

# Zebra OTA API



**ZEBRA**

## **Developer Guide** for Version 3.1 and Later

ZEBRA and the stylized Zebra head are trademarks of Zebra Technologies Corporation, registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners.  
© 2024 Zebra Technologies Corporation and/or its affiliates. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements.

For further information regarding legal and proprietary statements, please go to:

SOFTWARE: [zebra.com/linkoslegal](https://zebra.com/linkoslegal)

COPYRIGHTS: [zebra.com/copyright](https://zebra.com/copyright)

PATENTS: [ip.zebra.com](https://ip.zebra.com)

WARRANTY: [zebra.com/warranty](https://zebra.com/warranty)

END USER LICENSE AGREEMENT: [zebra.com/eula](https://zebra.com/eula)

## Terms of Use

### Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries (“Zebra Technologies”). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

### Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

### Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

### Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

## Revision History

Changes to the original guide are listed below:

Change	Date	Description
-01 Rev A	10/2021	Initial release.
-02 Rev A	12/2021	Updates for version 3.1
-03 Rev A	12/2021	Fix typos on page 40.
-04 Rev A	02/2022	Updates for version 3.3
-05 Rev A	08/2022	Updates for POST /v1/notification.
-06 Rev A	01/2023	Additional updates for existing features.
-07 Rev A	08/2023	Updated Deployment Request Description table and a note added. Added a note in Overview and "Retry Mechanism In Zebra Auto Update" for Auto Update Mode. Added A13 Behaviour on SDM660 Models. Updated description for expires tag in the Enrollment Detail Status Response Description table. Updated Restrictions and Behavior. Added Artifacts GET /v1/ota-api/artifacts/latest.
-08 Rev A	10/2023	Updated Interface Definitions. Added A13 Behaviour in EMMs.
-09 Rev A	07/2024	Added "model_serialNumber" filter to POST /v2/deployments/details. Added new API: POST /v3/deployments/cancel. Added "model_serialNumber", "fromDate", and "toDate" filters to POST /v1/devices.

# Contents

<b>About This Document</b> .....	<b>9</b>
Requirements .....	9
New in OTA API 3.1 .....	9
Notational Conventions .....	9
Icon Conventions .....	10
<b>Overview</b> .....	<b>11</b>
Architecture Flow .....	11
Zebra Update Access .....	11
Zebra Update Delivery .....	12
Application Flow .....	13
Enrollment Usage Flow .....	14
API Sequence Diagram .....	15
<b>Interface Definitions</b> .....	<b>16</b>
POST /v2/deployments .....	17
Deployment Request .....	18
Deployment Response .....	19
Deployment Request Description .....	20
Deployment Package Response Description .....	26
POST /v2/deployments/status .....	27
Input Simple Status Request .....	27
Input Simple Status Sample .....	27
Input Simple Status Response .....	28
Input Simple Status Request Description .....	28
Input Simple Status Response Description .....	29
POST /v2/deployments/details .....	31
Deployment Detail Status Request .....	31
Deployment Detail Status Response .....	32
Deployment Detail Status Request Description .....	33

## Contents

Deployment Detail Status Response Description .....	33
POST /v2/deployments/cancel.....	34
Cancellation Request .....	34
Cancellation Response.....	34
Cancellation Request Description .....	35
Cancellation Response Description.....	35
POST /v3/deployments/cancel.....	36
Cancellation Request .....	36
Cancellation Response.....	36
Cancellation Request Description .....	36
POST /v1/notification .....	37
Notification Request (1) .....	37
Notification Response (1) .....	38
Notification Request Description .....	39
Notification Response Description.....	40
Notification Request (2) .....	42
Notification Response (2) .....	43
POST /v1/devices .....	45
Device Detail Status Request.....	45
Device Detail Status Response .....	45
Device Detail Status Request Description .....	46
Device Detail Status Response Description .....	46
ENROLLMENT GET /v1/enrollment.....	48
Enrollment Detail Status Request.....	48
Enrollment Detail Status Response .....	48
Enrollment Detail Status Response Description .....	48
Asset GET /v1/asset .....	49
Asset Get Detail Status Request .....	49
Asset Get Detail Status Response .....	49
Asset Get Detail Status Response Description .....	49
Asset DELETE /v1/asset [De-Enrollment].....	50
Asset Detail Status Request .....	50
Asset Detail Status Response .....	50
Asset Detail Status Request Description .....	50
Asset Detail Status Response Description .....	51
Artifacts POST /v1/ota-api/artifacts/sort .....	52
Asset Detail Status Request .....	52
Asset Detail Status Response .....	53
Asset Detail Status Request Description .....	54
Asset Detail Status Response Description .....	54
Artifacts GET /v1/ota-api/artifacts/latest.....	55

Asset Detail Status Request .....	55
Asset Detail Status Response .....	55
Asset Detail Status Request Description .....	58
Asset Detail Status Response Description .....	58
<b>OS Update Management .....</b>	<b>59</b>
OTA API Data fields .....	59
Release Number Field .....	59
Android OS Release Numbers .....	60
OTA Code and Tokens .....	60
<b>Using a Local Media Server .....</b>	<b>61</b>
Local Media Server Requirements .....	61
Obtaining the Necessary Files .....	61
<b>Response Codes .....</b>	<b>64</b>
<b>Auto Update Mode .....</b>	<b>65</b>
LifeGuard Enablement Auto Update .....	65
Zebra Managed Auto Update .....	65
Enterprise Managed Auto Update .....	66
Comparing Managed Auto Update Methods .....	66
Step 1: Configure Devices for Auto update .....	67
Auto Deployment Request Example .....	69
Auto Deployment Response Example .....	70
Auto Deployment Request Example .....	70
Step 3: Update Auto Deployment Configurations .....	71
Auto Deployment Cancel Response Example .....	72
Zebra Managed Auto Update .....	72
Enabling from Device Settings .....	72
Enabling from StageNow .....	73
Confirming Auto Update is Enabled .....	74
Restrictions and Behavior .....	75
Retry Mechanism in Zebra Auto Updates .....	75
Skip Auto Updates .....	76
Special Considerations .....	77
Deprecation of Simulate .....	77
Migrating to Notification API .....	78
Notification API .....	80

## Contents

Notification API Flow .....	80
Notification API Consumption .....	82
Pseudo Code for Auto Updates .....	82
Pseudo Code for Manual Deployments .....	84
<b>Appendix .....</b>	<b>85</b>
ENUMS .....	85
Reset Types .....	85
Network Types.....	85
Deployment Query Status .....	85
updateType.....	86
target .....	86
Deployment Status .....	86
Job Status.....	86
Detailed Job Status .....	87
Request Status .....	87
authType.....	87
Device Model .....	88
Android Desserts .....	89
installMode .....	89
notificationType .....	89
deployment_query status.....	89
Deployment Status Triggers.....	90
CREATED .....	90
INPROGRESS.....	90
COMPLETED .....	90
Job Status Triggers .....	90
CREATED .....	90
INPROGRESS.....	90
COMPLETED .....	91
CANCELLED.....	91
UNKNOWN.....	91
FAILED .....	91
Date Time Format .....	92
Examples.....	92
Clearing Enrollment Information.....	93
A13 Behaviour on SDM660 Models .....	94
A13 Behaviour in EMMs.....	94

# About This Document

This guide provides information about development for Zebra's Over the Air (OTA) API. The OTA API extends the Zebra OTA service to cover the last mile delivery of operating system (OS) updates to Zebra mobile computers.

## Requirements

Support of API development for Zebra's OTA API service is limited to:

- Authorized Enterprise Mobility Management (EMM) clients
- Properly enrolled devices
- Devices running Android™ 7 or higher

## New in OTA API 3.1

OTA API 3.1 introduces the following new features.

- Full automation for auto update managed by Zebra, allowing non-EMM users to keep their device up-to-date by opting in to Zebra auto update.
- Network and power options for deployments.
  - Download over cellular or restrict download to Wi-Fi.
  - Installation only while charging.
  - Installation only above a certain battery charge level.
- Customizable message which lets end-users know what to expect during deployment.
- Customizable postponement message for end-users.

## Notational Conventions

The following conventions are used in this document:

- Bullets (•) indicate:
  - Action items
  - Lists of alternatives
  - Lists of required steps that are not necessarily sequential.
- Sequential lists (e.g., those that describe step-by-step procedures) appear as numbered lists.



## Icon Conventions

The documentation set is designed to give the reader more visual clues. The following graphic icons are used throughout the documentation set. These icons and their associated meanings are described below.



