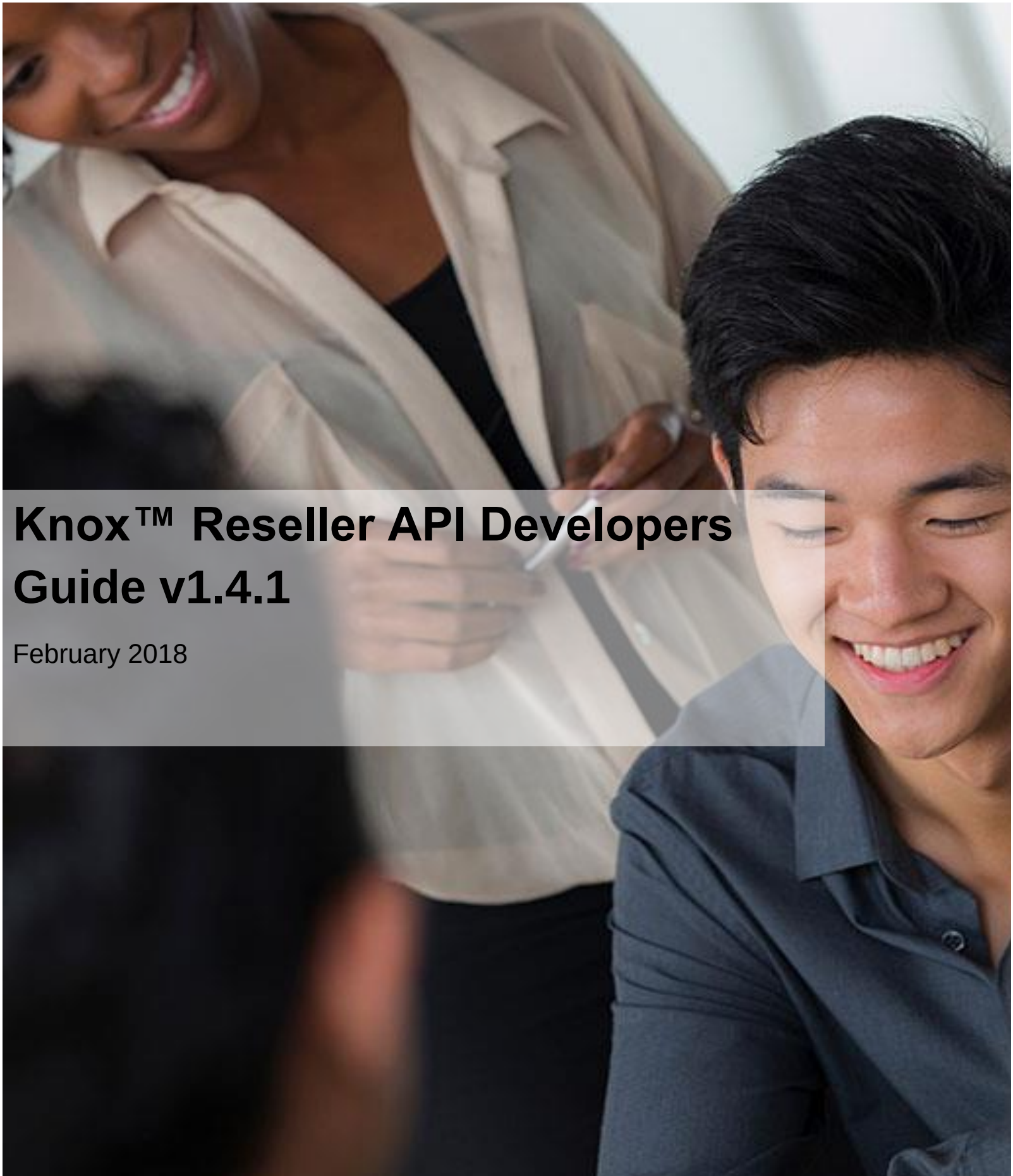
SAMSUNG Knox

Developers Guide



Knox™ Reseller API Developers Guide v1.4.1

February 2018

Copyright Notice

Document History

Date	Changes
January 2017	Version 1.2.3
July 2017	Version 1.3
September 2017	Version 1.4 – Carrier automation API version 2
February 2018	Version 1.4.1 – Result code definitions added

Contact Information

If you want to contact us about …	You have these options …
General Knox questions	Knox Portal for comprehensive information about Knox
How to get Knox	Contact Knox Sales to try Knox, start a free trial, get pricing info, or buy Knox
Technical questions	Knox Support for self-help resources like videos, guides, and FAQs If you already have a Knox portal account, log in to see all the resources available to you. If you do not have an account, you can register for one.
Other support options	Contact Knox Support

About the Knox™ Deployment Program

Knox Deployment Program resellers, distributors and carriers can upload Samsung devices on behalf of customers and provide proof of purchase for devices a customer intends to:

- Enroll into their EMM environment using Knox Mobile Enrollment (KME). For more information about KME, visit: https://www.samsungknox.com/en/solutions/mobile-enrollment.
- Configure employee devices, B2B2C devices and special purpose devices such as kiosks using Knox Configure (KC). For more information about KC, visit: https://www.samsungknox.com/en/solutions/knox-configure.

The Reseller APIs upload devices to a customer's Knox Mobile Enrollment and Knox Configure account and make them available to customers for configuration or EMM association.

This guide describes the following for program managers and developers:

- Device provisioning processes for single-tier and two-tier resellers
- API integration process
- API technical specifications

A. Device Provisioning Process for Single-Tier Resellers

A participating reseller sells devices directly to customers and uploads them to customer's KME and/or KC account using Knox Reseller APIs.

This section describes the steps and pre-requisites for an enterprise customer to get their purchased devices through a Knox deployment program participating reseller.

