notification service)
2.URI: https://{host-name}/api/v1/dm/bulk/mac/sendmessage	
Http Method:	POST
Format:	xml, json
Request:	
deviceWiFiMacAddress	Required. List each Wi-Fi MAC address as a query parameter that has the name macAddress.
message	Required.
subject	Valid only when the mode is email.
mode	Required. Possible values: sms email pns

Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
< deviceManagementWebServiceResponse >	
<messages>	
<message>Message sent successfully for all devices.</message>	Status information.
</messages>	
<invalidDevices>	If the request contains one or more invalid device uuids, this field lists them.
<uuid> 1ac8bd81-4ab9-4e3e-b3a8-0c4f70521d23</uuid>	
<uuid> 623094f9-645b-4ecf-8840-78597cc1254b</uuid>	
</invalidDevices>	
</messageSentFailed>	Indicates the message was not sent for at least one specified device, due to, for example, an invalid device uuid.
< /deviceManagementWebServiceResponse>	

Get Profiles for a Device

This API returns the configurations and policies for a specified device uuid.

Example:

```
https://app027.auto.mobileiron.com/api/v1/dm/devices/profiles/e6d4f5f0-d883-41d2-8e87-c76fb4ef4cde
```

URI: https://{host-name}/api/v1/dm/devices/profiles/{deviceUuid}	All profiles applied to the specified device are returned.

Http Method:	GET
Format:	xml, json
Request:	
deviceUuid	Required. Unique ID of the device. This ID can be retrieved in the response of other API calls, such as Device Registration or Get Device Details.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<deviceManagementWebServiceResponse>	
<messages />	
<profiles>	
<profile id="-7">	
<uuid>misystem-default-docs-policy</uuid>	
<name>Default Docs@Work Policy</name>	
<policyType>DOCS</policyType>	
<status>Applied</status>	
<profileType>POLICY</profileType>	

<lastUpdatedAt>1347343156739</lastUpdatedAt>	
</profile>	
<profile id="-3">	
<uuid>misystem-default-security-policy</uuid>	
<name>Default Security Policy</name>	
<policyType>SECURITY</policyType>	
<status>Applied</status>	
<profileType>POLICY</profileType>	
<lastUpdatedAt>1347343165503</lastUpdatedAt>	
</profile>	
<profile id="-2">	
<uuid>misystem-default-privacy-policy</uuid>	
<name>Default Privacy Policy</name>	
<policyType>PRIVACY</policyType>	
<status>Applied</status>	
<profileType>POLICY</profileType>	
<lastUpdatedAt>1347343156731</lastUpdatedAt>	
</profile>	
<profile id="-2">	
<name> System - iOS MDM</name>	
<policyType>MDM</policyType>	
<status>Applied</status>	
<profileType>APP</profileType>	
<lastUpdatedAt>1347343165501</lastUpdatedAt>	
</profile> </profiles>	

</deviceManagementWebServiceResponse>	

Re-push Profiles for a Device

This API re-pushes the specified configuration or policy for the device uuid.

Example:

```
https://app027.auto.mobileiron.com/api/v1/dm/devices/repushprofile/e6d4f5f0-d883-41d2-8e87-c76fb4ef4cde
https://app386.auto.mobileiron.com/api/v1/dm/devices/repushprofile/1faaaf43-c99d-4c21-bab4-c9e810bd9606?id=3&type=APP
```

URI: https://{host-name}/api/v1/dm/devices/repushprofile/{deviceUuid}	All profiles applied to the specified device are returned.
Http Method:	PUT
Format:	xml, json
Request:	
deviceUuid	Required. Unique ID of the device. This ID can be retrieved in the response of other API calls, such as Device Registration or Get Device Details.
id	Profile ID. Use the Get Profiles API to get the profile ID.
type	APP for configuration POLICY for policy
Response Status Code:	
'404 – No Data Found'	There is no data.

'200 – OK'	Data is present and the response is returned.
Response:	

Exchange ActiveSync (EAS)

List All ActiveSync Devices

This API returns a list of ActiveSync unique device IDs (uuid) that use ActiveSync to connect to the enterprise. Devices are grouped as follows in the return list: Registered Allowed, Registered Blocked, Unregistered Allowed, Unregistered Blocked, and Wiped.

An administrator may wish to block an ActiveSync device to prevent it from connecting to the enterprise (i.e., get email). If a device is blocked, any previously synchronized email is removed. Use the allow feature to permit a device to connect to the enterprise which was previously blocked.

Example:

```
https://mycore.mobileiron.com/api/v1/eas/devices
```

URI: https://{host-name}/api/v1/eas/devices	All ActiveSync devices are returned.	
Http Method:	GET	
Format:	xml, json	
Response Status Code:		
'404 – No Data Found'	There is no data.	
'200 – OK'	Data is present and the response is returned.	
Response:		
<easWebServiceReponse>		
<messages>		
<message>212 Device(s) returned</message>	Status message. Lists the number of devices found, or that no devices were found.	
</messages>		
<registeredAllowedDevices>		
<registeredAllowedDevice>	All the child elements of the	

	<registeredAllowedDevice> element are also in the <registeredBlockedDevice>, <unregisteredAllowedDevice>, <unregisteredBlockedDevice>, and <wipedDevice> elements.	
<uuid>hdgd-e93c-49b1-88d6-222f54132445</uuid>	ActiveSync unique identifier for the device.	
<domain>exchdomain.com</domain>	The Exchange ActiveSync domain of the device.	
<deviceId>Appl87025CNUA4S</deviceId>	ActiveSync device identifier.	
<mailboxId>jdoe113</mailboxId>	ActiveSync mailbox ID for the device.	
<userName>jdoe113</username>	ActiveSync username associated with the device.	
<phoneNumber>6505551212</phoneNumber>	Phone number associated with the device.	
<model>iPhone</model>	Device model as recorded by the ActiveSync server.	
<platform>iOS</platform>	Device operating system as recorded by the ActiveSync server.	
<platformCode>11</platformCode>	Device operating system code as recorded by te ActiveSync server.	
<status>Registered</status>	MobileIron status for the device.	
<activeSyncStatus>Allowed</activeSyncStatus>	ActiveSync status for the device.	
<firstSyncTime>1326179585000</firstSyncTime>	The timestamp for the first time the device synchronized ActiveSync data. This time field is expressed in Unix Epoch Time, which is the number of milliseconds since January 1, 1970.	
<lastSyncTime>1326180768000</lastSyncTime>	The timestamp for te last time the device synchronized ActiveSync data. This time field is expressed in Unix Epoch Time, which is the number of milliseconds since January 1, 1970.	
<miDeviceUuid>6f72cabb-1d8b-4965-aa8e-	MobileIron unique identifier for the device.	

a355deab8222</miDeviceUuid>		
<actionSource>EXCHANGE</actionSource>		
</registeredAllowedDevice>	ActiveSync unique identifier for a registered device with Allowed status.	
</registeredAllowedDevices>	A list of ActiveSync unique identifiers for registered devices with Allowed status.	
<registeredBlockedDevices>	A list of ActiveSync unique identifiers for registered devices with Blocked status.	
<registeredBlockedDevice>		
<uuid>hgdgd-fsg-4wfsb1-dgdg-dgfdg</uuid>	ActiveSync unique identifier for a registered device with Blocked status.	
</ registeredBlockedDevice >		
</ registeredBlockedDevices>		
<unregisteredAllowedDevices>	A list of ActiveSync unique identifiers for unregistered devices with Allowed status.	
<unregisteredAllowedDevice>		
<uuid>8herw5345d711cdc-e93c-dfg-hgdf-hssgfd</uuid>	ActiveSync unique identifier for an unregistered device with Allowed status.	
</ unregisteredAllowedDevice >		
</ unregisteredAllowedDevices>		
<unregisteredBlockedDevices>	A list of ActiveSync unique identifiers for unregistered devices with Blocked status.	
<unregisteredBlockedDevice>		
<uuid>34gdrtger-4err-gd-88d6-2fes</uuid>	ActiveSync unique identifier for an unregistered device with Blocked status.	
</ unregisteredBlockedDevice >		
</ unregisteredBlockedDevices>		
<wipedDevices>	A list of ActiveSync unique identifiers for devices that have been wiped via ActiveSync	

	wipe.	
< wipedDevice >		
<uuid>sersdfsc-e93c-49b1-88d6-sg2wefwef</uuid>	ActiveSync unique identifier for a wiped device.	
</ wipedDevice >		
</ wipedDevices >		
</easWebServiceReponse>		

Device Details for ActiveSync

This API returns a variety of details for devices using ActiveSync to connect to the Enterprise. Details ranging from the first time such device was synced to the ActiveSync version are returned.