**NOTE:** The text here indicates information that is supplemental for the user to know and that is not required to complete a task.



**IMPORTANT:** The text here indicates information that is important for the user to know.



**CAUTION:** If the precaution is not heeded, the user could receive minor or moderate injury.



**WARNING:** If danger is not avoided, the user CAN be seriously injured or killed.



**DANGER!** If danger is not avoided, the user WILL be seriously injured or killed.

# Overview

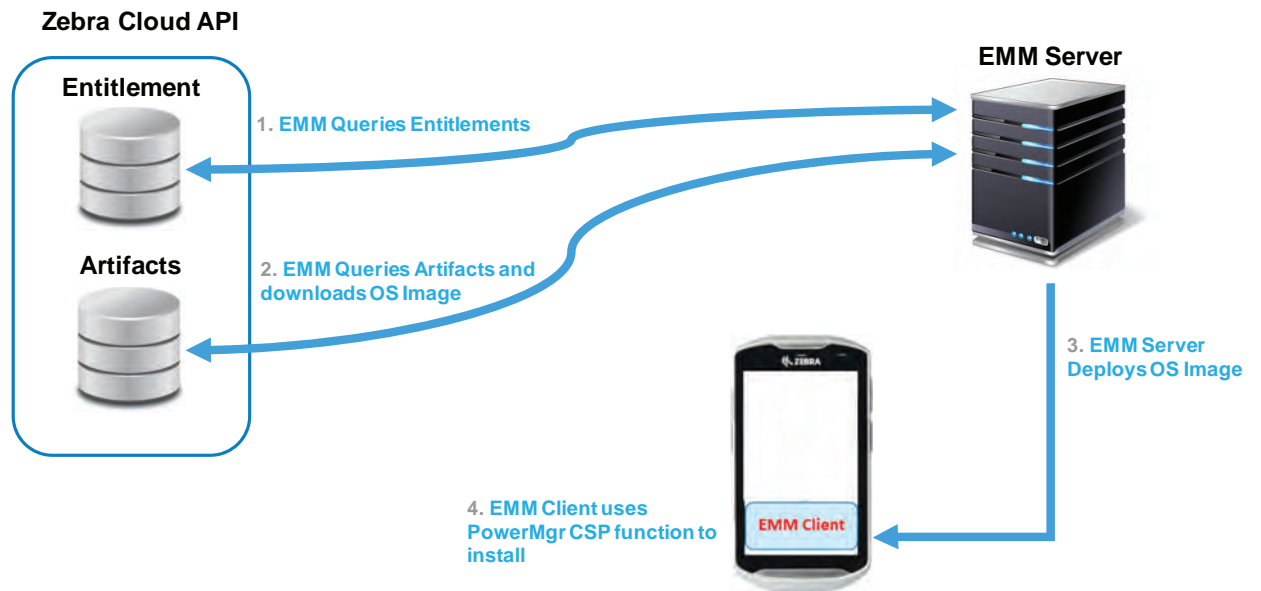
This section provides enrollment and deployment examples of the development process using Zebra's OTA API.

## Architecture Flow

The following sections provide architecture flow diagrams for Zebra Update Access and Zebra Update Delivery.

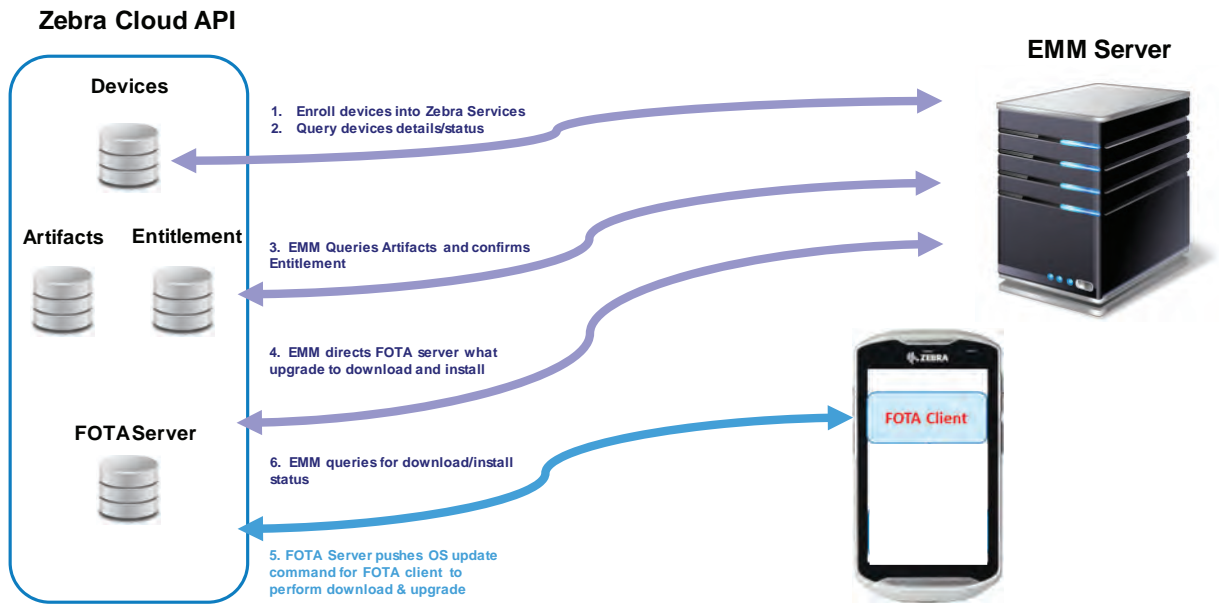
### Zebra Update Access

The EMM handles the last mile delivery of OS updates.



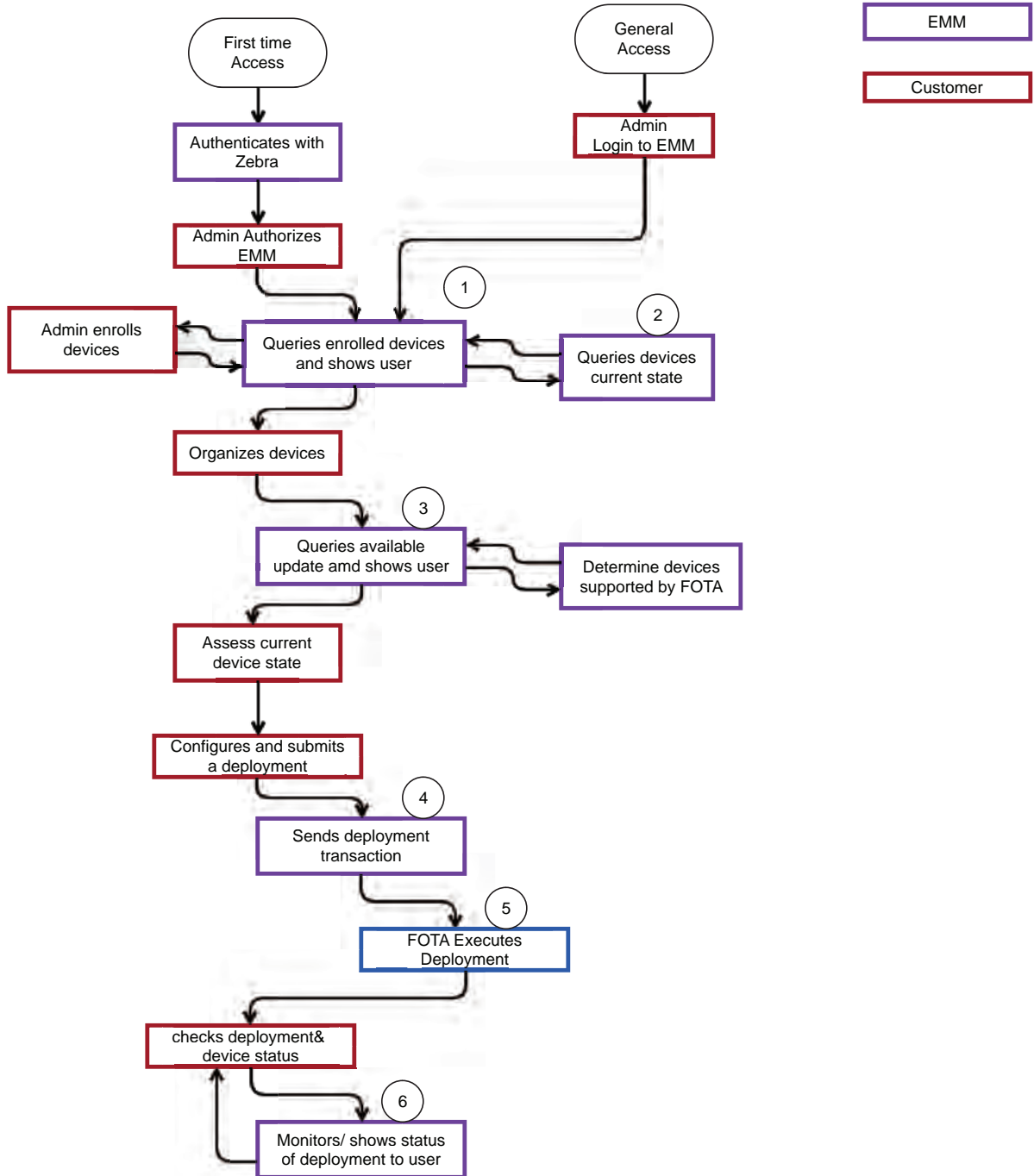
## Zebra Update Delivery

Zebra handles the last mile delivery of OS updates.