1. The reseller obtains a reseller ID.
    a. The reseller signs up for a Knox Reseller Portal account.
    To register for a Knox account, go to: https://www2.samsungknox.com/en/user/register/sdr.
    The Knox Reseller Portal is a cloud service that provides tools for distributors and tier 1 resellers such as generation of Reseller ID, API key, device upload, device delete, etc. A limited version of the Reseller Portal is also available for Tier 2 resellers and vendors.
    b. Once approved as a reseller, the reseller generates a reseller ID.
    c. The reseller completes the technical integration process covered in Section C below and prepares for deployment.
2. The Enterprise IT admin obtains a customer ID.
    a. IT admin creates either a Knox Configure or a Knox Mobile Enrollment account.
    To get started with Knox Configure, go to: https://www.samsungknox.com/en/solutions/knox-configure.
    To get started with KME, go to https://www.samsungknox.com/en/solutions/mobile-enrollment.
    b. IT admin generates a Knox Customer ID.
    c. The customer obtains their reseller's ID from the reseller.
    d. The customer registers their reseller's ID in their account authorizing the reseller to upload devices directly for their account.
        - The customer can optionally choose to approve each incremental upload as their reseller uploads devices.
        OR
        The customer can provide a blanket reseller approval, so devices automatically add to their customer account.

    NOTE – A customer can also automatically assign an EMM profile to a reseller in KME, so devices can always automatically be assigned to an EMM.
3. The Enterprise IT admin places an actual purchase order.
    a. The IT admin shares their Knox customer ID with their reseller.
    b. The IT admin places order with the carrier/reseller.
    c. In return, the reseller can provide an order number for the customer to track their devices.

4. The reseller completes their customer orders and uploads their devices.
    a. The reseller fulfills the customer's order by shipping the actual devices to the customer or employees directly.
    b. The reseller uploads device IMEIs and serial numbers using an automation API. Typically, the following information is required to make a successful API call:
        - Knox Customer ID
        - Reseller ID
        - Order No (Optional)
        - List of devices
5. The Enterprise Customer takes ownership of their devices.
    a. The devices automatically show up as a new upload within the customer's account.
    b. The customer does not need to approve the upload if setup up for blanket reseller upload approval, as described in Step 2.
    c. The customer selects specific devices and assigns an EMM profile or KC configuration profiles to them.
    d. If auto-setup is selected in KME, the EMM profile automatically assigns to the target devices.
6. End users unbox their assigned devices and connects them to the Internet. The device automatically locates its assigned enterprise EMM profile or configuration.

NOTE – The Google and Samsung consumer setup is available on select carriers and device models.

B. Device Provisioning Process for Two-Tier Distributors

The KDP integrating company sells devices to a mid-tier reseller who indirectly sells it to a customer. The actual purchase order comes to the mid-tier reseller. The customer sees the mid-tier reseller in their user interfaces.

The automating reseller (the distributor) provides this service to smaller resellers who cannot provide direct proof of purchase to Samsung via direct integration. The distributors should only upload devices sold via their own sales channel.

This section describes the steps pre-requisites for how an enterprise customer obtains their purchased devices through a KDP participating reseller.

1. The distributor obtains a Reseller ID
    a. The distributor signs up for the Knox Reseller portal.
    For more information on signing up as a Knox reseller, go to:
    https://www2.samsungknox.com/en/user/register.
    b. Once approved as a reseller, the distributor generates a reseller ID.
    c. The distributor completes the technical integration process described in Section C and prepares for deployment.
2. Smaller reseller obtains a Reseller ID
    a. The reseller utilizing their distributor's automation signs up for Knox Reseller portal. These small resellers are referred to as Vendors.
    For more information on signing up as a Knox reseller, go to:
    https://www2.samsungknox.com/en/user/register.
    b. The reseller generates a reseller ID (also known as Vendor ID).
    c. The reseller completes the integration process and prepares for deployment (covered in detail in the sections that follow).
3. The Enterprise Customer creates KME and/or KC accounts.
    a. Create either a Knox Configure or Knox Mobile Enrollment account.
    To get started with Knox Configure, go to: https://www.samsungknox.com/en/solutions/knox-configure. To get started with KME, go to https://www.samsungknox.com/en/solutions/mobile-enrollment.
    b. Generate a Knox Customer ID.
    c. The customer obtains their reseller's ID from their vendor (not the distributor).
        - The customer registers their reseller's ID in their account authorizing the vendor to upload devices directly for their account. The customer can optionally choose to approve each incremental upload as their vendor uploads devices.
        OR
        The customer can provide a blanket reseller approval, so devices add automatically to their customer account.

    NOTE – A customer can also automatically assign an EMM profile to a reseller in KME, so devices can automatically assign to an EMM.

d. The Enterprise Customer places an actual purchase order with a reseller.
e. The customer shares their Knox customer ID with their vendor.
f. The customer places an order with their vendor.
g. If necessary, the vendor can provide an order number for tracking.

4. The vendor creates a purchase order with one or more distributors.
    a. The vendor (also known as tier-2 reseller) places an order with the distributor, and provides the following information:
        - Customer ID
        - Vendor reseller ID
        - Order number (optional)
        - Additional purchase information (device model, date of delivery, quantity etc.).
5. The distributor fulfills the orders and uploads devices to the customer's Knox account.
    a. The distributor can fulfill their order by sending the devices to the customer or employee directly.
    b. The list of device IMEIs and serial numbers can be uploaded the customer account using the automation API. Typically, the following information is sent to the customer account.
        - Knox Customer ID
        - Distributor's Reseller ID
        - Vendor or small reseller's ID (as a vendor ID)
        - Order number (optional)
        - List of devices
    c. The reseller validates the accuracy of the Knox account device upload.
6. The Enterprise Customer takes ownership of their devices.
    a. The devices automatically list as a new upload within the customer's account.
    b. The customer does not need to approve the upload if setup up for blanket reseller upload approval.
    c. The customer selects specific devices and assigns an EMM profile or KC configuration profile.
    d. If auto-setup is selected in KME, the EMM profile is automatically assigned to the target devices.
7. End users unbox their assigned devices and connect them to the Internet. The device automatically locates its assigned Enterprise EMM profile or configuration.

    NOTE – The Google and Samsung consumer setup wizard is available on select carriers and device models.

C. API integration process

1. Pre-requisites

- Samsung creates an account
- Samsung generates a reseller ID
- Samsung generates a test API token
- The integrating reseller selects direct integration or 2-tier integration.

2. Configuring CA certificate

The integrating company shares their server's CA certificate that Samsung can whitelist for both their development and production environments.