Example:

```
https://mycore.mobileiron.com/api/v1/eas/devices/ee8198d9-5d79-4961-94c4-e21bf04b2467
```

URI: https://{host-name}/api/v1/eas/devices/{EASDeviceUuid}	Device details of the input Exchange ActiveSync device uuid is returned
Http Method:	GET
Format:	xml, json
Request:	
EASDeviceUuid	Required. Exchange ActiveSync device uuid.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	

<easDevice>	
<uuid>f5def24e-4380-4565-bd2f-8cf002fd64cd</uuid>	
<details>	A series of key/value pairs determined by ActiveSync.
<entry>	
<key>ActionSource</key>	EXCHANGE('e', "Exchange"): Initial state set by the Exchange server. AUTOBLOCK('a', "Auto"): state set by Auto Block action POLICY('p', "Policy"): state set by policy enforcement MANUAL('m', "Manual"): state set manually by administrator UNKNOWN('u', "Unknown")
<value>Exchange</value>	
</entry>	
<entry>	
<key>LastPingHeartbeat</key>	
<value>600</value>	
</entry>	
<entry>	
<key>DeviceID</key>	
<value>Appl9C0180RF75J</value>	
</entry>	
<entry>	
<key>FirstSyncTime</key>	
<value>7/6/2010 11:29:33 AM</value>	

</entry>	
<entry>	
<key>DevicePolicyApplicationStatus</key>	
<value>AppliedInFull</value>	
</entry>	
<entry>	
<key>LastSyncAttemptTime</key>	
<value>7/7/2010 12:14:52 PM</value>	
</entry>	
<entry>	
<key>NumberOfFoldersSynced</key>	
<value>2</value>	
</entry>	
<entry>	
<key>DeviceType</key>	
<value>iPod</value>	
</entry>	
<entry>	
<key>DeviceModel</key>	
<value>iPod</value>	
</entry>	
<entry>	
<key>DeviceUserAgent</key>	
<value>Apple-iPod/705.18</value>	
</entry>	

<entry>	
<key>Status</key>	
<value>DeviceOk</value>	
</entry>	
<entry>	
<key>Guid</key>	
<value>61a8a847-8e3b-4496-8da6-587b845b77cf</value>	
</entry>	
<entry>	
<key>DeviceAccessState</key>	
<value>Allowed</value>	
</entry>	
<entry>	
<key>DeviceEnableOutboundSMS</key>	
<value>False</value>	
</entry>	
<entry>	
<key>Identity</key>	
<value>newyork.mobileiron.com/Users/Sang Truong/ExchangeActiveSyncDevices/iPod§Appl9C0180RF75J</value>	
</entry>	
<entry>	
<key>DeviceAccessStateReason</key>	
<value>Individual</value>	
</entry>	
<entry>	

<key>DevicePolicyApplied</key>	
<value>Default</value>	
</entry>	
<entry>	
<key>LastPolicyUpdateTime</key>	
<value>7/6/2010 11:29:34 AM</value>	
</entry>	
<entry>	
<key>IsRemoteWipeSupported</key>	
<value>True</value>	
</entry>	
<entry>	
<key>LastSuccessSync</key>	
<value>7/7/2010 12:14:52 PM</value>	
</entry>	
<entry>	
<key>RecoveryPassword</key>	
<value>********</value>	
</entry>	
<entry>	
<key>DeviceActiveSyncVersion</key>	
<value>12.1</value>	
</entry>	
</details>	
</easDevice>	

Request Action on ActiveSync Device

This API requests status changes to devices using ActiveSync to connect to the Enterprise.

Example:

```
https://mycore.mobileiron.com/api/v1/eas/devices?action=BLOCK_DEVICE&uuids=ee8198d9-5d79-4961-94c4-e21bf04b2467&uuids=fe816c9-4c68-3850-83b3-d10ae93a1356
```

URI: https://{host-name}/api/v1/eas/devices	The requested action will be applied on the device.
Http Method:	PUT
Format:	xml, json
Request:	
uuids	Required. One or more Exchange ActiveSync device uuids.
action	Required. Valid Actions are: BLOCK_DEVICE: Block the device from accessing ActiveSync server. REINSTATE_DEVICE: Allow the device to access ActiveSync server. WIPE: Wipe the device, which returns its settings to the factory defaults.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<easWebServiceResponse>	
<messages>	
<message>	
1 device(s) modified to BLOCK_DEVICE	Status message. Displays action taken on EAS device.

status	
</ message >	
</messages>	
</easWebServiceResponse>	

Security Management

The Security Management API addresses authentication tasks. These tasks apply to both local users and LDAP users.

Update Password for a User

This API changes the password for a single user.

Example:

```
https://mycore.mobileiron.com/api/v1/sm/authentication/users/jdoe
For security reasons, include the old and new passwords in the HTTP request body rather than
as query parameters. For example:
PUT /api/v1/sm/authentication/users/jdoe HTTP/1.1
Host: mycore.mobileiron.com
Content-Length: 44
Accept: application/json
Authorization: Basic amRvZTphYmNkMTIzNA==
Content-Type: application/x-www-form-urlencoded
oldpassword=abcd1234&newpassword=wxy!13579
```

URI: https://{host-name}/api/v1/sm/authentication/users/{username}	Updates password for input username.
Http Method:	PUT
Format:	xml, json
Request:	
username	Required. Unique login user name.
oldpassword	Current password of the user. Note: For security reasons, include this parameter in HTTP request body.

	Required only if the MobileIron Core setting to save the user password is set to Yes. You can set this value in the Admin Portal, using Settings | Preferences. When oldpassword is required, make sure that the value you provide in the request is correct. If it is not included or is not correct, the response contains a failure message. Note: When you create a local user using the API to Register a Device, MobileIron Core sets the user's password to the user ID (called username in this request).
newpassword	Required. New password of the user. The password must be between 8 and 20 characters. Note: For security reasons, include this parameter in HTTP request body.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<securityManagementWebServiceResponse>	
<userName>jdoe</userName>	

<messages>	
<message>	
Password changed successfully for user: jdoe	Status Message. Success shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</ message >	
</messages>	
</securityManagementWebServiceResponse>	

Find a User

This API finds a single user by username or email address. User details will be returned only if the search finds an exact match of the username or email address.

Example:

```
https://mycore.mobileiron.com/api/v1/sm/users/jdoe
```

URI: https://{host-name}/api/v1/sm/users/{username}	Finds the user specified for input username or email address
Http Method:	GET
Format:	xml, json
Request:	
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<securityManagementWebServiceResponse>	

```
<userName>miadmin</userName>
<messages/>
<user id="9001">
<uuid>f89d8cbf-59d7-47e6-97c2-4681ed8f954a</uuid>
<principal>miadmin</principal>
<createdAt>1374085200000</createdAt>
<displayName>miadmin</displayName>
<email>miadmin@mobileiron.com</email>
<enabled>true</enabled>
<firstName>miadmin</firstName>
<forcePasswordChange>false</forcePasswordChange>
<googleAppsEncryptionAlgVersion>0</googleAppsEncryptionAlgVersion>
<lastAdminPortalLoginTime>1374178220915</lastAdminPortalLoginTime>
<lastName></lastName>
<opaque>true</opaque>
<roles>ROLE_MPW_LOCK</roles>
<roles>ROLE_USER_MANAGEMENT_RW</roles>
<roles>ROLE_MAI_RW</roles>
<roles>ROLE_APPS_AND_FILES_RW</roles>
<roles>ROLE_SENTRY_FOR_IPAD</roles>
<roles>ROLE_ADMIN_LOCATE</roles>
<roles>ROLE_LOG_R</roles>
```

<roles>ROLE_TROUBLESHOOTING_RW</roles> <roles>ROLE_EVENT_CENTER_RW</roles> <roles>ROLE_ADMIN_WIPE</roles> <roles>ROLE_SELECTIVE_WIPE</roles> <roles>ROLE_MPW_REG</roles> <roles>ROLE_SECURITY_AND_POLICIES_RW</roles> <roles>ROLE_MPW_LOCATE</roles> <roles>ROLE_API</roles> <roles>ROLE_SMARTPHONES_AND_DEVICES_RW</roles> <roles>ROLE_MPW_WIPE</roles> <roles>ROLE_USER_PORTAL_RW</roles> <roles>ROLE_CONNECTOR</roles> <roles>ROLE_SETTINGS_RW</roles> <userSource>76</userSource> </user> </securityManagementWebServiceResponse>	

Search LDAP Users

This API finds users by username. The search string cannot be less than 2 characters. If the search results are more than the search limit (can be configured in mifs.properties) an error is returned. Default search limit is 100.