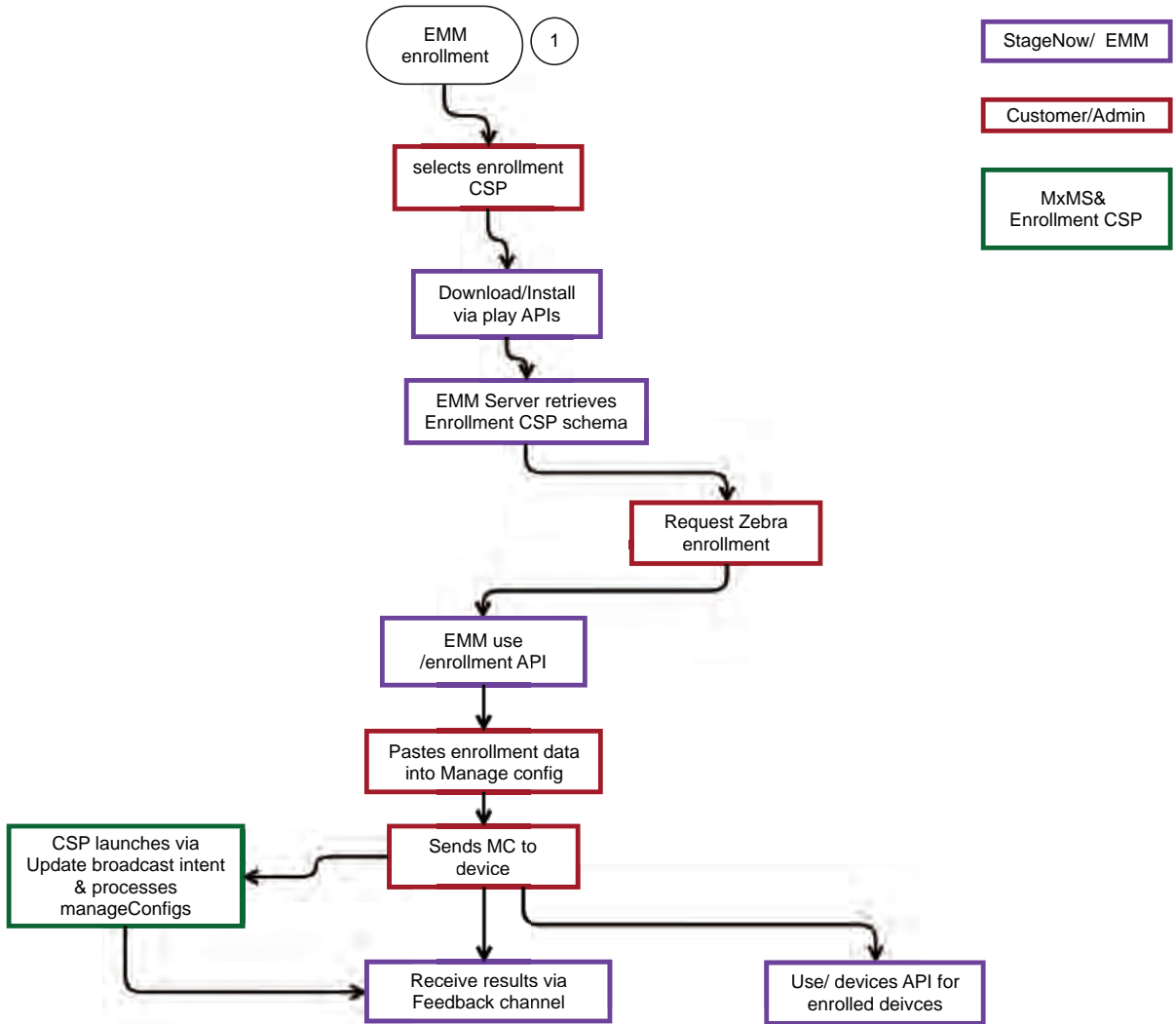
# Application Flow

This diagram outlines the end-to-end flow of using the Over the Air (OTA) API's, including both Update Access and Update Delivery.

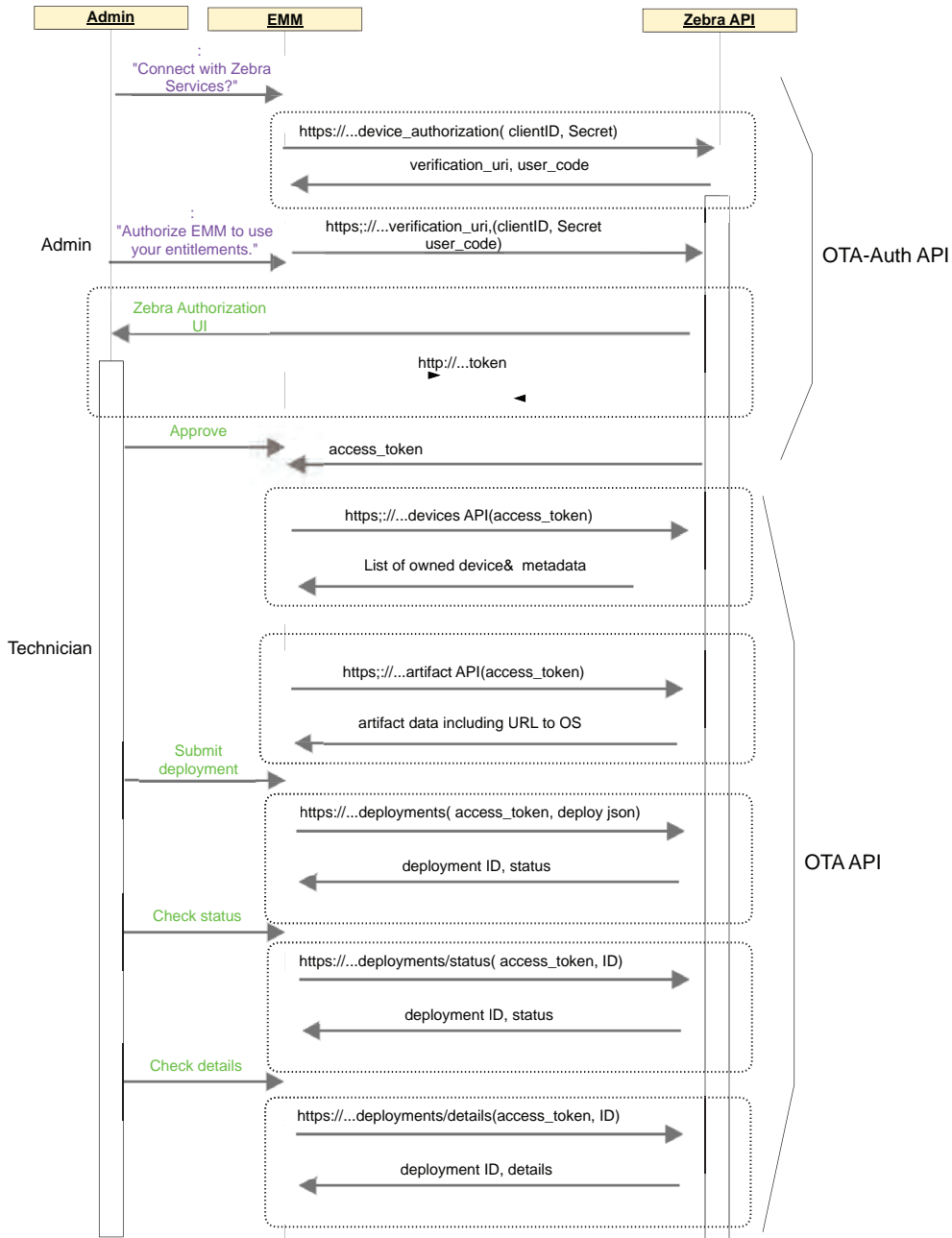


# Enrollment Usage Flow

This diagram outlines the end-to-end flow of device enrollment, a component of the Update Delivery API.