3. API integration

1. Samsung assigns a Technical support point of contact to lead the integration effort.
2. Samsung provides process and architecture guidance during the integration process.
3. Samsung provides test data and user acceptance documents to the Integrating company.
4. The integrating company begins integration using a test account, completes phase one of user acceptance testing, and shares their results with Samsung.
5. Samsung reviews the results and provides guidance on production account creation.
6. The integrating company completes phase two testing using a production account and shares their results with Samsung.

4. User acceptance testing

This section describes the steps required by the KDP participating reseller to verify their API integration. Test data is provided for user acceptance tests for the first phase of testing.

Phase-1 testing

Verify API integration with test reseller account

a. Carrier adds some devices to customer account
b. Carrier deletes devices from customer account
c. Carrier adds deleted device to a second customer account
d. Carrier adds new device to customer account on behalf of a third party vendor or reselller (this test can be skipped if you are not supporting this feature)
e. Carrier adds two devices to customer one's account, one of which already exists in the system (under another customer account)
f. Carrier deletes two devices from customer one's account, one of which does not exist in the system at all
g. Carrier adds in invalid device (non-Samsung device)

Phase-2 testing

Conduct end-to-end testing on the production reseller account. To conduct end-to-end testing, resellers need to sign up for a reseller portal account. To register for a Knox account, go to: https://www2.samsungknox.com/en/user/register/sdr.

NOTE – The reseller must accept the terms of the Knox Deployment Program.

NOTE – If the reseller already has a KME reseller ID, they must use the exact same email they used to register their reseller in KME.

Test steps

1. Add a production test device to a customer's account and configure a test profile. At the end of the test, an MDM agent installs on the target test device.
2. Delete a device from the customer's account that is currently enrolled to mobile enrollment. At the end of the test, the device no longer appears in the customer's account.

Release and availability

1. Once the reseller finishes acceptance testing, they are ready make the devices available for customer use.
2. Based on availability, resellers can deploy their solution in multiple supported countries either by choosing multiple reseller IDs for individual countries or the same reseller ID. Currently, KDP requires two different IDs for:
    a. Region 1: North and South America
    b. Region 2: Europe, Asia (except China) and Australia

Security token management

The Reseller portal UI should be used for generating a new security token and discarding the existing token.

D.Technical API Spec

Supported devices: only devices supporting Knox Mobile Enrollment and Knox Configure services are authorized to be uploaded using this tool. For a list of supported Knox devices, go to: https://www.samsungknox.com/en/knox-platform/supported-devices/.

Format Supported

JSON

Base URI

US and European servers

For US server

https://us-be-reseller.samsungknox.com

For EU server

https://eu-be-reseller.samsungknox.com

Authentication *Message Header*

A valid token is passed in the http header (`X-WSM-API-TOKEN`).

HTTP Status Codes

Code	Description
200	OK ... on success
400	Bad Request ... for any validation error at server
401	Unauthorized ... for bad credentials
404	Not Found ... if Resource not found
409	Conflict ... Indicates the request could not be processed because of conflict in the request
415	Unsupported Media Type ... The request entity has a media type which the server or resource does not support
500	Server Error ... for any internal server error

Result Codes

These parameters are passed for all responses, with values specific to the success or failure scenario of the event.

Code	Message
2000000	SUCCESS A successful API call was made to a Samsung KME server by one of the 6 APIs.
4000000	RESOURCE_INVALID_PARAM The request received by server contains invalid parameters. Example - For the customer search api - **`/bulkenroll/reseller/customers`**, the expected parameters are resellerId, and list of Customer IDs. If the list of IDs is empty, the return is 4000000.
4000007	RESOURCE_DUPLICATE_PARAM The request received by server for the given parameters has already been received. Example - User 1 invites User 2 as an additional admin. User 1 invites User 2 again. This second request returns 4000007.
4010000	AUTHORIZATION_FAIL An API key has been inserted incorrectly, is null or has been revoked via the Knox Deployment Portal.
4040000	RESOURCE_NOT_FOUND The request received by the server contains parameters whose details cannot be found in the system. Example - User 1 and User 2 exist for a Tenant. User 1 invites User 3. User 2 deletes the invitation. When User 1 tries to retrieve invitation details, the request returns 4040000.
4090000	RESOURCE_CONFLICT The request received by the server contains parameters used already in a request already received and processed successfully.
5000000	INTERNAL_SERVER_ERROR A valid API call has been made, but structured in a way the server cannot parse, so the server cannot respond with a valid success message.
4150000	UNSUPPORTED_MEDIA_TYPE
4002100	RESELLER_TRANSACTION_ID_MISSING The Reseller Transaction ID was missed in the API call. To proceed, enter a valid Transaction ID in the API call.
4002101	RESELLER_ID_MISSING The Reseller ID was missed in the API call. To proceed, enter a valid Reseller ID within the API call.
4002102	CUSTOMER_ID_MISSING The Customer ID was missed within the API call. To proceed, enter a valid Customer ID in the API call.
4002103	RESELLER_DEVICE_LIST_MISSING IMEI, SN or MEID list missing from the API call. To proceed, enter a valid device list in the API call.
4002104	CUSTOMER_ID_AND_TRANSACTION_ID_MISSING Customer ID & Transaction ID were missed in the API call. To proceed, enter a valid Customer ID & Transaction ID in the API call.
4002105	RESELLER_DEVICE_LIST_AND_REMOVE_TRANSACTION_ID_MISSING IMEI, SN or MEID & Remove Transaction ID list were missed in the API call. To proceed, enter a valid IMEI, SN or MEID & Remove Transaction ID list within the API call
4002106	RESELLER_DEVICE_LIST_AND_REMOVE_TRANSACTION_ID_EXIST IMEI, SN or MEID & Remove Transaction ID list currently exists. Make sure the IMEI, SN or MEID used in the API call has not already been used in the Save & Delete API.
4002107	ORDER_NUMBER_TOO_LONG The order number's maximum length has been exceeded. No more than 64 alphanumeric characters are permitted for an order number.
4042100	RESELLER_NOT_FOUND No reseller found. Ensure the reseller in your query is from the same KME region as you.