Example:

```
https://mycore.mobileiron.com/api/v1/sm/users/search/ldap/?userid=jdoe
```

URI: https://{host-name}/api/v1/sm/users/search/ldap/{userid}	Finds the users for the specified username search string.
Http Method:	GET

Format:	xml, json
Request:	
userid	Required. Username search string. Minimum 2 characters.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
'400- Bad Request'	1. If the input search string is less than 2 characters. 2. If the search results are more than the limit.
Response:	
<securityManagementWebServiceResponse> <userName>testuser000</userName> <messages/> <users> <user> <principal>testuser0001</principal> <displayName>testuser0001</displayName> <email>testuser0001@auto1.mobileiron.com</email> <enabled>false</enabled> <firstName>Test</firstName> <forcePasswordChange>false</forcePasswordChange>	

<lastName>User0001</lastName> <opaque>true</opaque> <userSource>68</userSource> </user> <user> <principal>testuser0003</principal> <displayName>testuser0003</displayName> <email>testuser0003@auto1.mobileiron.com</email> <enabled>false</enabled> <firstName>Test</firstName> <forcePasswordChange>false</forcePasswordChange> <lastName>User0003</lastName> <opaque>true</opaque> <userSource>68</userSource> </user> </users> </securityManagementWebServiceResponse>	

Authenticate a User

This API authenticates a single user by username.

Example:

```
https://mycore.mobileiron.com/api/v1/sm/authentication
```

For security reasons, include the password in the HTTP request body rather than as a query parameter. For example:

```
POST /api/v1/sm/authentication HTTP/1.1
```

```
Host: mycore.mobileiron.com
Content-Length: 31
Accept: application/json
Authorization: Basic amRvZTphYmNkMTIzNA==
username=jdoe&password=abcd1234
```

URI: https://{host-name}/api/v1/sm/users/{username}	Finds the user specified for input username.
Http Method:	POST
Format:	xml, json
Request:	
username	String Required Note: For security reasons, include this parameter in HTTP request body.
Password	String Required The password must be between 8 and 20 characters. Note: For security reasons, include this parameter in HTTP request body.
Response Status Code:	
'401 – Unauthorized'	If the username/password is invalid.
'200 – OK'	If username and password are valid then User details are returned in the response.
Response:	
<securityManagementWebServiceResponse>	

```
<userName>miadmin</userName>
<messages/>
<user id="9001">
<uuid>f89d8cbf-59d7-47e6-97c2-4681ed8f954a</uuid>
<principal>miadmin</principal>
<createdAt>1374085200000</createdAt>
<displayName>miadmin</displayName>
<email>miadmin@mobileiron.com</email>
<enabled>true</enabled>
<firstName>miadmin</firstName>
<forcePasswordChange>false</forcePasswordChange>
<googleAppsEncryptionAlgVersion>0</googleAppsEncryptionAlgVersion>
<lastAdminPortalLoginTime>1374178220915</lastAdminPortalLoginTime>
<lastName></lastName>
<opaque>true</opaque>
<roles>ROLE_MPW_LOCK</roles>
<roles>ROLE_USER_MANAGEMENT_RW</roles>
<roles>ROLE_MAI_RW</roles>
<roles>ROLE_APPS_AND_FILES_RW</roles>
<roles>ROLE_SENTRY_FOR_IPAD</roles>
<roles>ROLE_ADMIN_LOCATE</roles>
<roles>ROLE_LOG_R</roles>
```

<roles>ROLE_TROUBLESHOOTING_RW</roles> <roles>ROLE_EVENT_CENTER_RW</roles> <roles>ROLE_ADMIN_WIPE</roles> <roles>ROLE_SELECTIVE_WIPE</roles> <roles>ROLE_MPW_REG</roles> <roles>ROLE_SECURITY_AND_POLICIES_RW</roles> <roles>ROLE_MPW_LOCATE</roles> <roles>ROLE_API</roles> <roles>ROLE_SMARTPHONES_AND_DEVICES_RW</roles> <roles>ROLE_MPW_WIPE</roles> <roles>ROLE_USER_PORTAL_RW</roles> <roles>ROLE_CONNECTOR</roles> <roles>ROLE_SETTINGS_RW</roles> <userSource>76</userSource> </user> </securityManagementWebServiceResponse>	

Alerts

MobileIron's Event Center enables administrators to connect events to specific alerts. The following events are recognized:

- International Roaming Event
- Threshold Reached Event
- SIM Changed Event
- Storage Size Exceeded Event
- System Event
- Policy Violations Event

This API can retrieve alerts generated by an above named event.

Alerts include a variety of characteristics, such as severity, lifecycle status, and read/unread status. Alert Lifecycle statuses are defined as follows:

1. Created: the conditions for generating the alert have been met.
2. Processed: the alert has been generated.
3. Dispatched: the alert has been sent.
   - Dispatch Pending: alert is ready for dispatch.
   - Dispatching: dispatch is in progress.
   - Dispatched: dispatch has been completed successfully.
   - Dispatch Failed: dispatch failed.

Get All Alerts

This API returns all alerts. You can filter the alerts by their read/unread status.

Examples:

Get all alerts:

```
https://mycore.mobileiron.com/api/v1/alerts
```

Get all alerts that have not been read:

```
https://mycore.mobileiron.com/api/v1/alerts?isRead=false
```

URI: https://{host-name}/api/v1/alerts	Returns list of all alerts.
Http Method:	GET

Format:	xml, json
Request:	
isRead	Filter by the read status of the alert. True returns all alerts that are marked read. False returns all the alerts that are marked unread.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<alertWebServiceResponse>	
<messages>	
<message>1 alert(s) returned</message>	Status Message. Alert count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<alerts>	
<alert id="5">	Internal database ID uniquely identifies alert.
<deviceUuid>6482dce2-ea75-400a-9f2c-d67766d942cf</deviceUuid>	

<dispatchDeviceUuid>6482dce2-ea75-400a-9f2c-d67766d942cf</dispatchDeviceUuid>	
<userUuid>asdasd-34sd-234sdf-sfsdfd</userUuid>	uuid of the user.
<labelId>163</labelId>	The internal id of the label that triggered the alert. .
<eventSubscriptionName>m1</eventSubscriptionName>	
<alertDate>2010-05-01T01:02:00+00:00</alertDate>	
<alertText>WARNING::Memory size exceeded 1% for Phone #: 14085551212 (miadmin), Total Memory Size: 154.86MB, Free Memory Size: 133.59MB</alertText>	Alert content.
<isActive>false</isActive>	true -- the alert is unread. false -- the alert is read.
<retries>2</retries>	Number of attempts that have been made to send this alert.
<updateBy>alertprocessor</updateBy>	Name of user/system component which updated this alert.
<updatedAt>2010-04-30T01:04:00+00:00</updatedAt>	The time at which this alert record was last modified.
<userName>miadmin</userName>	Recipient user name.
<alertDefnname>MEMORY_SIZE_EXCEEDED_ ALERT</alertDefnname>	Alert type: INTERNATIONAL_ ROAMING_ALERT THRESHOLD_REACHED_ ALERT SIM_CHANGED_ALERT

	MEMORY_SIZE_ EXCEEDED_ALERT SYSTEM_ALERT POLICY_VIOLATIONS_ ALERT
<severity>WARNING</severity>	Alert severity: INFORMATION WARNING CRITICAL
<transport>EMAIL</transport>	Means by which alert is communicated: EMAIL SMS APNS (iPhone only)
<status>DISPATCHED</status>	Alert dispatch status: CREATED PROCESSED DISPATCH_PENDING DISPATCHING DISPATCHED DISPATCH_FAILED
<isAlertRead>true</isAlertRead>	Not used. The isActive field indicates whether the alert is read or unread.
</alert>	
</alerts>	
</alertWebServiceResponse>	

Get All Alerts for Phone Number

This API returns all alerts for a single device phone number. You can further filter the alerts by their read/unread status.

The fields in the response are the same as the fields in the Get All Alerts Response. However, the set of alerts is limited to alerts for the phone number specified in the request.

Examples:

Get all alerts for a phone number:

```
https://mycore.mobileiron.com/api/v1/alerts/phones/6505551212
```

Get all alerts for a phone number that have been read:

```
https://mycore.mobileiron.com/api/v1/alerts/phones/6505551212?isRead=true
```

URI: https://{host-name}/api/v1/alerts/phones/{phonenumber}	Returns list of all alerts for input phone number.
Http Method:	GET
Format:	xml, json
Request:	
phoneNumber	Required. Phone number.
isRead	Filter by the read status of the alert. True returns all alerts which are marked read. False returns all the alerts which are marked unread.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	

<alertWebServiceResponse>	
<messages>	
<message>1 alert(s) returned</message>	Status Message. Alert count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<alerts>	
<alert>	
<id>5</id>	
<deviceUuid>6482dce2-ea75-400a-9f2c-d67766d942cf</deviceUuid>	
<dispatchDeviceUuid>6482dce2-ea75-400a-9f2c-d67766d942cf</dispatchDeviceUuid>	
<userUuid>asdasd-34sd-234sdf-sfsdfd</userUuid>	uuid of the user.
<labelId>1</labelId>	The internal id of the label that triggered the alert.
<eventSubscriptionName>m1</eventSubscriptionName>	
<alertDate>2010-05-01T01:02:00+00:00</alertDate>	
<alertText>WARNING::Memory size exceeded 1% for Phone #: 14085551212 (miadmin), Total Memory Size: 154.86MB, Free Memory Size: 133.59MB</alertText>	Alert content.
<isActive>false</isActive>	true -- the alert is unread. false -- the alert is read.