# API Sequence Diagram



# Interface Definitions

The OTA API set is a RESTful interface provided by Zebra for partners to build a Firmware Over the Air (FOTA) solution. The functions are targeted at simplifying the management and deployment of OS updates to Zebra devices.

Endpoint	HTTP Verb	Description	Inputs	Outputs	EMM Use case
<b>Deployment API</b>					
/v2/deployments	POST	Creates a deployment transaction by specifying options and devices.	Deployment Package	Application Response	Initiate a deployment.
/v2/deployments/status	POST	Discovers on-going non-device specific status of a deployment list.	InputSimpleStatus	Simple Status	Get status of a list of deployments (summary only).
/v2/deployments/detail	POST	Gets full details down to device level of a deployment.	InputDetailStatus	Detailed Status	Get the full status of a deployment.
/v2/deployments/cancel	POST	Stops an on-going deployment.	Cancellation	Application Response	Cancel a deployment.
<b>Notification API</b>					
/v1/notification	POST	Queries recently released artifacts	Notification filter	Notification Response	Check for new artifacts and notify end users of new updates when using auto deployments.
<b>Device API</b>					
/v1/devices	POST	Provides list of devices with relevant metadata.	Automatic based on user.	List of devices	Get a list of devices with relevant metadata. For example, device "FOTA ready".
<b>Enrollment API</b>					
/v1/enrollment	GET	Returns enrollment XML.	Automatic based on user.	Enrollment TOKEN	Get enrollment token to send to device.

## Interface Definitions

Endpoint	HTTP Verb	Description	Inputs	Outputs	EMM Use case
<b>Asset API</b>					
/v1/asset	GET	Returns a list of assets enrolled by that user	Automatic based on user.	List of assets.	Retrieve the list of assets enrolled by that user.
/v1/asset	DELETE	De-enroll the device.	Model and serial number	Application response	Remove the association of device and user. Call this API to de-enroll a device.
<b>Sort API</b>					
/v1/ota-api/artifacts/sort	POST	Returns a sorted list of OS updates.	Strings: deviceModel artifactBuildId	JSON Array Ordered list of applicable OS updates.	Simplifies organizing the update list for end users.
/v1/ota-api/artifacts/latest	GET	Returns latest applicable OS updates per model per sub type.	detailLevel	JSON array containing latest applicable OS update per model per sub type (GMS/AOSP).	Simplifies the need to view latest available update for a model sub type combination in the system.

## POST /v2/deployments

Creates a deployment.



## Deployment Request

```

{
  "simulate": false,
  "profile": {
    "updateType": "update type",
    "timeOffset": "+03:00",
    "target": {
      "type": "targetENUM",
      "value": {
        "artifactName": " target_artifact_name",
        "bsp": "target_bsp",
        "patch": "target_patch",
        "OSVersion": "target_OS_version"
      }
    }
  },
  "mediaServer": [
    [
      {
        "artifactUrl": "media_server_url_root",
        "authHeaderName": " Authorization",
        "authHeaderValue": "type credentials"
      }
    ]
  ]
},
"schedule": {
  "mode": "SCHEDULED",
  "duration": "deployment duration in days"
},
"rules": {
  "download": {
    "startDate": "2018-07-25T10:20:32"
    "network": "ANY",
    "autoUpdateDelay" : 10
  },
  "install": {
    "startDate": "2018-07-25T10:20:32",
    "timeWindowStart": "19:00:00",
    "timeWindowEnd": "23:00:00",
    "allowUserPostpone": true,
    "maxPostponeDuration":5,
    "postponeMessage": "Message",
    "userMessage": "Display this helpful text on device"
  },
  "battery": {
    "level": 30,
    "enforceOnCharger": true
  },
},
"devices": {
  "model": "TC75",
  "serial": [
    "3453453",
    "353453453"
  ]
}
}

```

## Deployment Response

```
{
  "data": {
    "deploymentId": "12345678764",
    "action": "deployments | simulated deployments",
    "message": "response message , contains success | Failure messages ",
    "code": "2XX | 4XX | 5XX",
    "timestamp": "2018- 07-25T00:00:00+03:00"
    "artifacts":[
      //only in case of simulated deployments
      {
        "artifactId":"name.zip",
        "artifactDownloadURL":" artifact download url",
        "optional":true,
        "packageSize":128888123
      },
      {
        "artifactId":"name.zip",
        "artifactDownloadURL":" artifact download url",
        "packageSize":128888123
        //by default optional:true
      }
    ]
  }
}
```

## Deployment Request Description

Tag	Required	Type	Boundaries (If Provided)	Description
simulate	No	Boolean	Default false if not specified	<p>Set this flag to true to create and run a simulated deployment.</p> <p>To use a local media server, run a simulated deployment. The simulated deployment returns the required OS files, which must be uploaded by the customer to their local media server. Using a local media server allows the device to download the OS files from a local server instead of through the internet (Zebra's server), saving broadband traffic.</p> <p>Note: When simulate is set to true, only the profile attributes are required.</p> <p>The simulate feature is deprecated in version 3.3 and later.</p>
profile	Yes	object		
profile.updateType	No	string/ENUM	Default: CUSTOM updateType ENUM	<p>CUSTOM - When Custom Mode is set the request must indicate one of two types:</p> <ul style="list-style-type: none"> <li>• Provide artifactName (AN - see profile.target.value)</li> <li>• Provide artifactValues (AV - see profile.target.value)</li> </ul> <p>LATEST - The latest released update becomes the target OS. If LATEST is specified, the target object is not required.</p> <p>Note: LATEST may update the device to a new Android version.</p> <p>AUTO: The device always looks for the latest package available in smart repo and tries to update whenever a new package is available. This continues until the admin cancels the auto update.</p> <p>While other modes return an ID starting with FOTA-x, AUTO mode returns an ID starting with AUTO-x.</p> <p>Only CUSTOM, LATEST and AUTO types are supported.</p> <p>Security Patch Level (SPL) is considered for arriving LATEST and AUTO update types.</p>

## Interface Definitions

Tag	Required	Type	Boundaries (If Provided)	Description
profile.timeOffset	No	String/Date		<p>This attribute indicates the deployment time offset. The time offset is the time timezone where the devices are located.</p> <p>The deployment start and end data uses this timezone.</p> <p>For example, "timeOffset": "+03:00",</p>
profile.target	Yes	object		<p>This input is ignored when profile.updateType is set to LATEST or AUTO.</p>
profile.target.type	Yes	string/ENUM	target ENUM	<p>Indicates what kind of value is being passed to indicate the destination artifact.</p> <p>Must be one of the following:</p> <ul style="list-style-type: none"> <li>AN - Specify artifact name.</li> <li>AV - Specify artifact value.</li> </ul>
profile.target.value	Yes	object	targetValue	<p>Depending on profile.upgradeDestination.type the input should be:</p> <p>type: AN, value:</p> <pre>{   "artifactName": "artifact name here" }</pre> <p>type: AV, value:</p> <pre>{   "patch": "patch_number_to_upgrade",   "bsp": "bsp_version_to_upgrade",   "OSVersion": "OS version to upgrade" }</pre> <p>Where OSVersion corresponds to one of the following number values:</p> <ul style="list-style-type: none"> <li>"7.1.2" - Android Nougat</li> <li>"8.1.0" - Android Oreo</li> <li>"10" - Android 10</li> <li>"11" - Android 11</li> </ul>
profile.media_server	No	array string		<p>Media server details.</p> <p>For information on using a local media server, see <a href="#">Using a Local Media Server</a>.</p> <p>It is not recommended to use this API when profile.updateType is set to AUTO. Instead, use the Notification API. See <a href="#">Notification API</a>.</p>
profile.media_server.authorizationHeaderName	No	string	Authorization	<p>Authorization header name for the authorization model used by the customer.</p> <p>Typically: "Authorization".</p> <p>Provide the header name if you are using a custom header name. For example, token etc.</p>