4042101	RESELLER_ASSOCIATION_NOT_FOUND The customer wants to un-associate a Reseller, but no data or record of the Reseller association can be found.
4042102	RESELLER_DEVICE_NOT_FOUND IMEI, SN or MEID list cannot be found in the KME system
4042103	RESELLER_TRANSACTION_NOT_FOUND Transaction not found in the system. Verify the Transaction ID is correct and in the proper format.
4042104	CUSTOMER_NOT_FOUND No customer found. Ensure the customer in the query is from the same KME region as you.
4042105	VENDOR_NOT_FOUND Only applicable for tier 2 integrations. Verify the tier 2 reseller's KDP account has been set as a Vendor account, since by default, the account is created as a Reseller account. Contact the admin within the vendor's country of origin if an account modification ID is required
4042106	RESELLER_DEVICE_INVALID The IMEI, SN or MEID list passed to the API is in an invalid format.
4092100	RESELLER_DEVICE_ALREADY_EXISTS The IMEI, SN or MEID is already present in the KME system.
4092102	RESELLER_TRANSACTION_ALREADY_EXISTS This Reseller Transaction ID has been used previously in another transaction. Please use a different Transaction ID.
4092103	RESELLER_ASSOCIATION_ALREADY_EXISTS The customer tries associating with a reseller they are have already associated with.

1 Carrier Automation API

1.1 Save Devices

PUT	https://{server}/bulkenroll/v1/reseller/devices
Save devices to Knox Mobile Enrollment and Knox Configure.	

[Request]

Category	Key	Forced	Format	Description
Header	X-WSM-API-TOKEN	M	String	API authentication key (max = 32)
Body	transactionId	M	String	Transaction Id (64 alphanumeric character limit)
	resellerId	M	Numeric	Reseller Id (min = 1, max = 100)
	vendorId	O	Numeric	Vendor Id (min = 1, max = 100)
	customerId	M	Numeric	Customer Id (min = 1, max = 255)
	transactionType	M	String	Value can be “New” or “Refurbished”
	orderNo	O	String	An identifier tagged with every upload request. It is usually a business account number of the partner. 64 alphanumeric character limit.
	devices	M	Object	Device List
	orderTime	**O**	**Numeric**	**Order Time**
	type	M	String	Device type value can be “IMEI”,”MEID”,”SN”
	list	M	Array	Device IMEI/MEID/SN list

[Response]

Category	Key	Forced	Format	Description
Header	-	-	-	-
Body	transactionId	M	String	Transaction Id
	state	M	String	State of Transaction State value can be one of “Progress”, ”Rejected”
	code	M	Numeric	Result code
	message	M	String	Result code message
	data	O	String	Result description

[Example]

Request

```
Request URI
    -   PUT /bulkenroll/v1/reseller/devices
Request Header
    -   X-WSM-API-TOKEN : 1490BC5D8DAC4D97953261275069BAC4
Request Body
{
  "transactionId": "7500657464",
  "resellerId": "545904624",
  "customerId": "35863870",
  "transactionType": "New",
  "orderNo": "Order 31/17",
  "devices": {
    "type": "IMEI",
    "list": [
      "807212627606673",
      "21975175979444",
      "40649504931126"
    ]
  }
}
```

Response: State Progress

```
HTTP/1.1 200 OK

{
  "transactionId": "7500657464",
  "state": "Progress",
  "code": 2000000,
  "message": "SUCCESS"
}
```

Response: State Rejected

```
HTTP/1.1 400 Bad Request

{
  "transactionId": "7011874889",
  "state": "Rejected",
  "code": 4002101,
  "message": "RESELLER_ID_MISSING",
  "data": "reseller id is missing"
}
```

1.2 Delete Devices

PUT	https://{server}/bulkenroll/v1/reseller/devices/delete
Delete devices from a customer's Knox Mobile Enrollment and Knox Configure accounts.	

[Request]

Category	Key	Forced	Format	Description
Header	X-WSM-API-TOKEN	M	String	API authentication key (max = 32)
Body	transactionId	M	String	Transaction Id (64 alphanumeric character limit)
	removeTransactionId	O	String	Existing transaction Id is removed If this parameter is specified, the device list should not be included. One of these parameters is needed (min = 1, max = 64).
	transactionType	M	String	Transaction type can be "Delete" or "Returned" Default value is "Delete" Delete transaction is called to revert a wrong transaction Returned transaction occurs when an actual device is returned to be re-furbished
	resellerId	M	Numeric	Reseller Id (min = 1, max = 100)
	vendorId	O	Numeric	Vendor Id (min = 1, max = 100)
	customerId	M	Numeric	Customer Id (min = 1, max = 255)
	devices	O	Object	Device List If "removeTransactionId" is included, this should not be included. If "removeTransactionId" is not included, this should be included. Shortly, one of them should be included when you call this delete API.
	type	O	String	Value can be one of "IMEI","MEID","SN"
	list	O	Array	Device IMEI/MEID/SN list

[Response]

Category	Key	Forced	Format	Description
Header	-	-	-	-
Body	transactionId	M	String	Transaction Id
	state	M	String	State of Transaction State value can be one of "Progress", "Rejected"
	code	M	Numeric	Result code
	message	M	String	Result code message
	data	O	String	Result description

[Example]

Request

```
Request URI
   -  PUT /bulkenroll/v1/reseller/devices/delete
Request Header
   -  X-WSM-API-TOKEN : 1490BC5D8DAC4D97953261275069BAC4
```

```
Request Body
{
  "transactionId": "7500657464",
  "resellerId": "545904624",
  "customerId": "35863870",
  "transactionType": "Delete",
  "devices": {
    "type": "IMEI",
    "list": [
      "807212627606673",
      "21975175979444",
      "40649504931126"
    ]
  }
}
```

Response: State Progress

```
HTTP/1.1 200 OK

{
  "transactionId": "7500657464",
  "state": "Progress",
  "code": 2000000,
  "message": "SUCCESS"
}
```

Response: State Rejected

```
HTTP/1.1 400 Bad Request

{
  "transactionId": "7011874889",
  "state": "Rejected",
  "code": 4002101,
  "message": "RESELLER_ID_MISSING",
  "data": "reseller id is missing"
}
```

1.3 Query Transaction Status

GET	https://{server}/bulkenroll/v1/reseller/devices/status?resellerId={resellerId}&pageNum={pageNum}&pageSize={pageSize}&vendorId={vendorId}&customerId={customerId}&transactionId={transactionId}&timestamp={timestamp}

Use this API to query the success or failure of devices passed via a Save or Delete API call.