<retries>0</retries>	The number of attempts that have been made to send this alert.
<updateBy>alertprocessor</updateBy>	
<updatedAt>2010-04-30T01:04:00+00:00</updatedAt>	
<userName>miadmin</userName>	Recipient user name.
<alertDefnname>MEMORY_SIZE_EXCEEDED_ ALERT</alertDefnname>	Alert type.
<severity>WARNING</severity>	Alert severity: INFORMATION WARNING CRITICAL
<transport>EMAIL</transport>	Means by which alert is communicated: EMAIL SMS APNS (iPhone only)
<status>DISPATCHED</status>	Alert dispatch status (as described in the Alerts section above): CREATED PROCESSED DISPATCH_PENDING DISPATCHING DISPATCHED DISPATCH_FAILED
<isAlertRead>true</isAlertRead>	Not used. The isActive field indicates whether the alert is read or unread.

</alert>	
</alerts>	
</alertWebServiceResponse>	

Get all Alerts for User

This API returns all alerts for a single user. Because users may have multiple devices, this API returns all alerts on all devices matching the username. You can further filter the alerts by their read/unread status.

The fields in the response are the same as the fields in the Get All Alerts Response. However, the set of alerts is limited to alerts for the user specified in the request.

Examples:

Get all alerts for a user:

```
https://mycore.mobileiron.com/api/v1/alerts/users/jdoe
```

Get all unread alerts for a user:

```
https://mycore.mobileiron.com/api/v1/alerts/users/jdoe?isRead=false
```

URI: https://{host-name}/api/v1/alerts/users/{username}	Returns list of all alerts for the input user name.
Http Method:	GET
Format:	xml, json
Request:	
Username	Required. Unique login user name.
isRead	Filter by the read status of the alert. True returns all alerts which are marked read. False returns all the alerts which are marked unread.

Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<alertWebServiceResponse>	
<messages>	
<message>1 alert(s) returned</message>	Status Message. Alert count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<alerts>	
<alert>	
<id>5</id>	
<deviceUuid>6482dce2-ea75-400a-9f2c-d67766d942cf</deviceUuid>	
<dispatchDeviceUuid>6482dce2-ea75-400a-9f2c-d67766d942cf</dispatchDeviceUuid>	
<userUuid>asdasd-34sd-234sdf-sfsdfd</userUuid>	uuid of the user.
<labelId>1</labelId>	The internal id of the label that triggered the alert.
<eventSubscriptionName>m1</eventSubscriptionName>	
<alertDate>2010-05-01T01:02:00+00:00</alertDate>	

<alertText>WARNING::Memory size exceeded 1% for Phone #: 14085551212 (miadmin), Total Memory Size: 154.86MB, Free Memory Size: 133.59MB</alertText>	Alert content.
<isActive>false</isActive>	true -- the alert is unread. false -- the alert is read.
<retries>2</retries>	Number of attempts that have been made to send this alert.
<updateBy>alertprocessor</updateBy>	
<updatedAt>2010-04-30T01:04:00+00:00</updatedAt>	
<userName>miadmin</userName>	Recipient user name.
<alertDefnname>MEMORY_SIZE_EXCEEDED_ ALERT</alertDefnname>	Alert type.
<severity>WARNING</severity>	Alert severity: INFORMATION WARNING CRITICAL
<transport>EMAIL</transport>	Means by which alert is communicated: EMAIL SMS APNS (iPhone only)
<status>DISPATCHED</status>	Alert dispatch status: CREATED PROCESSED DISPATCH_PENDING DISPATCHING

	DISPATCHED DISPATCH_FAILED
<isAlertRead>true</isAlertRead>	Not used. The isActive field indicates whether the alert is read or unread.
</alert>	
</alerts>	
</alertWebServiceResponse>	

Get All Alerts for a Phone Number of a User

This API returns all alerts for a single phone number of a user. You can further filter the alerts by their read/unread status.

The fields in the response are the same as the fields in the Get All Alerts Response. However, the set of alerts is limited to alerts for the user specified in the request.

Example:

```
https://mycore.mobileiron.com/api/v1/alerts/users/jdoe/phones/16505551212
```

URI: https://{host-name}/api/v1/alerts/users/{username}/phones/{phonenumber}	Returns list of all alerts for the input phone number of the input user name.
Http Method:	GET
Format:	xml, json
Request:	
Username	Required. Unique login user name.
phoneNumber	Required. Phone number.
isRead	Filter by the read status of the alert. True returns all alerts which are marked read.

	False returns all the alerts which are marked unread.
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<alertWebServiceResponse>	
<messages>	
<message>1 alert(s) returned</message>	Status Message. Alert count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<alerts>	
<alert>	
<id>5</id>	
<deviceUuid>6482dce2-ea75-400a-9f2c-d67766d942cf</deviceUuid>	
<dispatchDeviceUuid>6482dce2-ea75-400a-9f2c-d67766d942cf</dispatchDeviceUuid>	
<userUuid>asdasd-34sd-234sdf-sfsdfd</userUuid>	uuid of the user.
<labelId>1</labelId>	The internal id of the label that triggered the alert.
<eventSubscriptionName>m1</eventSubscriptionName>	

<alertDate>2010-05-01T01:02:00+00:00</alertDate>	
<alertText>WARNING::Memory size exceeded 1% for Phone #: 14085551212 (miadmin), Total Memory Size: 154.86MB, Free Memory Size: 133.59MB</alertText>	Alert content.
<isActive>false</isActive>	true -- the alert is unread. false -- the alert is read.
<retries>2</retries>	Number of attempts that have been made to send this alert.
<updateBy>alertprocessor</updateBy>	
<updatedAt>2010-04-30T01:04:00+00:00</updatedAt>	
<userName>miadmin</userName>	Recipient user name.
<alertDefnname>MEMORY_SIZE_EXCEEDED_ ALERT</alertDefnname>	Alert type.
<severity>WARNING</severity>	Alert severity: INFORMATION WARNING CRITICAL
<transport>EMAIL</transport>	Means by which alert is communicated: EMAIL SMS APNS (iPhone only)
<status>DISPATCHED</status>	Alert dispatch status: CREATED PROCESSED DISPATCH_PENDING

	DISPATCHING DISPATCHED DISPATCH_FAILED
<isAlertRead>true</isAlertRead>	Not used. The isActive field indicates whether the alert is read or unread.
</alert>	
</alerts>	
</alertWebServiceResponse>	

Update Alert

This API updates the read/unread status and comments to a particular alert.

Example:

```
https://mycore.mobileiron.com/api/v1/alerts/ 3936?isRead=false&comments=Reset
```

URI: https://{host-name}/api/v1/alerts/{id}	Updates the alert designated by the alert ID.
Http Method:	PUT
Format:	xml, json
Request:	
Id	Required. Alert ID to be updated.
isRead	Required. True updates the alert as read. False updates the alert as unread.
Comments	Required. Comments to be added to the alert. Free form text field (255 character limit).

Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<alertWebServiceResponse>	
<messages>	
<message>Updated alert 3936 successfully</message>	Status Message. Success is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
</alertWebServiceResponse>	

Update List of Alerts

This API updates the read/unread status and comments to multiple alerts, designated by a list of IDs.

Example:

```
https://mycore.mobileiron.com/api/v1/alerts?id=3936&id=3934&isRead=true&comments=”Jdoe read this alert”
```

URI: https://{host-name}/api/v1/alerts/	Updates multiple alerts, designated by alert IDs.
Http Method:	PUT
Format:	xml, json
Request:	
id	Required. Alert IDs to be updated. Note: The IDs are query parameters. For example: https://{host-name}/api/v1/alerts?id=1&id=2&id3

	Three alerts with ids= 1, 2 and 3 are updated with the specified isRead value and comments value.
isRead	Required. True updates the alert as read. False updates the alert as unread.
Comments	Required. Comments to be added to the alert. Free form text field (255 character limit).
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<alertWebServiceResponse>	
<messages>	
<message> Updated 10 alert(s) successfully</message>	Status Message. Alert update count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
</alertWebServiceResponse>	

Policies

Get Policies

This API returns the list of all polices across all devices in the MobileIron system.

Example:

```
https://mycore.mobileiron.com/api/v1/policies
```

URI: https://{host-name}/api/v1/policies	Returns list of all policies installed across all devices.
Http Method:	GET
Format:	xml, json
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<policyWebServiceResponse>	
<messages>	
<message> 1 policy returned.</message>	Status Message. Policy count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<policies>	
<policy id="-2">	
<policyName>Default Privacy Policy</policyName>	Policy name.

