SAMSUNG Knox Documentation

Knox Guard API reference v1.1.3

The Knox Guard API servers provide both an EU and US connection. Besides host location they support the same REST API functionalities.

Region	Production	Development
EU	https://eu-kcs-api.samsungknox.com/kcs/v1.1/kg	https://eu-kcs-api.samsungknox.co integration
US	https://us-kcs-api.samsungknox.com/kcs/v1.1/kg	https://us-kcs-api.samsungknox.co integration

Knox Guard provides both a test and development URL to create and manage your cloud service. When first building your cloud interface, use the development server for a light weight Knox Guard experience with all the REST APIs enabled. Ensure that you have followed all the steps mentioned in authenticating your key, before you begin development. Once you are ready to deploy your solution, use the production server to create a scalable product.

Common parameters

Device identifier

There are three ways to identify a device.

Parameter	Description
objectId	Unique ID generated by Knox Guard when you approve a device.
deviceUid	Device IMEI or Serial number.
approveId	Custom ID that you specify when you approve a device. This can be an ID used by a financial institution to identify a customer or approved loan.

You can use these IDs to identify a device to lock, unlock, blink, message, or cancel a request.

Device state

For more information as well as a diagram about the life cycle of a device, see How Knox Guard works .

State	Description	External State(on Console)
Pending	The device is pending when a reseller uploads the device to a customer. Customers can then see the device in the list and reject or accept it.	Pending
Rejected	When a customer rejects the device, it's state will change to rejected.	Rejected
Accepted	Once accepted, a Knox Guard agent on the client side is triggered. If auto accept is on, the devices' state is changed from "Pending" to "Accepted" automatically.	Activating
Enrolled	This state allows the device to be controlled by Knox Guard to lock, unlock, send a message, or blink a device.	Active
Exchanging	Status of a device, which is undergoing maintenance, being replaced with a new	Exchanging

State	Description	External State(on Console)
	device/IMEI number. This requires Samsung customer support assistance and is for tracking purposes only.	
Resetting	A device will report the resetting state if a license has expired or been deleted. Once reported, it will transition to the pending state	Resetting
StartingReminder	The state is changed to "StartingReminder" when a customer sends the start-reminder event to the client.	Starting Reminder
StoppingReminder	The state is changed to "StoppingReminder" when a customer sends the stop-reminder event to the client.	Stopping Reminder
Blinked	Reminder is activated on the client.	Reminder On
Locking	The state is changed to "Locking" when a customer sends the lock event to the client.	Locking
Locked	Confirmed that the device is locked.	Locked
Unlocking	The state is changed to "Unlocking" when a customer sends the unlock event to the client.	Unlocking
Completing	The state is changed to "Completing" when a customer sends the complete event to the client.	Completing
Completed	Device is out of control of Knox Guard anymore. customer can complete when the loan is over.	Completed

License state

State	Description	External state(on console)
Registered	The license has been created but has yet to be activated.	Registered
Active	The activation period has started and the license is still valid.	Active
Expired	License has surpassed the expiration date and is no longer valid.	Expired

Paging

APIs that return a list of objects are queried using the following paginated syntax.

```
{
  "pageNum" : 0,
  "pageSize" : 25
}
```


Use pageNum to specify the page to return, which starts at 0. Use pageSize to specify the number of objects to return at once. If you get fewer than this number of objects, that is it. If you get a full page, resubmit the query with pageNum="1", then pageNum="2", etc.

Rate limiting of API calls

A "rate limit" is the number of API calls that can be made within a given time period.

It helps us provide reliable and scalable API services to protect your operation from erratic and unpredictable spikes in the rate of requests.

Knox Guard sets the rate limit such that an API can be called up to 100 times/second from the same IP address, same user under the same Knox Guard tenant, or the same API token.

If your application exceeds the request limits detailed above, the Knox Guard API will return a 429 "Too Many Requests" error message and the API calls will need to be made again.

> **NOTE** — The rate limits come into effect on Jan 16th, 2023.