## Interface Definitions

Tag	Required	Type	Boundaries (If Provided)	Description
profile.media_server.authorizationHeaderValue	Yes	string	authTypeENUM	<p>Authorization type and credentials - For more information, go to: <a href="https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Authorization">developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Authorization</a></p> <p>The following types are supported:</p> <ul style="list-style-type: none"> <li>• Bearer</li> <li>• Basic</li> <li>• Custom - Only supported if authHeaderName is not "Authorization".</li> </ul>
profile.media_server.artifactUrl	Yes	string		Root folder of the media server URL where the downloads will be stored.
schedule	Yes	object		<p>This Window (startDate to endDate) indicates the time window when the upgrade process can begin (Upgrade = Download + Install).</p> <p>If the download begins before the scheduled endDate, a complete installation is attempted, even if the installation starts past the endDate.</p>
mode	No	string/ ENUM	installModeEnum	<p>Default is INSTALL_NOW.</p> <p>All scheduled deployments date and time are in the device's timezone.</p> <p>For Install Now, the date and time are in UTC (same date and time anywhere in the world).</p>
duration	No	string	Maximum 28 days	<p>Default: 28 days.</p> <p>Sequence of dates are:</p> <ol style="list-style-type: none"> <li>1. Download start date.</li> <li>2. Install start date.</li> <li>3. Schedule end date.</li> </ol> <p>If any of the values are not provided, the date provided in the preceding step of the sequence is used.</p> <p>If no values are provided, the string value of the current UTC is used.</p>
rules:download	No	object		Rules related to downloads. If the device is AB stream-able, download time is used to initiate the streaming and installation.
startDate	No	string	UTC Now Maximum 10 days	<p>Date and time in the device time zone when the download will start.</p> <p>If the download fails due to network conditions or because the file is unavailable in local media server, the download is automatically reattempted a few times every day until the deployment end duration.</p>

## Interface Definitions

Tag	Required	Type	Boundaries (If Provided)	Description
autoUpdateDelay	No	integer	1-10	Optional integer that can be set when profile.updateType is AUTO. Number of days after which the OS updates are pushed to a device. This delay begins when a new artifact is detected and notification is issued from the Notification API. Maximum: 10 days Default: 1 day.
network	No	string	network_type ENUM	See Network type ENUM for attributes Default: ANY For Wireless LAN devices, it is not recommended to provide network rules to avoid downloads that are larger than the available device storage. While a device has an Ethernet connection, this value is ignored.
rules:install	No	object		Rules related to Install.
startDate	No	date	Download start date	Date and time in device time zone when the install will start. Default - download startDate. Ignored when profile.updateType is AUTO.
timeWindowStart	No	time	00:00:00 - 23:30:00	Time of day when installation should begin. The time is expressed in a 24-hour format, as hh:mm, and is in the device time zone. Default - 00:00:00. Respected for all values of profile.updateType, including AUTO.
timeWindowEnd	No	time	00:30:00 - 23:59:59 Should be greater than timeWindowStart by 30 mins.	Time of day after which the install can not start. The time is expressed in a 24-hour format, as hh:mm, and is in the device time zone. Default - 23:59:59. Respected for all values of profile.updateType, including AUTO.
allowUserPostpone	No	boolean	true - false	Allows a user to postpone the installation on the device. This is helpful when a user is actively running a business app and cannot stop for an update. Default: false NOTE: When enforceOnCharge is set to true, it will override this setting and the user will not be given an option to postpone.

## Interface Definitions

Tag	Required	Type	Boundaries (If Provided)	Description
maxPostponeDuration	No	integer in hours	Minimum: 1 hour  Maximum: Duration of the deployment in hours.	Delay or snooze duration that is hard coded on the device for MVP. Users can delay an update for an unlimited number of times, starting from the first time the delay confirmation was presented to user, up to the maxPostponeDuration.  EMMs must make sure that the user is not able to enter a value that extends past the last installation window.
postponeMessage	No	string	0 - 128 char range Default: Empty value	Message in postpone popup. Default: "" If specified, a custom message is displayed to users when the postpone option is selected.
userMessage	No	string	0 - 128 char range Default: Empty value	Message sent to device that is visible on the FOTA Update UI. Default: ""
rules:battery	No	object		Battery rules are to be evaluated with an <b>AND</b> condition.  Battery rules are only applicable during the installation phase.
level	No	number (%)	maximum 100	Minimum battery level required for both download and installation. Default: -1 (LGE system defaults to the system default minimum levels.) Most Zebra devices require a minimum of 30% battery for a successful OS update.
enforceOnCharger	No	boolean	true - false	When set to <b>false</b> the client can install updates whether the device is in or out of the charger. Applied only for installation. Default: false
device	Yes	Object	Maximum 2500 devices	Target devices.
model	Yes	string	Model ENUM	The device model.  Only one model can be specified for each deployment.
serial	Yes	string array	array size >0, each serial number size 4 - 20	Serial number of the device.



**NOTE:** Summary of deployment behaviour:

1. Below A11 updates - FOTA downloads the OS package and subsequently installs the OS update.
2. A11 and above updates - FOTA streams all updates. Download and install happen at the same time.
3. Deployment API has 2 parameters:
  - Download time - defines when the download starts.
  - Install time - defines when the reboot will occur after installation is complete.
4. If POSTPONE is allowed (`allowUserPostpone` is true), then install time is ignored.
5. If download time is left out – downloading starts immediately.
6. If install time is left out and `allowUserPostpone` is false – the device will reboot immediately after installation.



## Deployment Package Response Description

Tag	Description
deploymentId	This is a system generated deployment id provided during creation of the deployment. Returned only if operation was a success (code == 200)
name	String that is a human readable name of the deployment. Not required, but useful for debugging.
action	Name of the action in response to which this response was generated.
message	Success or failure messages.
status	String, human readable response code. If one or more devices are already configured for auto mode, returns <b>ALREADY IN AUTO MODE</b> .
timestamp	Time at which this response was generated. Can be used to sequence status. Always returned in deployment timezone.
code	Request result. This means that the request has been accepted and will be worked on. This does not mean that the operation was successful. Possible values: request_status ENUM
artifacts	Only included in the response for action type "simulated deployments". Array of artifacts required as per input profile.
artifactId	Name of the artifact.
artifactDownloadURL	Download link for the artifact.
packageSize	Size of the artifact in bytes.
optional	Not currently used.

## POST /v2/deployments/status

Provides the status summary for given deployment.

### Input Simple Status Request

```
{
  "deploymentTag": " ",
  "deploymentId": [],
  "state": "Active/All",
  "fromTime": "2018-07-25T00:00:00+00:00",
  "toTime": "2018-07-28T00:00:00+00:00",
  "offset": 50,
  "limit": 100
}
```

### Input Simple Status Sample

```
Get simple status by deployment Id
{ "deploymentId": [ "12345" ]}
Get Simple status by deployment Tag
{ "deploymentTag": "Client Demo" }
Get all active deployments in past 24 hours
{
  "state": "ACTIVE",
  "fromTime": "2018- 07-25T00:00:00+00:00",
  "toTime": "2018-07- 26T00:00:00+00:00"
}
```

## Input Simple Status Response

```

{
  "head": {
    "page": 10,
    "offset": 50,
    "limit": 100,
    "total": 1000
  },
  "data":
  [{
    "autoUpdateId": "AUTO-1"
    "deploymentTag": "abc",
    "deploymentId": "342353",
    "deploymentStatus": "InProgress",
    "totalDevices": 1000,
    "created": 5,
    "scheduled": 5,
    "downloading": 20,
    "awaitingInstall": 10,
    "completed": 30,
    "cancelled": 5,
    "unknown": 5,
    "cancelRequested": "true",
    "failed": {
      "download": 20,
      "install": 10
    }
  }
  "completedOn": "2018-07-25T00:00:00+03: 00",
  "cancelledOn": "2018-07-25T00:00:00+03: 00",
  "associatedDeployments": ["FOTA-1", "FOTA-2", "FOTA-3"]
}]
}

```

## Input Simple Status Request Description

Tag	Required	Type	Boundaries	Description
deploymentTag	No	string	1 - 50 chars	DeploymentTag derives a list of deploymentIds that need to be returned. FTS server resolves the deploymentTags and obtains deployment Ids, then adds them to the deploymentId array. Default: ""
deploymentId	No	string array	5 - 50 chars	DeploymentId for which the status has to be returned. Default: ""
state	No	string	deployment_query status ENUM	Deployments in these states will be returned. Default: "ALL"
fromTime	No	date	Now minus 90 days	From Time for generation of the response. <b>Default:</b> Now minus 24 hours

## Interface Definitions

Tag	Required	Type	Boundaries	Description
toTime	No	date	> fromTime < Now	To Time for generation of the response. <b>Default:</b> Now
offset	No	int	>-1	Data offset, <b>Default:</b> 0
limit	No	int	5 - 250	Number of records to be returned, <b>Default:</b> 100