<policyType>DEFAULT</policyType>	Policy Type. Either DEFAULT or ENTERPRISE.
<profileType>PRIVACY</profileType>	Profile type. Either PRIVACY, SECURITY, LOCKDOWN, or SYNC.
<status>Active</status>	Active or Inactive.
<active>true</active>	Whether the policy is active. true means Active. false means Inactive.
<defaultPolicy>false</defaultPolicy>	Deprecated.
<description>Default Privacy Policy</description>	Policy description.
<deviceCount>1</deviceCount>	Number of devices for which the policy is applied.
<pendingCount>1</pendingCount>	Number of devices for which the policy is pending.
<priority>1</priority>	Priority
<rules>	Policy rules, which consist of type-value pairs. The set of type-value pairs are listed in Section Policy Rules Policy Rules. The rule shown here is only an example.
<rule>	
<type>PRIVACY_SYNC_CALLLOGS</type>	Rule type
<value>store</value>	
<clientValue>off</clientValue>	
</rule>	

….	
<rules>	
</policy>	
</policies>	
</ policyWebServiceResponse >	

Get Policies by DeviceUUID

This API returns the list of all polices by device uuid in the MobileIron system.

Example:

```
https://mycore.mobileiron.com/api/v1/policies/devices/027d9439-0f75-4d30-8d7d-120b4cb8646b
```

URI: https://{host-name}/api/v1/policies/devices/{deviceuuid}	Returns list of all policies by device uuid.
Http Method:	GET
Format:	xml, json
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<policyWebServiceResponse>	
<messages>	
<message> 1 policy returned.</message>	Status Message. Policy count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.

</messages>	
<policies>	
<policy id="-2">	
<policyName>Default Privacy Policy</policyName>	Policy name.
<policyType>DEFAULT</policyType>	Policy Type. Either DEFAULT or ENTERPRISE.
<profileType>PRIVACY</profileType>	Profile type. Either PRIVACY, SECURITY, LOCKDOWN, or SYNC.
<status>Active</status>	Active or Inactive.
<active>true</active>	Whether the policy is active. true means Active. false means Inactive.
<defaultPolicy>false</defaultPolicy>	Deprecated.
<description>Default Privacy Policy</description>	Policy description.
<deviceCount>0</deviceCount>	This field is not applicable for this request.
<pendingCount>0</pendingCount>	This field is not applicable for this request.
<priority>1</priority>	Priority
<rules>	Policy rules, which consist of type-value pairs. The set of type-value pairs are listed in Section Policy Rules Policy Rules. The rule shown here is only an example.
<rule>	
<type>PRIVACY_SYNC_CALLLOGS</type>	Rule type

<value>store</value>	
<clientValue>off</clientValue>	
</rule>	
….	
<rules>	
</policy>	
</policies>	
</ policyWebServiceResponse >	

Apply/Remove policy for a label.

This API applies a policy to a label or removes a policy from a label.

Example:

```
https://mycore.mobileiron.com/api/v1/policies/-2?action=apply_label&label=Testlabel
```

URI: https://{host-name}/api/v1/policies/{policyid}	Returns status.
Http Method:	PUT
Format:	xml, json
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Request:	
policyid	Required. The internally generated policy ID. Use the Get Policies and Get Policies by DeviceUUID to determine a policy's ID.
action	Required. This parameter is a query parameter.

	“apply_label” – to apply the label to the policy “remove_label” – to remove the label from the policy
label	Required. Label name. This parameter is a query parameter.
Response:	
<policyWebServiceResponse>	
<messages>	
<message> Policy applied to label Android successfully. </message>	Status Message. A descriptive error message is shown if the method execution failed.
</messages>	
<policyWebServiceResponse>	

Policy Rules

An HTTP response that contains information about a policy includes a <rules> element made up of many <rule> elements.

For example:

```
<rules>
       <rule>
               <type>SYNC_HEARTBEAT_INTERVAL</type>
               <value>14</value>
               <clientValue>840</clientValue>
       </rule>
<rule>
               <type>SYNC_MULTITASK_INTERVAL</type>
               <value>15</value>
               <clientValue>15</clientValue>
       </rule>
</rules>
```

The following tables show the values of these <type> elements, their meanings, and possible values.

Note: The <clientValue> element is deprecated. Ignore its values.

Security policy rules

The following table shows the rules for security policies, listed alphabetically by the name of the <type> field.

Note: Not all the security rules apply to all device types.

For information about security policies, see the MobileIron® Administration Guide.

Security policy rule <type> field	Description	Values
EAS_BLOCK_ ANDROID_ DATA_ENC	Whether to take an action when data encryption is disabled on an Android device.	Value: true or false ClientValue: deprecated.
EAS_BLOCK_ ANDROID_ DEVICE_ ADMIN_ DEACTIVE	Whether to take an action when MobileIron detects that the device administrator privilege has been removed from the MobileIron app.	Value: true or false ClientValue: deprecated.
EAS_BLOCK_ ANDROID_OS	The version of Android below which MobileIron takes an action.	Value: An Android version number. For example: 2.3 ClientValue: deprecated.
EAS_BLOCK_ ANDROID_ ROOTED	Whether to take an action when MobileIron detects an Android device that has been rooted.	Value: true or false ClientValue: deprecated.
EAS_BLOCK_ IOS_DEVICE_ MDM_ DEACTIVE	Whether to take an action when MobileIron detects that the MDM profile has been removed from an iOS device.	Value: true or false ClientValue: deprecated.
EAS_BLOCK_ IPHONE_ DATA_ENC	Whether to take an action when data encryption is disabled on an iOS device.	Value: true or false ClientValue: deprecated.
EAS_BLOCK_ IPHONE_HW	Whether to take an action for particular iOS device models that the administrator	Value:

Security policy rule <type> field	Description	Values
	has specified as disallowed.	false – No devices are disallowed. Disallowed devices are specified in a comma-separated list of numbers. The numbers are: 1 – iPhone, original version 2 – iPhone 3G 3 – iPhone 3GS 4 – iPod touch, 1st gen 5 – iPod touch, 2nd gen 6 – iPod touch, 3rd gen 7 – iPad 16 – iPhone 4 18 – iPod touch, 4th gen 22 – iPad 2 28 – iPhone 4s ClientValue: deprecated.
EAS_BLOCK_ IPHONE_ JAILBROKEN	Whether an iOS device has been compromised (jailbroken).	Value: true – the device has been compromised. false – the device has not been compromised. ClientValue: deprecated.
EAS_BLOCK_ IPHONE_OS	The iOS version below which MobileIron takes an action.	Value: An iOS version number. Example: 3.0 ClientValue: deprecated.
EAS_BLOCK_	The number of days a device cannot	Value: A number.

Security policy rule <type> field	Description	Values
OOC_DAYS	connect to MobileIron before MobileIron takes an action.	ClientValue: deprecated.
EAS_BLOCK_ POLICY_DAYS	The specified number of days after which MobileIron takes an action when it detects that a device has not met policy requirements.	Value: A number. ClientValue: deprecated.
SEC_NCA_ ANDROID_ DATA_ENC	The action to take when data encryption is disabled on an Android device.	Value: 0 – No action. 1 – Send alert. 2 – Block ActiveSync and send alert. Other positive integers – Corresponds to a compliance action the MobileIron Core administrator created to disable or quarantine the device. MobileIron Core assigned an integer value to the action when the MobileIron Core administrator created the action. ClientValue: deprecated.
SEC_NCA_ ANDROID_ DEVICE_ ADMIN_ DEACTIVE	The action to take when the device administrator is removed from an Android device.	Value: 0 – No action. 1 – Send alert. 2 – Block ActiveSync and send alert. Other positive integers – Corresponds to a compliance action the MobileIron Core administrator created to disable or quarantine the device. MobileIron Core assigned an integer value to the action when the MobileIron Core administrator created the

Security policy rule <type> field	Description	Values
		action. ClientValue: deprecated.
SEC_NCA_ ANDROID_ ROOTED	The action to take when an Android device that has been “rooted,” that is, root access has been given to an app. A rooted Android device is also called compromised.	Value: 0 – No action. 1 – Send alert. 2 – Block ActiveSync and send alert. Other positive integers – Corresponds to a compliance action the MobileIron Core administrator created to disable or quarantine the device. MobileIron Core assigned an integer value to the action when the MobileIron Core administrator created the action. ClientValue: deprecated.
SEC_NCA_ ANDROID_SW	The action to take when the version of Android on a device is less than a specified version.	Value: 0 – No action. 1 – Send alert. 2 – Block ActiveSync and send alert. Other positive integers – Corresponds to a compliance action the MobileIron Core administrator created to disable or quarantine the device. MobileIron Core assigned an integer value to the action when the MobileIron Core administrator created the action. ClientValue: deprecated.
SEC_NCA_ APP_	The action to take when a device has violated app control rules.	Value:

Security policy rule <type> field	Description	Values
CONTROL		No value – No action. 1 – Send alert. 2 – Block ActiveSync and send alert. Other positive integers – Corresponds to a compliance action the MobileIron Core administrator created to disable or quarantine the device. MobileIron Core assigned an integer value to the action when the MobileIron Core administrator created the action. ClientValue: deprecated.
SEC_NCA_IOS_DATA_ENC	The action to take when data encryption is disabled on an iOS device.	Value: 0 – No action. 1 – Send alert. 2 – Block ActiveSync and send alert. Other positive integers – Corresponds to a compliance action the MobileIron Core administrator created to disable or quarantine the device. MobileIron Core assigned an integer value to the action when the MobileIron Core administrator created the action. ClientValue: deprecated.
SEC_NCA_IOS_HW	The action to take when an iOS device is disallowed. See EAS_BLOCK_IPHONE_HW for the list of disallowed devices.	Value: 0 – No action. 1 – Send alert. 2 – Block ActiveSync and send alert.