[Request]

Category	Key	Forced	Format	Description
Header	X-WSM-API-TOKEN	M	String	API Key for authentication (max = 32)

Query Parameter	resellerId	M	Numeric	Reseller Id (min = 1, max = 100)
	pageNum	O	Numeric	Page number Start from Zero(0), Default value is 0
	pageSize	O	Numeric	Page size Default value is 100
	vendorId	O	Numeric	Vendor Id Retrieve device IMEI list of specific vendor (min = 1, max = 100)
	customerId	O	Numeric	Customer Id Retrieve device IMEI list of specific customer Either Customer Id or Transaction Id should be provided (min = 1, max = 255)
	transactionId	O	String	Transaction Id Retrieve device IMEI list of specific transaction Either Customer Id or Transaction Id should be provided (min = 1, max = 64)
	timestamp	O	Long	Timestamp from which transactions are returned Format is a Unix timestamp

[Response]

Category	Key	Forced	Format	Description
Header	-	-	-	-
Body	totalCount	M	Numeric	Total count of transaction status List
	totalPage	M	Numeric	Total page of transaction status list This value is calculated by pageSize
	pageNum	M	Numeric	Current page count of transaction status list Start from Zero(0)
	transactions	O	Array	Transaction list
	transactionId	M	String	Transaction Id
	removeTransactionId	O	String	Existing Transaction Id is removed This value exists only when transaction type is “Delete” If this value exists, device list will not be included
	state	M	String	State of Transaction State value can be one of Progress”, ”Complete”
	type	M	String	Transaction type Transaction type can be “Put” or “Delete”
	orderNo	O	String	An identifier tagged with every upload request. It is usually a business account number of the partner. 64 alphanumeric character limit.
	orderTime	**O**	**Numeric**	**Order Time**
	code	O	Numeric	Result code
	message	O	String	Result code message
	data	O	String	Result description
	devices	O	Array	Successfully saved or deleted Device List
	imei	O	String	Device IMEI
	meid	O	String	Device MEID
	serialNumber	O	String	Device serial number
	code	O	Numeric	Error code
	message	O	String	Error code message
	data	O	String	Error description

[Example]

Request

Request URI

- GET /bulkenroll/v1/reseller/devices/status?resellerId=545904624&customerId=7853146381

Request Header

- X-WSM-API-TOKEN : 1490BC5D8DAC4D97953261275069BAC4

Response: Status Progress

```
HTTP/1.1 200 OK

{
  "totalCount": 2,
  "totalPage": 1,
  "pageNum": 0,
  "transactions": [
    {
      "transactionId": "7500657464",
      "state": "Progress",
      "type": "Put"
    },
    {
      "transactionId": "7500657462",
      "state": "Progress",
      "type": "Put"
    }
  ]
}
```

Response: Status Complete

```
HTTP/1.1 200 OK

{
  "totalCount": 2,
  "totalPage": 1,
  "pageNum": 0,
  "transactions": [
    {
      "transactionId": "7500657464",
      "state": "Complete",
      "type": "Put",
     "orderNo": "Order 31/17",

      "devices": [
        {
          "imei": "807212627606673"
        },
        {
          "imei": "21975175979444"
        },
        {
```

```
          "imei": "40649504931126",
          "code": 4092100,
          "message": "RESELLER_DEVICE_ALREADY_EXISTS",
          "data": "device imei[097436145498341] already exists"
        }
      ]
    },
    {
      "transactionId": "7500657462",
      "state": "Complete",
      "type": "Put",
      "devices": [
        {
          "imei": "407212627606673"
        },
        {
          "imei": "91975175979444"
        }
      ]
    }
  ]
}
```

1.4 Retrieve a List of Devices

GET	https://{server}/bulkenroll/v1/reseller/devices?resellerId={resellerId}&pageNum={pageNum}&pageSize={pageSize}&vendorId={vendorId}&customerId={customerId}&timestamp={timestamp}
Retrieve list of devices	

[Request]

Category	Key	Forced	Format	Description
Header	X-WSM-API-TOKEN	M	String	API Key for authentication (max = 32)
Query Parameter	resellerId	M	Numeric	Reseller Id (min = 1, max = 100)
	pageNum	O	Numeric	Page number Start from Zero(0), Default value is 0
	pageSize	O	Numeric	Page size Default value is 100
	vendorId	O	Numeric	Vendor Id Retrieve device IMEI list of specific vendor (min = 1, max = 100)
	customerId	O	Numeric	Customer Id Retrieve device IMEI list of specific customer Either Customer Id or Transaction Id should be provided (min = 1, max = 255)
	transactionId	O	String	Transaction Id Retrieve device IMEI list of specific transaction Either Customer Id or Transaction Id should be provided (min = 1, max = 64)
	timestamp	O	Numeric	Timestamp from which devices need to be returned Format is a Unix timestamp

[Response]

Category	Key	Forced	Format	Description
Header	**-**	-	-	-
Body	totalCount	M	Numeric	Total device list count
	totalPage	M	Numeric	Total device list pages This value is calculated by page size
	pageNum	M	Numeric	Current page of Device List Start from Zero(0)
	devices	O	Array	Device List
	imei	O	String	Device IMEI
	meid	O	String	Device MEID
	serialNumber	O	String	Device serial number
	orderNo	O	String	An identifier tagged with every upload request. It is usually the partner's business account number. 64 alphanumeric character limit.
	orderTime	**O**	**Numeric**	**Order Time**

[Example]

Request

Request URI

- GET /bulkenroll/v1/reseller/devices?resellerId=545904624&customerId=7853146381

Request Header

- X-WSM-API-TOKEN : 1490BC5D8DAC4D97953261275069BAC4

Response: Success


```
{
  "totalCount": 3,
  "totalPage": 1,
  "pageNum": 0,
  "devices": [
    {
      "imei": "807212627606673"
    },
    {
      "imei": "21975175979444"
    },
    {
      "imei": "40649504931126"
     "orderNo”:”Order 31/05/2017"

    }
  ]
}
```

1.5 Retrieve Resellers Details

POST	https://{server}/bulkenroll/v1/reseller/resellers
Retrieve a list of resellers for given IDs. Useful for checking if a reseller ID or Vendor ID is correct.	