### Input Simple Status Response Description

Tag	Description
head	Meta Data about the response.
page	The page number of the response. This is usually the same as the offset or limit, unless the limit is changed in subsequent requests.
offset	An integer that indicated the offset. The <code>OFFSET</code> clause skips the <code>offset</code> rows before beginning to return the rows.
limit	An integer that indicates how many rows are returned.
total	An integer that indicates the total number of rows available in the database.
data	The response data as an array of objects, one for each deployment ID.
deploymentTag	This field is deprecated and can be ignored.
autoupdateId	A string used to uniquely identify the auto update deployment.
deploymentId	A string used to uniquely identify the transaction. Requests made for auto update deployments return the simple status of the latest <code>deploymentId</code> for that auto update.
deploymentStatus	Active indicates the transaction server is still attempting to complete its work.  Possible values: See Deployment <b>Status enum</b> .
totalDevices	An integer that indicates the total number of devices in the deployment.
created	An integer that indicates the total number of devices that have a job in the CREATED state. Typically indicates jobs that did not reach the devices.
scheduled	An integer that indicates the total number of devices that received the json and are scheduled.
downloading	An integer that indicates the total number of devices where the download is in progress.
awaitingInstall	An integer that indicates the total number of devices where the download is complete, but installation has not begun.
completed	An integer that indicates the total number of devices where installation was successful.
cancelled	An integer that indicates the total number of devices where deployment was cancelled successfully.
unknown	An integer that indicates the total number of devices where no deployment status or end state has not received, even after the scheduled end date was reached.
cancelRequested	A boolean that indicates if a cancellation was requested on the deployment. NOTE: A cancellation request does not guarantee that the deployment was cancelled.

## Interface Definitions

Tag	Description
failure:download	An integer that indicates the total number of devices that have failed to download the new OS file.
failue:install	An integer that indicates the total number of devices that have failed to install the new OS file.
completedOn	<p>The date and time when this deployment was completed. If the deployment is not completed before the deployment end date, then completedOn and endDate are the same. This is always in the deployment timezone.</p> <p>Note: An installation that is in progress can continue past the deployment end date.</p>
cancelledOn	The date and time when this deployment was cancelled. If the deployment is not cancelled, this attribute does not appear. This is always in the deployment timezone.
associatedDeployments	An array containing the deployments associated with the auto update for which the status is requested. Returns only for auto update status requests.

## POST /v2/deployments/details

Request detailed status.

### Deployment Detail Status Request

```
{
  "deploymentId": "",
  "appendStatus": true,
  "offset": 0,
  "limit": 100,
  "model_serialNumber": ["TC27_12345", "TC72_12345"]
}
```

## Deployment Detail Status Response

```

{
  "head": {
    "page": 10,
    "offset": 50,
    "limit": 100,
    "total": 1000
  },
  "data": {
    {
      "deploymentTag": "abc",
      "deploymentId": "342353",
      "deploymentStatus": "InProgress",
      "totalDevices": 1000,
      "created": 5,
      "scheduled": 5,
      "downloading": 20,
      "awaitingInstall": 10,
      "completed": 30,
      "cancelled": 5,
      "unknown": 5,
      "cancelRequested": "true",
      "completedOn": "2018-07-25T00:00:00+03:00",
      "failed": {
        "download": 20,
        "install": 10
      }
    }
  },
  "cancelledOn": "2018-07-25T00:00:00+03:00",
  "devices": [
    {
      "model": "TC75",
      "serialNumber": "3424645645634",
      "status": "job_status",
      "detailedStatus": "job_detailed_status",
      "message": {
        "code": null,
        "args": null
      },
      "downloadCompletionTime": "2018-07-25T00:00:00+03:00",
      "installationCompletionTime": "2018-07-25T00:00:00+03:00",
      "associatedDeployments": ["FOTA-1", "FOTA-2", "FOTA-3"]
    }
  ]
}

```

## Deployment Detail Status Request Description

Tag	Required	Type	Boundaries	Description
deploymentId	Yes	string	5 - 50 chars	A deployment Id for which the status has to be returned.
appendStatus	No	boolean	true-false	If true, appends simple status and the details.
offset	No	int	> -1	Data offset. Default: 0
limit	No	int	5 - 250	Number of records to be returned. Default: 100
model_serialNumber	No	string array	1 - 250	Number of records to be returned based on the filter values.

## Deployment Detail Status Response Description

Tag	Description
head	Metadata about the response.
page	The page number of the response. This is usually same as the offset or limit, unless the limit is changed in subsequent requests.
offset	The <code>OFFSET</code> clause skips the <code>offset</code> rows before beginning to return the rows.
limit	How many rows to return.
total	Total number of rows available in the database.
data	The response data as an array of objects, one each for deployment ID.
simple_status attributes	See <a href="#">Input Simple Status Response Description on page 29</a> for more information on simple status attributes.
devices	Array of devices with their detailed status.
model	Device model.
serialNumber	Device serial number.
status	Job status. This is for each device. Possible values: Status job_status ENUM
detailedStatus	A more precise job status. Possible values: job_detailed_status ENUM



Tag	Description
message	<p>A message describing the deployment progress or reason for failures.</p> <p>This is an object that contains two attributes:</p> <ul style="list-style-type: none"> <li>code - An integer code which indicates the pre-canned message.</li> <li>args - Arguments to this canned message, if any.</li> </ul> <p>Please refer to appendix.</p> <p>If the status is DOWNLOAD_INPROGRESS, then the message object response code is 5087 with the following details in the args:</p> <ul style="list-style-type: none"> <li>download (25%, 50%, 75%)</li> <li>number of files (1/2 and 2/2)</li> </ul> <p>Download percentage and number of files are available only for Recovery and AB Seamless mode deployments.</p>
downloadCompletionTime	Date indicating the date and time when the download was completed. This is always in the device timezone.
installationCompletionTime	Date indicating the date and time when the installation was completed. This is always in the device timezone.
associatedDeployments	An array containing the deployments associated with the auto update for which the status is requested. Returns only for auto update status requests.

## POST /v2/deployments/cancel

### Cancellation Request

```
{
  "deploymentId": "12345678764"
}
```

### Cancellation Response

```
{
  "data": {
    "deploymentId": "12345678764",
    "action": "cancel",
    "message": "response message, contains success | Failure messages ",
    "code": "2XX | 4XX | 5XX",
    "timestamp": "2018-07-25T00:00:00+00:00"
  }
}
```

## Cancellation Request Description

Tag	Required	Type	Boundaries	Description
deploymentId	Yes	string	5 - 50 chars	A deploymentId that needs to be cancelled. If auto update is cancelled, the active deployment under the auto update is also cancelled.

## Cancellation Response Description

See [Deployment Response](#) on page 19.

## POST /v3/deployments/cancel

### Cancellation Request

```
{
  "deploymentIds": ["FOTA-1", "FOTA-2"]
}
{
  "deploymentIds": ["AUTO-1", "AUTO-2"]
}
```

### Cancellation Response

```
{
  "apiVersion": "Server.Version",
  "dataList":
  [
    {
      "deploymentId": "FOTA-1",
      "action": "CANCELLED",
      "message": "human readable response, contains success | Failure messages",
      "code": "2XX | 4XX | 5XX",
      "timestamp": "2018-07-25T00:00:00+00:00"
    },
    {
      "deploymentId": "FOTA-2",
      "action": "CANCELLED",
      "message": "human readable response, contains success | Failure messages",
      "code": "2XX | 4XX | 5XX",
      "timestamp": "2018-07-25T00:00:00+00:00"
    }
  ]
}
```

### Cancellation Request Description

Tag	Required	Type	Boundaries	Description
deploymentId	Yes	string array	Default 1 FOTA-ID/ AUTO-ID  Max 10 FOTA-IDs/ AUTO-IDs	A list of FOTA or AUTO IDs that are to be cancelled.  Each FOTA-ID/AUTO-ID can have max 2500 jobs. Hence, 10*2500 = 25000 jobs can be cancelled per call.