Basic

Retrieve information about your Knox Guard solution such as the access token status or troubleshooting information about a previous REST API.

Check Authorization

Check the authorization key and return information about the account.

HEADER PARAMETERS

`x-knox-apitoken` `required`	string The authentication token used to verify the REST API and specify the account. Generate a JWT using the Knox Cloud APIs token request . > *This parameter will need to be generated every 30 minutes.*
`x-knox-transactionId`	string This is an optional user generated unique alpha numeric code. Please create a different code for each request which can be used for debugging. /kcs/v1.1/kg/transaction.

Responses

> 200

OK

> 401

Unauthorized

— 403

Forbidden

— 404

Not Found

GET /kcs/v1.1/kg/authorization

Request samples

Curl

Copy

```
curl -X GET --header 'Accept: application/json' --h
```

Response samples

200 401

Content type

application/json

Copy Expand all Collapse all

```
{
  "result": "string",
```

```
  - "user": {
        "companyName": "string",
        "country": "string",
        "email": "string",
        "name": "string",
        "phoneNumber": "string"
    }
}
```


Get Async result

Get the result of an asynchronous API request, based on the unique transaction ID for the request.

HEADER PARAMETERS

x-knox-apitoken required	string The authentication token used to verify the REST API and specify the account. Generate a JWT using the Knox Cloud APIs token request . *This parameter will need to be generated every 30 minutes.*
x-knox-transactionId	string This is an optional user generated unique alpha numeric code. Please create a different code for each request which can be used for debugging. /kcs/v1.1/kg/transaction.

REQUEST BODY SCHEMA: application/json

Your previously unique alpha numeric code you wish to look up.

transactionId required	string Unique ID that you provided in the header of the asynchronous API request that you want the result for.

Responses

> 200

OK

— 201

Created

> 401

Unauthorized

— 403

Forbidden

— 404

Not Found

POST /kcs/v1.1/kg/transaction

Request samples

Payload | Curl

Content type

application/json

Copy

```
{
    "transactionId": "string"
}
```

Response samples

200 | 401

Content type

application/json

Copy Expand all Collapse all


```
{
    "count": 0,
    "fail": 0,
  - "resultList": [
     + { … }
    ],
    "success": 0,
    "transactionId": "string"
}
```


Upload

Upload devices to Knox Guard. This API was designed to support uploading devices to KG directly not through Knox Reseller Portal.

> *This functionality is only available upon request. Please contact your Knox Guard administrator to enable this feature.*

Upload Devices

Upload devices to Knox Guard. To check if the device uploaded to KG, use the Get upload information /kcs/v1.1/kg/devices/uploads/{uploadId}.

HEADER PARAMETERS

x-knox-apitoken required	string The authentication token used to verify the REST API and specify the account. Generate a JWT using the Knox Cloud APIs token request . > *This parameter will need to be generated every 30 minutes.*
x-knox-transactionId	string

	This is an optional user generated unique alpha numeric code. Please create a different code for each request which can be used for debugging. /kcs/v1.1/kg/transaction.

REQUEST BODY SCHEMA: application/json

deviceList > required	Array of objects (deviceListAPI) A list of devices to be uploaded. List size min = 0, max = 10,000
autoAccept	boolean Set auto-accept configuration for the uploaded devices. User must have the corresponding role permission to set this value to TRUE else this will default to FALSE.
autoLock	boolean Set the auto-lock policy for uploaded device. User must have the corresponding feature and role permission to set this value to TRUE else this will default to FALSE.
applySimControl	boolean Set the SIM policy for the uploaded device. SIM Policy will be applied upon enrollment. User must have the corresponding feature and role permission to set this value to TRUE else this will default to FALSE.
enableBlockFactoryReset	boolean Set the factory reset policy for the uploaded device. Factory reset function will be blocked upon enrollment. User must have the corresponding feature and role permission to set this value to TRUE else this will default to FALSE.