[Request]

Category	Key	Forced	Format	Description
Header	X-WSM-API-TOKEN	M	String	API Key for authentication (max = 32)
Body	ids	M	Array	List of Reseller Ids, maximum 500 Ids can be included in a single request.
	resellerId	M	Numeric	Reseller Id (min = 1, max = 100)

[Response]

Category	Key	Forced	Format	Description
Header	-	-	-	-
	users	M	Array	Reseller list
	invalidusers	M	Array	Reseller list
	companyId	M	String	Reseller Id
	companyName	M	String	Reseller company name
	address	O	String	Reseller company address
	code	O	Numeric	Result code
	message	O	String	Result code message
	data	O	String	Result description

Error codes handled: `4042100, 4092103 and 4002108`; refer to prior *Result codes* section.

[Example]

Request

```
Request URI
   -  POST /bulkenroll/v1/reseller/resellers
Request Header
   -  X-WSM-API-TOKEN : 1490BC5D8DAC4D97953261275069BAC4
Request Body
{
    "resellerId": "545904624",
    "ids": [
      "545904624",
      "254590462",
      "254590467"
    ]
   -  }
```

Response: Success

```
{
  "users": [
    {
      "companyId ": "545904624",
      "companyName": "ABDC",
```

```
      "address": "245 Park Avenue, Sunnyvale, CA"
    },
    {
      "companyId ": "254590462",
      "companyName": "XYZ",
      "address": "601 Fermont Avenue, Sunnyvale, CA"
    }
  ],
  “invalidusers” : [
    {
      "companyId ": "254590467",
       "code": 4042100,
       "message": " RESELLER_NOT_FOUND ",
       "data": "Reseller not found"
    }
  ]
}
```

1.6 Retrieve Customer Details

POST	https://{server}/bulkenroll/v1/reseller/customers
Retrieve customer information for a list of given IDs. Useful for a reseller to validate customer IDs before making API calls.	

[Request]

Category	Key	Forced	Format	Description
Header	X-WSM-API-TOKEN	M	String	API key for authentication (max = 32)
Body	ids	M	Array	List of customer Ids. Maximum 500 Ids can be requested in single request.
	resellerId	M	Numeric	Reseller Id (min = 1, max = 100)

[Response]

Category	Key	Forced	Format	Description
Header	**-**	**-**	**-**	**-**
	users	M	Array	Reseller list
	invalidusers	M	Array	Reseller list
	companyId	M	String	Customer Id
	companyName	M	String	Customer company name
	address	O	String	Customer company address
	code	O	Numeric	Result code
	message	O	String	Result code message
	data	O	String	Result description

Error codes handled: `4042104, 4042104, 4092104` **and** `4002108;` refer to prior *Result codes* section.

[Example]

Request

```
Request URI
   -  POST /bulkenroll/v1/reseller/customers
Request Header
   -  X-WSM-API-TOKEN : 1490BC5D8DAC4D97953261275069BAC4
Request Body
{
    "resellerId": "545904624",
    "ids": [
      "245904624",
      "954590462",
      "122438783"
    ]
   -  }
```

Response: Success

```
{
  "users": [
    {
      "companyId ": "245904624",
```

```
      "companyName": "ZPM",
      "address": "245 RMZ World ,Bangalore , INDIA"
    },
    {
      "companyId ": "954590462",
      "companyName": "RTG",
      "address": "601 Embassy Park, Bangalore, INDIA "
    },
  ] ,
  “invalidusers” : [
    {
      "companyId ": "122438783",
      "code": 4042104,
      "message": " CUSTOMER_NOT_FOUND ",
      "data": "Customer not found"
    }
  ]
}
```

2 Carrier Automation API – Version 2

2.1 Save Devices Version 2

PUT	https://{server}/bulkenroll/v2/reseller/devices
Save devices to KME and KC and permit multiple order numbers per upload	

Request]

Category	Key	Forced	Format	Description
Header	X-WSM-API-TOKEN	M	String	API Key for authentication
Body	transactions	M	Object	Array of Transactions. Currently only supports single transaction.

[Transaction object]

Category	Key	Forced	Format	Description
Transaction	transactionId	M	String	Transaction Id
	resellerId	M	Numeric	Reseller Id
	vendorId	O	Numeric	Vendor Id
	customerId	M	Numeric	Customer Id
	transactionType	M	String	Type of Transaction Value can be one of “New”, “Refurbished”
	devices	M	Object	Array of Devices

[Device object]

Category	Key	Forced	Format	Description
Device	type	M	String	Type of Device Value can be one of “IMEI”,”MEID”,”SN”
	list	M	Object	Array of device information

[List object]

Category	Key	Forced	Format	Description
Device Information	deviceIdentifier	M	String	IMEI or MEID or SN value
	orderNo	O	String	An identifier tagged with every upload request. It is usually a business account number of the partner.
	orderTime	O	Numeric	Number of milliseconds

[Response]

Category	Key	Forced	Format	Description
Header	-	-	-	-
Body	transactionId	M	String	Transaction Id
	state	M	String	State of Transaction State value can be one of “Progress”, ”Rejected”
	code	M	Numeric	Result Code
	message	M	String	Result Code Message
	data	O	String	Result Description

[Example]

Request

```
Request URI
   -  PUT /bulkenroll/v2/reseller/devices
Request Header
   -  X-WSM-API-TOKEN : 1490BC5D8DAC4D97953261275069BAC4
Request Body
{
  "transactions": [
  {
   "transactionId": "7500657464",
   "resellerId": "545904624",
   "customerId": "35863870",
   "transactionType": "New",
   "devices": {
     "type": "IMEI",
     "list": [
       {
        "deviceIdentifier":"807212627606673",
        "orderNo":"abc",
        "orderTime":"1503966152120"
       },
       {
        "deviceIdentifier":"236512627606673",
        "orderNo":"def"
       },
       {
        "deviceIdentifier":"738512627606673",
        "orderTime":"1578466152120"
       }
     ]
   }
  }
```

```
HTTP/1.1 200 OK

{
  "transactionId": "7500657464",
  "state": "Progress",
  "code": 2000000,
  "message": "SUCCESS"
}
```

Response : State Rejected

```
HTTP/1.1 400 Bad Request

{
  "transactionId": "7011874889",
  "state": "Rejected",
  "code": 4002101,
```

```
  "message": "RESELLER_ID_MISSING",
  "data": "reseller id is missing"
}
```

2.2 Query Transaction Status Version 2

This is a version 2 API for Transaction Query status. This API is called when devices are uploaded in a transaction using the version 2 upload API. This API has an undefined behavior if called for transactions uploaded through any other API.