## POST /v1/notification

### Notification Request (1)

```
{
  "notificationType": "NEW_ARTIFACT_AUTO_UPDATE",
  "offset": 0,
  "limit": 100,
  "params": {
    "from_time": "2020-11-15T00:00:00+00:00", "to_time": "2020-11-17T00:00:00+00:00"
  }
}
```

## Notification Response (1)

```
{
  "head": {
    "page": 10,
    "offset": 50,
    "limit": 100,
    "total": 1000
  },
  "notifications": [
    {
      "type": "NEW_ARTIFACT_AUTO_UPDATE",
      "value": {
        "model": "TC57",
        "artifactName": "HE_FULL_UPDATE_10-12-13.00-QG-U00-STD-HEL-04.zip",
        "artifact_url": "",
        "notification_timestamp": "",
        "releaseNotes": "release notes url"
      }
    },
    {
      "type": "NEW_ARTIFACT_AUTO_UPDATE",
      "value": {
        "model": "TC77",
        "artifactName": "HE_FULL_UPDATE_10-12-13.00-QG-U00-STD-HEL-04.zip",
        "artifact_url": "",
        "notification_timestamp": "",
        "releaseNotes": "release notes url"
      }
    }
  ],
  "notificationsError": [
    {
      "type": "NEW_ARTIFACT_AUTO_UPDATE",
      "value": [
        {
          "code": "4013",
          "value": [],
          "detail": "3400 TC52 devices are non-entitled ";
        }
      ]
    }
  ]
}
```

## Notification Request Description

Tag	Required	Type	Boundaries	Description
notificationType	No	string	notificationType ENUM array	<p>There are two typed of notifications are supported</p> <p>NEW_ARTIFACT_AUTO_UPDATE. NEW_DEPLOYMENT</p> <p>NEW_ARTIFACT_AUTO_UPDATE This means only the new artifact notification related only those devices models which are in autoupdate (perpetual) mode will be sent in the response. Default: NEW_ARTIFACT_AUTO_UPDATE if notificationType key is not provided. if notificationType key is provided with empty or null value then an error is returned. NEW_DEPLOYMENT Returns the list of Artifacts needed to upgrade given devices to provided Target artifact</p>
if notificationType == <b>NEW_ARTIFACT_AUTO_UPDATE</b> .				
deploymentType	No	string	notificationType ENUM array	<p>Type of notification. The only supported type is NEW_ARTIFACT_AUTO_UPDATE.</p> <p>Returns a list of devices and updates targeted for auto update deployments.</p>
fromTime	No	timestamp	maximum 3 months in past from now (now - 3 months)	<p>From Time for generation of the response. Default: Current timestamp minus 24 hours.</p>
toTime	No	timestamp	maximum 3 months in past from now (now - 3 months)	<p>By default current To Time for generation of the response. Default: Current timestamp</p>
offset	No	int	>-1	<p>Data offset if the response is paginated. Default: 0</p>
limit	No	int	5 - 250	<p>Number of records to be returned Default: 100</p>
if notificationType == <b>NEW_DEPLOYMENT</b>				
Model	Yes	String		Model of the device
serialNumbers	Yes	Array	Max 2500 devices	Array of serial numbers for the above model
targetBuild	Yes	String		Target artifact name

## Notification Response Description

Tag	Description
head	Meta Data about the response.
page	The page number of the response. This is usually the same as the offset or limit, unless the limit is changed in subsequent requests.
offset	An integer that indicated the offset. The <code>OFFSET</code> clause skips the <code>offset</code> rows before beginning to return the rows.
limit	An integer that indicates how many rows are returned.
total	An integer that indicates the total number of rows available in the database.
notifications	Contains the response data as an array of objects with one object for each notification.
type	The type of notification.
value	Contains the notification details.
model	The device model.
artifactName	Name of the artifact.
artifact_url	If entitled, contains the URL for the artifact. If not entitled, this is empty.
notificationTimestamp	An ISO8601 formatted date indicating the time of the notification event.
releaseNotes	Link to the release notes
isOptional	Indicates if an update is optional or mandatory In most cases a full update is marked as optional if a diff package is available and supplied as part of response Consumer may choose to download diff package or Full update This can be helpful in cases where there are multiple diff packages (in case devices are running different OS and want to upgraded to common latest update) and admin may decide to download only one Full update package instead which is applicable to all the devices.
Metadata	
availableFrom	In certain cases the artifact may not be immediately available , in such cases this attribute will be present This attribute will tell at what time will be artifact be ready for download
Type	This will tell the artifact type FULL - Full update - typical size 1. to 2 GB DIFF - True delta package - typical size 1 MB to 500 MB PATCH - Cumulative patch - applicable to A10 and below - typical size 400 to 600 MB
notificationsError	Any error in input or unavailability of artifacts will appear here NOTE: If the supplied devices are non entitled, those devices will appear in notification error section.
Type	type of notification

## Interface Definitions

Code	Error code, please refer LGE error code in Troubleshooting guide
Value	List of serial numbers for which artifact could not be created
details	Details of the error For non-entitled device, only the count of device per model will be returned



## Notification Request (2)

```
{
  "notificationType": "NEW_DEPLOYMENT",
  "offset": 0,
  "limit": 100,
  "params": {
    "model": "TC57",
    "serialNumbers": [100001224, 100003254],
    "targetBuild": "HE_FULL_UPDATE_11-16-05.00-RG-U00-STD-HEL-04" ,
    "subType": "GMS" // By default subtype will be "GMS"
  }
}
```

## Notification Response (2)

```

{
  "head": {
    "page": 10,
    "offset": 50,
    "limit": 100,
    "total": 1000
  },
  "notifications": [{
    "type": "NEW_DEPLOYMENT",
    "value": {
      "model": "TC57",
      "artifactName": "HE_FULL_UPDATE_10-12-13.00-QG-U00-STD-HEL-04.zip",
      "artifact_url": "APIGEE with q param or download url",
      "notification_timestamp": "",
      "releaseNotes": "release notes url",
      "isOptional": true/false;
      "metadata": {
        "availableFrom": "Timestamp",
        "type": "FULL/DIFF/PATCH"
      }
    }
  },
  {
    "type": "NEW_DEPLOYMENT",
    "value": {
      "model": "TC57",
      "artifactName": "HE_FULL_UPDATE_10-12-13.00-QG-U00-STD-HEL-04.zip",
      "artifact_url": "APIGEE with q param or download url",
      "notification_timestamp": "",
      "releaseNotes": "release notes url",
      "isOptional": true/false;
    }
  }
],
  "notificationsError": [{
    "type": "NEW_DEPLOYMENT",
    "value": [
      {
        "code": "4000",
        "value": [
          100000000000123,
          10000000000146
        ],
        "message": "some message about error";
      }
    ],
  },

```

```
{
  "code": "4001",
  "value": [
    100000000000123,
    100000000000146
  ]
  "message": "some message about error";
},
{
  "code": "4013",
  "value": [
  ]
  "message": " 1 TC57 devices is non entitled ";
}
]
}
]
}}
```

## POST /v1/devices

### Device Detail Status Request

```
{
  "fotaReady": "1 | 0",
  "detailLevel": " 1 | 0",
  "offset": 0,
  "limit": 100,
  "model_serialNumber": ["TC27_12345", "TC72_12345"],
  "fromDate": "2024-05-24",
  "toDate": "2024-06-24"
}
```

### Device Detail Status Response

```
{
  "head": {
    "page": 10,
    "offset": 50,
    "limit": 100,
    "total": 1000
  },
  "data": {
    "devices": [
      {
        "model": "TC52",
        "serialNumber": "12345678901234",
        "fotaReady": 1,
        "lastStatusReportTime": "05-13-2019 10:10:10+00:00",
        "buildId": "02-13-15.00-0G-U40-STD",
        "patchVersion": "7",
        "OSversion": "8.1.0",
        "OSsubType": "GMS",
        "spl": "",
        "cpl": "date",
        "fingerprint": "devicefingerprint",
        "timeOffset": "00:30",
        "autoUpdateEnabled": false,
        "autoUpdateId": "AUTO-1"
      }
    ]
  }
}
```

## Device Detail Status Request Description

Tag	Required	Type	Boundaries	Description
fotaReady	No	string	1 or 0	1 returns only devices which are FOTA ready 0 returns all the devices, whether they are FOTA ready or not. Default: 0
detailLevel	No	string	1 or 0	Level 0 returns a list of the device model and serial numbers. Level 1 returns Level 0 data and additional device meta data. Default: 1
offset	No	int		The <code>OFFSET</code> clause skips the <code>offset</code> rows before beginning to return the rows. Default: 0
limit	No	int		How many rows to return. Default: 100
model_serialNumber	No	string array	1 - 250	Number of devices to be returned based on the device values passed.
fromDate & toDate	No	string	(toDate - fromDate) <= 3 months	Date range filters to fetch devices based on the enrollment time. Format: yyyy-mm-dd If either or both of the fields have null values, then the filter will be ignored.