Security policy rule <type> field	Description	Values
		Other positive integers – Corresponds to a compliance action the MobileIron Core administrator created to disable or quarantine the device. MobileIron Core assigned an integer value to the action when the MobileIron Core administrator created the action. ClientValue: deprecated.
SEC_NCA_ IOS_ JAILBROKEN	The action to take when an iOS device has been compromised (jailbroken).	Value: 0 – No action. 1 – Send alert. 2 – Block ActiveSync and send alert. Other positive integers – Corresponds to a compliance action the MobileIron Core administrator created to disable or quarantine the device. MobileIron Core assigned an integer value to the action when the MobileIron Core administrator created the action. ClientValue: deprecated.
SEC_NCA_ IOS_MDM_ DEACTIVE	The action to take when MobileIron detects that the MDM profile has been removed from an iOS device.	Value: 0 – No action. 1 – Send alert. 2 – Block ActiveSync and send alert. Other positive integers – Corresponds to a compliance action the MobileIron Core administrator created to disable or quarantine the device. MobileIron Core assigned an integer value to the action when the

Security policy rule <type> field	Description	Values
		MobileIron Core administrator created the action. ClientValue: deprecated.
SEC_NCA_ IOS_SW	The action to take when the iOS version is below a specified level.	Value: 0 – No action. 1 – Send alert. 2 – Block ActiveSync and send alert. Other positive numbers – Corresponds to a compliance action the MobileIron Core administrator created to disable or quarantine the device. MobileIron Core assigned a number to the action when the MobileIron Core administrator created the action. ClientValue: deprecated.
SEC_NCA_ OOC_DAYS	The action to take when the device has not connected to MobileIron in a specified number of days.	Value: 0 – No action. 1 – Send alert. 2 – Block ActiveSync and send alert. Other positive integers – Corresponds to a compliance action the MobileIron Core administrator created to disable or quarantine the device. MobileIron Core assigned an integer value to the action when the MobileIron Core administrator created the action. ClientValue: deprecated.
SEC_NCA_ POLICY_DAYS	The action to take when MobileIron detects that a device has not met policy	Value:

Security policy rule <type> field	Description	Values
	requirements for the specified number of days.	0 – No action. 1 – Send alert. 2 – Block ActiveSynch and send alert. Other positive integers – Corresponds to a compliance action the MobileIron Core administrator created to disable or quarantine the device. MobileIron Core assigned an integer value to the action when the MobileIron Core administrator created the action. ClientValue: deprecated.
SECURITY_ AV_BLOCK_ DEVICES	Whether to run anti-virus scanning on the files of the device. Applicable only to Blackberry devices.	Value: off ClientValue: deprecated.
SECURITY_ BLACK_ WHITE_BMW_ ACLS	A list of App Control Rules of type Allowed or Disallowed that are enabled in the security policy. The type Allowed (white-listed) or Disallowed (black-listed) is specified by the value of SECURITY_ BLACK_WHITE_BMW_OPTION.	A comma-separated list of numbers that correspond to App Control Rules. MobileIron Core assigned a number to the App Control Rule when the MobileIron Core administrator created the rule. The list can be empty. For example: 3 3,6 ClientValue: deprecated.
SECURITY_ BLACK_ WHITE_BMW_ OPTION	Whether App Control Rules listed in SECURITY_BLACK_WHITE_BMW_ ACLS are of type Allowed (white-listed) or Disallowed (black-listed).	Value: WHITE – the App Control Rules are of type Allowed.

Security policy rule <type> field	Description	Values
		BLACK – the App Control Rules are of type Disallowed. ClientValue: deprecated.
SECURITY_ ENCRYPT_ DATA_TYPE	Whether the policy requires data encryption for each of these data types: Email, PIM, and documents.	Contains a <resourceDTOs> element for each data type. <resourceDTOs> <name>PIM</name> <resourceType>PIM</resourceType> <value>on</value> </resourceDTOs> Possible values for <name> and <resourceType>: Email, PIM, MY_ DOCUMENTS. Possible values for <value>: on, off.
SECURITY_ ENCRYPT_ DEVICE	Whether the security policy requires device encryption.	Value: on – Device encryption is required. off – Device encryption is not required. ClientValue: deprecated.
SECURITY_ ENCRYPT_ FILE_TYPE	Specifies which file types require data encryption. The possible file types are .doc, .xls, .pdf, .txt, media files, and others specified by the administrator.	A <resourceDTOs> element is specified for each of .doc, .xls, .pdf, .txt. and media files. The <name> and <resourceType> field values of each <resourceDTOs> element is either doc, xls, pdf, txt, MEDIA_FILES, or OTHER_FILE_TYPES.

Security policy rule <type> field	Description	Values
		The <value> field of each <resourceDTOs> element except OTHER_FILE_TYPES is either on or off. The <value> field of each the OTHER_FILE_TYPE <resourceDTOs> element is a space-separated list of other file types. ClientValue: deprecated.
SECURITY_ ENCRYPT_ SDCARD	Whether SD card encryption is required on the device.	Value: on – SD card encryption required. off – SD card encryption not required. ClientValue: deprecated.
SECURITY_ GRACE_ PERIOD	The period of time during which the user is still able to enter the correct password after the device has been locked.	Value: The number of minutes. ClientValue: deprecated. Note that this field is applicable only if the password is mandatory, the maximum inactivity timeout is 0, and the device is an iOS device.
SECURITY_ INACTIVITY_ TIMEOUT	The maximum amount of time to allow as an inactivity timeout.	Value: A string describing the time. Possible values are: 0 minute 1 minute 2 minutes 3 minutes 4 minutes 5 minutes

Security policy rule <type> field	Description	Values
		15 minutes 30 minutes 1 hour 1.5 hours 2 hours 12 hours 24 hours ClientValue: deprecated.
SECURITY_ MANDATORY_ BMW_ACLS	The list of App Control Rules of type Required that are enabled in the security policy.	A comma-separated list of numbers that correspond to App Control Rules of type Required. MobileIron Core assigned a number to the App Control Rule when the MobileIron Core administrator created the rule. The list can be empty. For example: 3 3,6 ClientValue: deprecated.
SECURITY_ PWD_ HISTORY	The number of passwords remembered to ensure that users define a different password.	Value: A number ClientValue: deprecated. Note that this field is applicable only if the password is mandatory.
SECURITY_ PWD_LENGTH	The minimum length for the password.	Value: A number ClientValue: deprecated.

Security policy rule <type> field	Description	Values
		Note that this field is applicable only if the password is mandatory.
SECURITY_ PWD_MAX_ AGE	The numbers of days after which the password will expire. 0 indicates no limit.	Value: The number of days. ClientValue: deprecated.
SECURITY_ PWD_MAX_ FAILED_ ATTEMPTS	The maximum number of times the user can enter an incorrect password before the device is wiped.	Value: A number. ClientValue: deprecated. Note that this field is applicable only if the password is mandatory.
SECURITY_ PWD_MIN_ COMPLEX_ CHAR	The minimum number of special characters that must be included in a password.	Value: A number ClientValue: deprecated.
SECURITY_ PWD_TYPE	Whether a mandatory password should be simple numeric input, be restricted to alphanumeric characters, or neither (that is, Don't Care).	Value: alphanumeric - restricted to alphanumeric characters. simple - restricted to simple numeric characters. simple,alphanumeric - restricted to either simple or alphanumeric characters. nc – no restrictions apply to the password characters. ClientValue: deprecated. Note that this field is applicable only if the password is mandatory.
SECURITY_ QUARANTINE_ DEVICES	Deprecated.	Deprecated.

Security policy rule <type> field	Description	Values
SECURITY_ REQUIRE_ PWD	Whether the user must enter a password before being able to access the device.	Value: on – a password is mandatory off – a password is optional ClientValue: deprecated.
SECURITY_ REQUIRE_ VPN	Deprecated.	Deprecated.
SECURITY_ VPN_PROFILE	Deprecated.	Deprecated.
SECURITY_ WIPE	The number of days before removing all data from the device if the MobileIron Client does not connect to the MobileIron Server.	Value: A number. ClientValue: deprecated.