GET	**https://{server}/bulkenroll/v2/reseller/devices/status?resellerId={resellerId}&pageNum={pageNum}&pageSize={pageSize}&vendorId={vendorId}&customerId={customerId}&transactionId={transactionId}&timestamp={timestamp}**
Use this API to query the success and failure of devices passed via a Save Version 2 API call.	

[Request]

Category	Key	Forced	Format	Description
Header	X-WSM-API-TOKEN	M	String	API Key for authentication
Query Parameter	resellerId	M	Numeric	Reseller Id
	pageNum	O	Numeric	Page Number Start from Zero(0), Default value is 0
	pageSize	O	Numeric	Page Size Default value is 100
	vendorId	O	Numeric	Vendor Id Retrieve Device IMEI list of specific vendor
	customerId	O	Numeric	Customer Id Retrieve Device IMEI list of specific customer Either Customer Id or Transaction Id should be provided
	transactionId	O	String	Transaction Id Retrieve Device IMEI list of specific transaction Either Customer Id or Transaction Id should be provided
	timestamp	O	Long	Timestamp from which transactions need to be returned Format is Unix Timestamp

[Response]

Category	Key	Forced	Format	Description
Header	**-**	-	-	-
Body	totalCount	M	Numeric	Total count of Transaction Status List
	totalPage	M	Numeric	Total page of Transaction Status List This value will be calculated by pageSize
	pageNum	M	Numeric	Current page of Transaction Status List Start from Zero(0)
	transactions	O	Array	List of Transactions

[Transaction]

Category	Key	Forced	Format	Description
Transaction	transactionId	M	String	Transaction Id
	removeTransactionId	O	String	Existing Transaction Id is removed
	state	M	String	State of Transaction State value can be one of “Progress”, ”Complete”
	type	M	String	Transaction Type Transaction Type can be one of “Put”
	resellerId	M	Numeric	Reseller Id
	vendorId	O	Numeric	Vendor Id
	customerId	O	Numeric	Customer Id
	data	O	String	Result Description
	devices	O	Array	List of Devices saved

[Device]

Category	Key	Forced	Format	Description
Device	imei	O	String	Device IMEI
	meid	O	String	Device MEID
	serialNumber	O	String	Device Serial Number
	code	O	Numeric	Error Code
	message	O	String	Error Code Message
	data	O	String	Error Description
	orderNo	O	String	An identifier tagged with every upload request. It is usually a business account number of the partner. If device has an error, will not be present.
	orderTime	O	Numeric	Number of milliseconds describing order time. If device has an error, will not be present.

[Example]

Request

```
Request URI
   - GET
     /bulkenroll/v2/reseller/devices/status?resellerId=545904624&customerId=7853146381
Request Header
   - X-WSM-API-TOKEN : 1490BC5D8DAC4D97953261275069BAC4
```

Response : Status Progress

```
HTTP/1.1 200 OK

{
  "totalCount": 2,
  "totalPage": 1,
  "pageNum": 0,
```

```
  "transactions": [
    {
      "transactionId": "7500657464",
      "state": "Progress",
      "type": "Put"
    },
    {
      "transactionId": "7500657462",
      "state": "Progress",
      "type": "Put"
    }
  ]
}
```


Response : Status Complete


```
HTTP/1.1 200 OK

{
  "totalCount": 2,
  "totalPage": 1,
  "pageNum": 0,
  "transactions": [
    {
      "transactionId": "7500657464",
      "state": "Complete",
      "type": "Put",
      "devices": [
        {
          "imei": "807212627606673",
          "orderNo": "abc",
          "orderTime": "1503966152120"
        },
        {
          "imei": "21975175979444"
        },
        {
          "imei": "40649504931126",
          "code": 4092100,
          "message": "RESELLER_DEVICE_ALREADY_EXISTS",
          "data": "device imei[097436145498341] already exists"
        }
      ]
    },
    {
      "transactionId": "7500657462",
      "state": "Complete",
      "type": "Put",
      "devices": [
        {
          "imei": "407212627606673",
          "orderNo": "def"
        },
        {
```

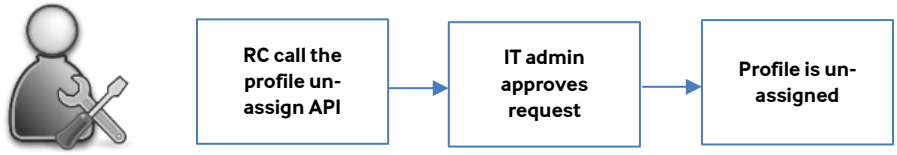
```
    "imei": "91975175979444"
  }
```

Appendix A

Warranty and replacement (in proposal state)

This section describes the current design direction for device warranty and replacement. Resellers are encouraged to verify all their use cases are covered, and provide as much information as they can about use cases that do not fit the model.

1. **Un-assign the KC/KME profile**



1.1 Profile un-assigned by Samsung Admin or Technical Support

Samsung Admin or Technical Support can request profile un-assign from their SA portal accounts.

a. Samsung Admin or Technical Support requests a profile be un-assigned
b. The profile is removed from the device
c. A notification email is sent to IT admin
d. The transaction is recorded in the activity log

Status=“Profile unassigned”

1.2 Profile re-assigned by Samsung Admin or Technical Support

a. Samsung Admin or Technical Support requests profile re-assign.
b. The profile is restored to the device
c. A notification email is sent to IT admin
d. The transaction is recorded in the activity log

Status=“Profile assigned”

1.3 Profile un-assigned by Repair Center

Repair center (RC) operators can use a profile un-assign API to request the temporary removal of a profile from a device.

a. Repair centers call the profile un-assign API with: IMEI, case ID, repair center email, and notes. If the end-user returned the device, an end-user email is provided.