Responses

> 200

OK

> 401

Unauthorized

POST /kcs/v1.1/kg/devices/uploads

Request samples

Payload

Content type

application/json

Copy Expand all Collapse all


```
{
 - "deviceList": [
    + { … }
   ],
   "autoAccept": true,
   "autoLock": true,
   "applySimControl": true,
   "enableBlockFactoryReset": true
}
```


Response samples

200 401

Content type

application/json

Copy Expand all Collapse all


```
{
   "result": "SUCCESS",
   "uploadID": "fcc5e33d-9f7f-4891-8c92-95f8978e9fb1",
   "uploadStatus": "Progress",
   "totalDeviceCount": 3,
```

```
    "validDeviceCount": 1,
    "invalidDeviceCount": 0,
  - "invalidReasons": [
     + { … }
    ]
}
```


Get device uploads

Check the status of the device upload action. This will return not just the results of the POST /kcs/v1.1/kg/devices/uploads API but devices that have been uploaded through any other method as well. Please note that this API provide the 'upload' result only not the individual device state.
To check if the device uploaded to KG, use the Get upload information /kcs/v1.1/kg/devices/uploads/{uploadId}

QUERY PARAMETERS

`pageNum`	integer <int32> Default: 0 The page number to get, which must be >= 0.
`pageSize`	integer <int32> Default: 1000 The number of items in a page, which must be >0 and <= 1000.

HEADER PARAMETERS

`x-knox-apitoken` required	string The authentication token used to verify the REST API and specify the account. Generate a JWT using the Knox Cloud APIs token request . > *This parameter will need to be generated every 30 minutes.*
`x-knox-transactionId`	string This is an optional user generated unique alpha numeric code. Please create a different code for each request which can

be used for debugging.
/kcs/v1.1/kg/transaction.

Responses

> 200

OK

> 401

Unauthorized

GET /kcs/v1.1/kg/devices/uploads

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
    "result": "string",
    "totalCount": 0,
  - "uploads": [
     + { … }
    ],
  - "error": {
        "code": 0,
        "message": "string",
        "reason": "string"
    }
}
```

Get upload information

Get the device information by uploadId

PATH PARAMETERS

`uploadId` required	string Upload id for the device upload.

QUERY PARAMETERS

`pageNum`	integer <int32> Default: 0 The page number to get, which must be >= 0.
`pageSize`	integer <int32> Default: 1000 The number of items in a page, which must be >0 and <= 1000.

HEADER PARAMETERS

`x-knox-apitoken` required	string The authentication token used to verify the REST API and specify the account. Generate a JWT using the Knox Cloud APIs token request . *This parameter will need to be generated every 30 minutes.*
`x-knox-transactionId`	string This is an optional user generated unique alpha numeric code. Please create a different code for each request which can be used for debugging. /kcs/v1.1/kg/transaction.

Responses

> 200

OK

> 401

Unauthorized

```
GET   /kcs/v1.1/kg/devices/uploads/{uploadId}
```

Response samples

200 401

Content type
application/json

Copy Expand all Collapse all


```
{
  "result": "string",
  "status": "string",
- "devices": [
    + { … }
  ]
}
```

Device

Use Knox Guard to control specific devices that are registered through the console or using the Knox Guard API. These REST APIs can be used to remind the consumer as well as lock a device until the completion of device management has been made.

Approve a Device

Accept the device in 'Pending' state.

HEADER PARAMETERS

x-knox-apitoken required	string The authentication token used to verify the REST API and specify the account. Generate a JWT using the Knox Cloud APIs token request . > *This parameter will need to be generated every 30 minutes.*
x-knox-transactionId	string This is an optional user generated unique alpha numeric code. Please create a different code for each request which can be used for debugging. /kcs/v1.1/kg/transaction.

REQUEST BODY SCHEMA: application/json

Use /kcs/v1.1/kg/devices/list to retrieve the deviceUid ("imei" or "serial") and approveId ("approveId"). approveId is an optional user entered description when the device is first registered. For more information, please refer to Register devices .

deviceUid required	string (deviceUid) IMEI or serial number used to uniquely identify a device.
approveId	string Unique ID that you specify when you approve a device for the KG system. (Limit 50)
approveComment	string Comment that you specify when you approve a device for the KG system. (Limit 1000)

Responses

> 200

OK

— 201

Created

> 400

> 401

Unauthorized

— 403

Forbidden

— 404

Not Found

POST /kcs/v1.1/kg/devices/approve

Request samples

Payload | Curl

Content type

application/json

Copy