Lockdown policy rules

The following table shows the rules for lockdown policies, listed alphabetically by the name of the <type> field.

Note: Not all the lockdown rules apply to all device types.

For information about lockdown policies, see the MobileIron® Administration Guide.

Lockdown policy rule <type> field	Description	Values
LOCKDOWN_ BLUETOOTH	Whether Bluetooth should be disabled in the event that device access must be restricted.	on – Bluetooth audio and data should not be disabled. audio - Bluetooth audio should not be disabled, but Bluetooth data should be disabled. off – Bluetooth audio and data should be disabled.

Lockdown policy rule <type> field	Description	Values
		ClientValue: deprecated.
LOCKDOWN_ CAMERA	Whether the camera should be disabled in the event that device access must be restricted.	on – the camera should not be disabled. off – the camera should be disabled. ClientValue: deprecated.
LOCKDOWN_ IRDA	Whether infrared should be disabled in the event that device access must be restricted.	on – infrared should not be disabled. off – infrared should be disabled. ClientValue: deprecated.
LOCKDOWN_ SDCARD	Whether the SD card should be disabled in the event that device access must be restricted.	on – the SD card should not be disabled. off – the SD card should be disabled. ClientValue: deprecated.
LOCKDOWN_ WIFI	Whether WIFI should be disabled in the event that device access must be restricted.	on – WIFI should not be disabled. off – WIFI should be disabled. ClientValue: deprecated.

Sync policy rules

The following table shows the rules for sync policies, listed alphabetically by the name of the <type> field..

Note: Not all the sync rules apply to all device types.

For information about sync policies, see the MobileIron® Administration Guide.

Sync policy rule <type> field	Description	Values
SYNC_ ALWAYS_ CONNECTED	Whether the MobileIron client app should remain connected to MobileIron Core during the sync interval.	on – Remain connected. off – Do not remain connected. ClientValue: deprecated.

Sync policy rule <type> field	Description	Values
SYNC_APN_ HOME_ NETWORK	Whether an Access Point Name (APN) connection type should be used in a Blackberry's home network.	on – Use an APN connection type. off – Do not use an APN connection type. ClientValue: deprecated.
SYNC_APN_ ROAMING_ NETWORK	Whether an Access Point Name (APN) connection type should be used in a Blackberry's roaming network.	on – Use an APN connection type. off – Do not use an APN connection type. ClientValue: deprecated.
SYNC_BIS_ HOME_ NETWORK	Whether a Blackberry Internet Service (BIS) connection type should be used in a Blackberry's home network.	on – Use a BIS connection type. off – Do not use a BIS connection type. ClientValue: deprecated.
SYNC_BIS_ ROAMING_ NETWORK	Whether a Blackberry Internet Service (BIS) connection type should be used in a Blackberry's roaming network.	on – Use a BIS connection type. off – Do not use a BIS connection type. ClientValue: deprecated.
SYNC_ BLOCK_ WHEN_ ROAMING	Whether to synch when the device is roaming.	on – Synchronization of all activity and content occurs even when roaming. mai – Synchronization of only voice, SMS, and data traffic occurs when roaming. roamingStatus – Synchronization is blocked when roaming. Sync only new country notification when roaming.

Sync policy rule <type> field	Description	Values
		off – Synchronization of all activity and content is blocked when roaming. ClientValue: deprecated.
SYNC_FULL_BG_MODE	Deprecated.	Deprecated.
SYNC_HEARTBEAT_INTERVAL	The maximum amount of time that the MobileIron client app will wait before sending a request to the MobileIron server to confirm that the client and server are connected.	A number in minutes. For example: 14 ClientValue: deprecated.
SYNC_INTERVAL	The frequency for starting the synchronization process between the device and the MobileIron server.	A number in minutes. For example: 240 ClientValue: deprecated.
SYNC_MIN_BATTERY_POWER	The percentage of battery power at which to synchronize files between the device and the MobileIron Server.	A number. For example: 20
SYNC_MIN_FILE_UPLOAD_BATTERY_POWER	The minimum battery level (%) to use for writing data from the device to MobileIron Core during the synchronization process.	A percentage. For example: 60 ClientValue: deprecated.
SYNC_MULTITASK_INVERVAL	The minimum duration between attempts to send iOS device details to MobileIron Core.	A number in minutes. For example:

Sync policy rule <type> field	Description	Values
		15 ClientValue: deprecated.
SYNC_ REQUIRE_ TLS	Whether to use Transport Layer Security for interactions between MobileIron Core and the MobileIron client app on the device.	on – Using TLS is required. off – Using TLS is not required. ClientValue: deprecated.
SYNC_ SDCARD	Whether to include files from removable storage devices, such as SD cards, when synchronizing files between the device and MobileIron Core.	on – Include files from removable storage. off – do not include files from removable storage. ClientValue: deprecated.
SYNC_ SERVERIP	The IP address or host name of MobileIron Core that the MobileIron client communicates with.	For example: someServerName.mydomain.com ClientValue: deprecated.

Privacy policy rules

The following table shows the rules for privacy policies, listed alphabetically by the name of the <type> field..

Not all the privacy rules apply to all device types.

For information about privacy policies, see the MobileIron® Administration Guide.

Privacy policy rule <type> field	Description	Values
PRIVACY_ APP_ MULTITASK	Whether the iOS MobileIron client periodically performs functions without user interaction.	on – Periodically wakes up. off – Does not periodically wake up. ClientValue: deprecated.

Privacy policy rule <type> field	Description	Values
PRIVACY_ EXCLUDE_ DIR	The file folders that are excluded from synchronization.	Contains a <resourceDTOs> element for each excluded folder. For example: <resourceDTOs> <name> /Windows</name> resourceType>DIR</resourceType> <value>on</value> </resourceDTOs> <name> - lists the excluded directory. <resourceType> - always has value DIR. <value> - always has value on.
PRIVACY_ LOG_APP_ ACTIVITY	Deprecated.	Deprecated.
PRIVACY_ LOG_FILE_ ACTIVITY	1. Deprecated.	Deprecated.
PRIVACY_ LOG_ WEBSITE_ ACTIVITY	Deprecated.	Deprecated.
PRIVACY_	Whether to sync information about the installed	track – Sync apps.

Privacy policy rule <type> field	Description	Values
SYNC_APPS	apps.	off – Do not sync apps. ClientValue: deprecated.
PRIVACY_ SYNC_ BOOKMARKS	Deprecated.	Deprecated.
PRIVACY_ SYNC_ CALLLOGS	Whether to collect statistics on voice calls.	store – Collect voice call statistics. off – Do not collect voice call statistics. ClientValue: deprecated.
PRIVACY_ SYNC_ CONTACTS	Whether to sync contacts.	store – Sync contacts. off – Do not sync contacts. ClientValue: deprecated.
PRIVACY_ SYNC_DATA_ LOG	Whether to sync data traffic statistics.	track – Sync data traffic. off - Do not sync data traffic.
PRIVACY_ SYNC_ DOCUMENTS	Whether to sync documents.	store – Sync documents. off – Do not sync documents. ClientValue: deprecated.
PRIVACY_ SYNC_ LOCATION	Whether to sync the location to the cell tower, the device's GPS position, or not at all.	celltower – Sync cell tower data. gps – Sync GPS data. off – Do not sync location data.
PRIVACY_ SYNC_MUSIC	Whether to sync music files.	store – Sync music files. off – Do not sync music files. ClientValue: deprecated.
PRIVACY_	Whether to sync other files not specified by other	The file extensions of the types of files

Privacy policy rule <type> field	Description	Values
SYNC_ OTHER_ MEDIA	privacy settings.	not to sync, as a comma-separated list. For example: ram,wav No <value> element means sync files of all types not specified by other privacy settings.
PRIVACY_ SYNC_ PICTURES	Whether to sync pictures.	store – Sync pictures. off – Do not sync pictures. ClientValue: deprecated.
PRIVACY_ SYNC_ SMSLOGS	Whether to collect SMS statistics, collect SMS statistics and store SMS data on the MobileIron server, or do neither.	track – Collect SMS statistics. store – Collect SMS statistics and store SMS data. off – Do not collect SMS statistics or store SMS data. ClientValue: deprecated.
PRIVACY_ SYNC_VIDEO	Whether to sync video files.	store – Sync video files. off – Do not sync video files. ClientValue: deprecated.

Application Settings

Get all Application Settings

This API returns the list of all application settings across all devices in the MobileIron system.