   A web page is also provided for repair centers.

b. Once the API is called:

   - If the RC is on file:

     A notification email is sent to the IT admin and the profile is un-assigned

   - If the RC is not on file:

     A notification email is sent to the IT admin with a link to approve the un-assign request.

     Once approved, the profile is removed from the device.

     The device Status is updated as:

     Status="Profile unassigned" Notes: <Case ID> in KC and/or KME.

c. An email is sent to the repair center with a link to confirm the device return.
d. Once a tracking number is entered, the old profile is re-assigned. An email is sent to the IT admin to notify the device is sent.

   Status="Profile assigned" – Notes: <Case ID>

1.4 Profile re-assigned by Repair Center

a. Repair centers call the profile re-assign API with: IMEI, case ID, repair center email, and notes.

   A web page is also provided for repair centers

b. Once API is called:

   - If the RC is on file:

     A notification email is sent to the IT admin and the profile is re-assigned

   - If the RC is not on file:

     An email is sent to IT the admin with a link to approve the profile re-assign request

     Once approved, the old profile is re-assigned to the device

     The device Status is updated as:

     Status="Profile assigned" Notes: <Case ID> in KC and/or KME.

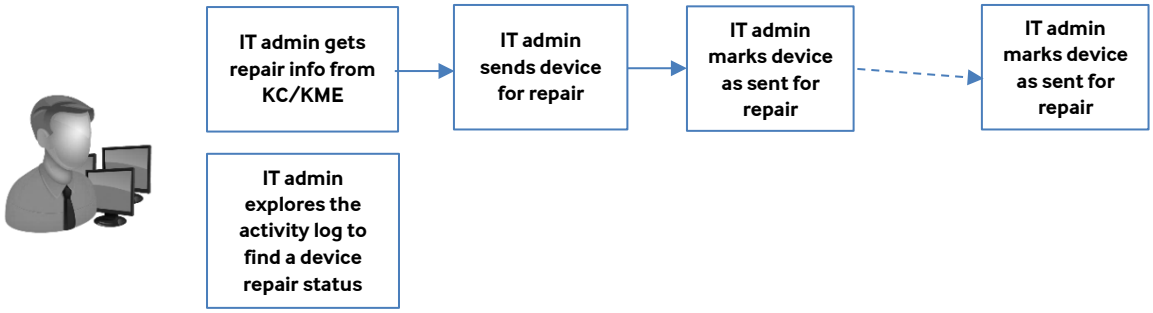
2. Setup repair centers

Resellers can register their repair centers within the Reseller Portal and provide the following information:

- Repair center contact information
- Device return instructions
- Services
- Repair or replace instructions

Once a repair center is registered, IT admins can access repair and replacement data via KC or KME.

3. Device repair status



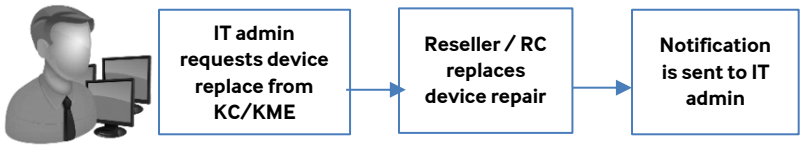
IT admins can view the device repair information that resellers configured for their region in their KC/KME accounts. The information includes instructions and links to repair center (RC) and reseller device repair portals.

IT admin can flag a device as sent for repair.
Status="Sent for repair" Notes: <Case ID> in KC and/or KME.
IT admin can flag a device as repaired.
Status="Assigned" Notes: <Case ID> in KC and/or KME.
IT admin can find all device status information in the activity log including repair information.

4. Replace the device

If the RC profile is configured and the RC supports device replacement, RC and IT admin can request device replacement from KC/KME.

4.1 Requested by IT Admin

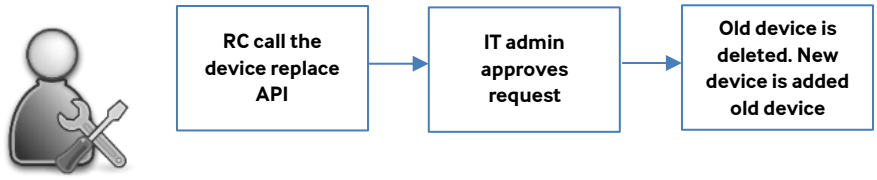


a. IT Admin requests device replace from KC/KME and provides old IMEI, new IMEI, case ID, repair center email, and notes.

b. An email is sent to the Reseller / RC email on file with a request to replace the device (add new IMEI to customer's account, remove old IMEI) and a confirmation link.

c. Once link is called, an email is sent to IT admin to confirm device upload and old device delete.

4.2 Requested by RC



a. Repair centers call the replace device API with: old IMEI, new IMEI, case ID, repair center email, and notes.

   A web page is also provided for repair centers.

b. Once API is called, an email is sent to IT admin with a link to approve the device replace request.

c. Once approved, the old device is deleted. The new device is added using the old device's profile.

   Old device Status="Deleted" Notes: <Case ID> in KC and/or KME.

   New device Status="Assigned" Notes: <Case ID> in KC and/or KME.

Appendix B

Knox Configure

Knox Configure (KC) is a cloud-based service that empowers both large to medium enterprises and B2B2C customers to configure, customize, and automate the enrollment of Samsung devices purchased from authorized Samsung resellers. To get started with Knox Configure, go to: https://www.samsungknox.com/en/solutions/knox-configure.

Knox Mobile Enrollment

Knox Mobile Enrollment (KME) is a tool to streamline the initial setup and enrollment of corporate-owned and employee-owned devices to an EMM. To get started with KME, go to https://www.samsungknox.com/en/solutions/mobile-enrollment.

About Samsung Electronics Co., Ltd.

Samsung Electronics Co., Ltd. is a global leader in technology, opening new possibilities for people everywhere. Through relentless innovation and discovery, we are transforming the worlds of televisions, smartphones, personal computers, printers, cameras, home appliances, LTE systems, medical devices, semiconductors and LED solutions. We employ 236,000 people across 79 countries with annual sales exceeding KRW 201 trillion.

For more information about Samsung Knox, visit http://www.samsungknox.com/.

Samsung Electronics Co., Ltd.
416, Maetan 3-dong, Yeongtong-gu
Suwon-si, Gyeonggi-do 443-772, Korea