```
{
    "deviceUid": "453700000000106",
    "approveId": "string",
    "approveComment": "string"
}
```

Response samples

200 | 400 | 401

Content type

application/json

Copy

```
{
    "result": "string",
    "objectId": "string",
    "requestedId": "string"
}
```


Set blinking reminders

Set the blinking reminder to send the repeating notifications so end customer make a overdue payment.

HEADER PARAMETERS

x-knox-apitoken required	string The authentication token used to verify the REST API and specify the account. Generate a JWT using the Knox Cloud APIs token request . *This parameter will need to be generated every 30 minutes.*
x-knox-transactionId	string This is an optional user generated unique alpha numeric code. Please create a different code for each request which can be used for debugging. /kcs/v1.1/kg/transaction.

REQUEST BODY SCHEMA: application/json

Use /kcs/v1.1/kg/devices/list to retrieve the objectId ("id"), deviceUid ("imei" or "serial"), and approveId ("approveId"). approveId is an optional user entered description when the device is first registered. For more information, please refer to Register devices . The reminder period interval must be between 3 and 86,400 seconds (24 hours). There is a 200 character maximum for the message parameter.

objectId	string (deviceIdDescriptive) Unique ID generated by the KG system when a device is registered. You must specify either an objectId, deviceUid, or approveId; if you specify

	multiple values, the precedence is objectId, deviceUid, then approveId.
`deviceUid`	string (deviceUidDescriptive) IMEI or serial number used to uniquely identify a device. You must specify either an objectId, deviceUid, or approveId; if you specify multiple values, the precedence is objectId, deviceUid, then approveId.
`approveId`	string (approveIdDescriptive) Unique ID that you specify when you approve a device for the KG system. You must specify either an objectId, deviceUid, or approveId; if you specify multiple values, the precedence is objectId, deviceUid, then approveId.
`email`	string Email address displayed on the device for the end user to contact about a delinquent payment. You must provide either a phone number or an email address.
`tel`	string Phone number displayed on the device for the end user to contact about a delinquent payment. You must provide either a phone number or an email address.
`interval` required	integer <int64> [3 .. 86400] Time in seconds (max 24 hours) between notification reminders.
`message` required	string Message to display on the device. Limit 200 characters.
`timeLimitEnable`	boolean Used to disable the blinking reminder for a specific time. If this value is true, timeLimit field is required. Example: disable blinking reminder from 11PM - 7AM; timeLimitEnable:true, daysLimitEnable:false, timeLimit:[23,7], daysLimit:null
`daysLimitEnable`	boolean Used to disable the blinking reminder on specific days and hours. If this value is true, timeLimit and daysLimit fields are required. Example: disable blinking reminder from 6AM SUN- 7AM MON; timeLimitEnable:false,

	daysLimitEnable:true, timeLimit:[6,7], daysLimit:[0,1]
timeLimit	Array of integers <int32> [items <int32 >] Set this value to [23, 7] when you want to disable the blinking reminder from 11 PM to 7 AM. If 'timeLimitEnable' or 'daysLimitEnable' is 'true', this field is mandatory. Array size must be 2, the values of timeLimit must be between 0 and 23.
daysLimit	Array of integers <int32> [items <int32 >] Set this value to [0, 1] when you want to disable the blinking reminder from Sunday to Monday (0: Sunday, 1: Monday ... and 6:Saturday). If 'daysLimitEnable = true', this field is mandatory. Array size must be 2, the values of daysLimit must be between 0 and 6.

Responses

> 200

OK

— 201

Created

> 400

> 401

Unauthorized

— 403

Forbidden

— 404

Not Found

POST /kcs/v1.1/kg/devices/blink