Example:

```
https://mycore.mobileiron.com/api/v1/appsettings
```

URI: https://{host-name}/api/v1/appsettings	Returns list of all appsettings.
Http Method:	GET
Format:	xml, json
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Request:	
type	This will allow the user to filter the Application settings by type. Application settings types - EXCHANGE, WIFI, BOOKMARK, EMAIL, VPN, GENERAL, CALDAV, SUBCAL, WEBCLIP, SCEP, APN, RESTRICTION, CERTIFICATE, MDM, PROVISIONING, CARDDAV,

	CONFIGURATION, OTHER Note: SCEP is the type used for all Certificate Enrollment settings.
Response:	
<appSettingsWebServiceResponse>	
<messages>	
<message> 1 application settting(s) returned.</message>	Status Message. Appsettings count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<appsettings>	
<appsetting id="-4">	
<name> System - iOS MDM CA Certificate</name>	Application setting name.
<description>This CA Certificate is distributed in conjunction with the system MDM profile. It is the certificate that the mobile device will trust for the purpose of accepting OTA MDM requests.</description>	Description.
<appType>CERTIFICATE</appType>	Application type
<deviceCount>2</deviceCount>	Number of devices for which the app setting is applied.
<pendingCount>1</pendingCount>	Number of devices for

	which the app setting is pending.
<quarantinedCount>0</quarantinedCount>	The number of devices that have had the corresponding configuration (i.e., profile) removed due to policy violations.
<type>SYSTEM</type>	System or Admin generated.
<clearCertificateCacheForSCEPSetting>false</clearCertificateCacheForSCEPSetting>	Not currently used.
<labels>-4</labels>	The ID of a label to which this app setting is assigned. This entry appears once for each label to which the app setting is assigned.
<labelNames>iOS</labelNames>	The displayed name of a label to which this app setting is assigned. This entry appears once for each label to which the app setting is assigned.
<properties>	Configuration properties for the appsetting
<entry>	
<key>CERT_SERIAL_NUMBER0</key>	
<value>10863527040561121251</value>	
</entry>	
<entry>	
<key>CERTPATH0</key>	

<value>/mi/files/groups/9002/99415d0e64ca8708/b34aab3122e3410e98a2ef5a078806ce</value>	
</entry>	
<entry>	
<key>ENABLE_MIRRORING</key>	This setting is to disable or enable iOS mirroring.
<value>false</value>	
</entry>	
<entry>	
<key>SUPPORT_MULTIPLE_CONSUMERS</key>	This setting is a property pertaining to SCEP settings. It indicates if the SCEP setting is being consumed by multiple app settings, for example, like Exchange and vpn. • If CACHE_KEYS=TRUE, SUPPORT_MULTIPLE_CONSUMERS is usually FALSE/NULL (both will work) • If CACHE_KEYS=FALSE,

	SUPPORT_MULTIPLE_CONSUMERS is FALSE if it is being consumed by only one app setting, and TRUE if it is being consumed by multiple apps.
<value>false</value>	
</entry>	
….	Can have multiple entries.
</properties>	
</appsetting>	
</appsettings>	
</appSettingsWebServiceResponse>	

Get Application Settings by Device UUID

This API returns the list of all application settings for a given device uuid in the MobileIron system.

Example:

```
https://mycore.mobileiron.com/api/v1/appsettings/devices/12849438-0d74-3c30-6b7d-121a3da8645d
```

URI: https://{host-name}/api/v1/appsettings/devices/{deviceuuid}	Returns list of all appsettings by device uuid.

Http Method:	GET
Format:	xml, json
Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Response:	
<appSettingsWebServiceResponse>	
<messages>	
<message> 1 application settting(s) returned.</message>	Status Message. Appsettings count is shown if the method execution is successful. A descriptive error message is shown if the method execution failed.
</messages>	
<appsettings>	

<appsetting id="-4">	
<name> System - iOS MDM CA Certificate</name>	Application setting name.
<description>This CA Certificate is distributed in conjunction with the system MDM profile. It is the certificate that the mobile device will trust for the purpose of accepting OTA MDM requests.</description>	Description.
<appType>CERTIFICATE</appType>	Application type
<deviceCount>0</deviceCount>	This field is not applicable for this request.
<pendingCount>0</pendingCount>	This field is not applicable for this request.
<type>SYSTEM</type>	System or Admin generated.
<state>67</state>	67 = ASCII "C" = Current 68 = ASCII "D" = Deleted
<clearCertificateCacheForSCEPSetting>false</clearCertificateCacheForSCEPSetting>	
<properties>	Configuration properties for the appsetting

<entry>	
<key>CERT_SERIAL_NUMBER0</key>	
<value>10863527040561121251</value>	
</entry>	
<entry>	
<key>CERTPATH0</key>	
<value>/mi/files/groups/9002/99415d0e64ca8708/b34aab3122e3410e98a2ef5a078806ce</value>	
</entry>	
….	Can have mulitiple entires.
</properties>	
</appsetting>	
</appsettings>	
</appSettingsWebServiceResponse>	

AppConnect for iOS and Android Analytics

Get Analytics for AppConnect Apps

This API returns a ZIP file that contains AppConnect analytics for a specified AppConnect app. The ZIP file contains CSV files, one for each device on which the AppConnect app has run. Each CSV file provides statistics about AppConnect app usage on a device.

The ZIP file has the following folder structure:

```
<app ID>
<first device ID>
       CSV file
       version.txt file
<second device ID>
       CSV file
       version.txt file
```

Each CSV file contains one row per app session. Each row contains:

- the start time of the foreground session, given in seconds since January 1, 1970 (Unix time). Example: 1381268056367
- the duration of the session, given in seconds

Example: 35

- the timezone where the session occurred, given in minutes from Greenwich Mean Time. Example for Pacific Daylight Time: -420
- the number of bytes the app transferred during the session (under construction)

Each version.txt file contains the version number of the AppConnect app.

Examples:

```
https://mycore.mobileiron.com/api/v1/apps/appconnect/analytics?appid=com.mycompany.myapp
```

URI: https://{host-name}/api/v1/apps/appconnect/analytics	Returns ZIP file containing usage statistics for the specified app.
Http Method:	GET
Format:	xml, json

Response Status Code:	
'404 – No Data Found'	There is no data.
'200 – OK'	Data is present and the response is returned.
Request:	
appid	Required. The app ID of the AppConnect app for which you want the analytics CSV files. For iOS AppConnect apps, the app ID is the bundle ID. For Android AppConnect apps, it is the package name.

Testing from a browser

If you use a browser to send one of the web services HTTP requests, URL encode the requests. For example, replace any white space with %20. For a plus sign (+), use %2B. For example, URL-encode Email+ as Email%2B.

For details about URL-encoding, see http://www.w3schools.com/tags/ref_urlencode.asp.

Test Client

A sample http test client implemented in Java is provided below. You can access this client on the MobileIron support site: https://support.mobileiron.com/support/clients/mobileiron-api-httpclient.zip.

1. Unzip the file, mobileiron-api-httpclient.zip . Below is the directory structure of the test client.

```
client
|-- pom.xml
`-- src
    |-- main
    |   |-- java
    |   |   `-- com
    |   |       `-- mobileiron
    |   |           |-- rs
    |   |           |   `-- client
    |   |           `-- ws
    |   |               `-- client
    |   |                   `-- http
    |   |                       |-- AlertWebServiceHttpClient.java
    |   |                       |-- AppStoreWebServiceHttpClient.java
    |   |                       |-- DMWebServiceHttpClient.java
    |   |                       |-- EASWebServiceHttpClient.java
    |   |                       |-- MAIWebServiceHttpClient.java
    |   |                       |-- SMWebServiceHttpClient.java
    |   |                       `-- WebServiceHttpClientBase.java
    |   `-- resources
    |       |-- applicationContext-miws-client.xml
    |       |-- miws-client.properties
    |       `-- miws-v1.wadl
    `-- test
        |-- java
        |   `-- com
        |       `-- mobileiron
        |           |-- rs
        |           `-- ws
        |               `-- client
        |                   `-- http
        |                       `-- WebServiceHttpClientTest.java
        `-- resources
            `-- log4j.xml
```

2. Edit the properties in miws-client.properties under src/main/resources.

webservice.hostname=<your mobileiron appliance host name or IP Address>

```
webservice.url=/api
webservice.version=/v1
user name who has the 'API' role assigned.  See Chapter- 2 Authentication
webservice.username=miadmin
Password of the above user.
webservice.password=<password>
```

3. Change directory to client and execute maven to build and run the test client.

```
cd client
mvn clean install
```

The above mvn command will do a clean build and execute the test client. The output of the tests are printed on the console and to a file named miws-test.log in the ‘client’ directory.

Change Log

Date	Description
July 17, 2019	Rebranded with new MobileIron logos.
January 18, 2019	Added note that ServiceNow can use the register, retire, wipe, lock, unlock, locate, and wakeup a device calls.
December 14, 2018	Added notifyuserbysms parameter to Register a Device call.
June 5, 2018	Changed title page to reflect 10.0.0.0.
March 6, 2018	Changed title page to reflect 9.7.0.0.
November 1, 2017	Changed title page to reflect 9.6.0.0.
August 9, 2017	Removed the following calls because MobileIron no longer supports them: • sm/certificates/upload • sm/certificates/delete • sm/certificates/list