## Device Detail Status Response Description

Tag	Description
head	Meta data about the response
page	The page number of the response. This is usually same as the offset or limit, unless the limit is changed in subsequent requests.
offset	The <code>OFFSET</code> clause skips the <code>offset</code> rows before beginning to return the rows.
limit	How many rows to return
total	The total number of rows available in the database.
data	The response data is an array of objects, one each for deployment ID.
model	Device model
serialNumber	Device serial number

## Interface Definitions

Tag	Description
fotaReady	<p>A response value of 0 indicates that the device is not FOTA ready.            A response value of 1 indicates that device is FOTA ready.            A device fotaReady status will show false for the following scenarios:</p> <ul style="list-style-type: none"> <li>• System update slider disabled from the device.</li> <li>• If the device is on a non-standard build.</li> <li>• If the device fails to upload the device information back to the server post enrollment.</li> <li>• If dependent applications - CTL/EnrollmentMgr/ZDS are not installed</li> </ul>
lastStatusReportTime	Last time any of the attributes were updated in UTC.
buildId	Device Build ID
patchVersion	Device patch
OSversion	Android version
OSsubtype	GMS   Non-GMS (restricted mode is not indicated)
spl	SPL
cpl	CPL
fingerprint	Device fingerprint
timeOffset	Timezone offset of the device
autoUpdateEnabled	If auto update is enabled the response is <b>true</b> . Enable auto update for a device by calling the deployment API with updateType as <b>AUTO</b> .
autoUpdateId	If the device is set to auto update mode, returns the auto update ID.

## ENROLLMENT GET /v1/enrollment

### Enrollment Detail Status Request

No input is required.

### Enrollment Detail Status Response

```
{
  "data": {
    "token_type": "TOKEN",
    "payload" : "data",
    "expires" : 1561363118,
    "scope": "enrollment",
    "message": "response message, contains success | Failure messages ",
    "code": "Empty|1001|1003"
  }
}
```

### Enrollment Detail Status Response Description

Tag	Description
httpstatuscode	200 - Success, 4XX, 500 - Internal Server Error (with server error codes mentioned within Data) This will be part of http response.
token_type	Type of response. Possible values are: <ul style="list-style-type: none"> <li>MXXML</li> <li>TOKEN - For MVP only TOKEN will be supported.</li> </ul>
payload	This will change depending on the <b>head.type</b> .
expires	EPOC time. Indicates for how long this payload will be valid. The default expiry is 10 years from the date of generation.
scope	Describes what you can do in this enrollment.
code	Possible values: <p>If the httpstatuscode is 500:</p> <ul style="list-style-type: none"> <li>1001 - Enrollment token generation failure</li> <li>1003 - Invalid user, if the httpstatuscode is 500</li> </ul> <p>If the httpstatuscode is 200:</p> <ul style="list-style-type: none"> <li>Empty</li> </ul>

## Asset GET /v1/asset

### Asset Get Detail Status Request

No input is required.

### Asset Get Detail Status Response

```
{
  "data": {
    "message": "response message, contains messages ",
    "code": "1102"
    "assets": [{
      "model": "TC55",
      "serialNumber": "17121930292"
    }]
  }
}
```

### Asset Get Detail Status Response Description

Tag	Description
httpstatuscode	Possible values: <ul style="list-style-type: none"> <li>200 - Success</li> <li>4XX or 500 - Internal server error, with server error codes included in the data. This is part of httpresponse.</li> </ul>
code	Possible values: <p>If the httpstatuscode is 500:</p> <ul style="list-style-type: none"> <li>1002 - Enrolled assets fetch failure</li> </ul> <p>If the httpstatuscode is 200:</p> <ul style="list-style-type: none"> <li>Empty</li> </ul>
model	Device model
serialNumber	Device serial number
lastEnrollmentTime	Timestamp of when the device was last enrolled.



## Asset DELETE /v1/asset [De-Enrollment]

Removes the association of a device from a user.

Call this API to de-enroll a device. De-enrollment takes 15 minutes to complete from the time a success response is received from the DELETE API. Devices continue to behave as an enrolled asset until the de-enrollment is complete. After de-enrollment, the device is removed from the tenant.

It is recommended to also clear the enrollment information from the device by clearing the application cache of the FOTA client and Enrollment Manager CSP. See [Clearing Enrollment Information on page 93](#).

If the device cache is not cleared and a re-enrollment is attempted, then a new token must be generated and pushed to the device as described in the original enrollment steps. This is true even if enrollment is by the same customer that previously enrolled the device.

### Asset Detail Status Request

```
{
  "data": [{
    "model": "TC55"
    "serialNumber": "17121930292"
  }]
}
```

### Asset Detail Status Response

```
{
  "data": {
    "action": "delete",
    "message": "response message , contains messages ",
    "timestamp": "2018- 07-25T00:00:00+00:00",
    "failedAssets": [
      {
        "model": "TC55",
        "serialNumber": "17121930292",
        "code": "1101 | 1104",
        "message": "Failure response message"
      }
    ]
  }
}
```

### Asset Detail Status Request Description

Tag	Required	Type	Boundaries	Description
model	Yes	String	NA	Device model
serialNumber	Yes	String	NA	Device serial number

## Asset Detail Status Response Description

Tag	Description
httpstatuscode	Possible values: <ul style="list-style-type: none"> <li>• 200 - Success</li> <li>• 4XX, 500 - Internal server error with server error codes mentioned in the data. This is part of httpresponse.</li> </ul>
action	Name of the action in response to which this response was generated.
status	Human readable response code as a string.
failedAssets	An array containing assets that failed to de-enroll.
timestamp	The time when this response was generated. This can be used to sequence the received status. This value is always returned in UTC.
code	Possible values: <p>If the httpstatuscode is 500:</p> <ul style="list-style-type: none"> <li>• 1104 - Asset de-enrollment failed.</li> </ul> <p>If the httpstatuscode is 200, the below values are sent to indicate to the user knowclients that the asset is de-enrolled:</p> <ul style="list-style-type: none"> <li>• 1101 - Asset is already dis-enrolled.</li> </ul>

## Artifacts POST /v1/ota-api/artifacts/sort

Takes the current state of a given device and returns a list of possible OS updates.

SPL is used as the primary sorting key. If an artifact with the same or higher SPL is available in the next OS, all newer SPL in current OS are not returned.

When using this API, keep in mind the following:

- This api is part of the OTA Update Access.
- Only use this API with Android N Patch 19 and above.
- The response format is different from other OTA API. Give special attention to the description.

## Asset Detail Status Request

```
{
  "deviceModel": "TC75x",
  "artifactBuildId": "01-01-49.00-NG-U18-STD"
}
```

## Asset Detail Status Response

```


{
  "respCode":{
    "code":"5200",
    "message": "ALL_ GOOD",
    "apiver": "3.2021.06.11.1"
  },
  "respData":[
    {
      "apiVersion": "1.24",
      "artifactActive": 0,
      "artifactBaseline": "01-01-48-NG-00-A",
      "artifactBspVersion": "01.01.49.00",
      "artifactBuildId": "01-01-49.00-NG-U18-STD",
      "artifactChecksum": "NA",
      "artifactDefaultLanguage": "en-US",
      "artifactDescription": "NA",
      "artifactDownloadCount": 2343,
      "artifactDownloadUrl": "NA",
      "artifactFingerprint": "Zebra/TC75x/TC75xDF:7.1.2/01-01-49.00-NG-U18-STD/140:user/release-keys",
      "artifactId": "1583",
      "artifactMimeType": "application/zip",
      "artifactOperatingSystem": "Android",
      "artifactOsVariant": "Nougat",
      "artifactPublishedDate": "2019-03-06T00:00:00Z",
      "artifactReleaseNotes": "NA",
      "artifactSize": 304358300,
      "artifactSubType": "GMS",
      "artifactType": "operating-system",
      "artifactUpdateType": "P",
      "carrier": "FUTURE",
      "criticalSecurityCount": 3,
      "criticalSecurityLevelDate": "NA",
      "cumulativeCveCount": 164,
      "deviceModel": "TC75x",
      "highSecurityCount": 19,
      "latestArtifactId": "CFE_ATLAS_01-01-49.00-NG-U18-STD.zip",
      "osVersionNumber": "7.1.2",
      "patchNum": "18",
      "patchNumSimplified": 18,
      "productPageUrl": "FUTURE",
      "protectedDownloadUrl": "NA",
      "releaseNumber": "012.000.018",
      "securityLevelDate": "2019-03-01T00:00:00Z",
      "solutionSupported": "[]",
      "targetBaseline": "01-01-48-NG-00-A",
      "targetBspVersion": "01.01.49.00",
      "updatedOn": "2020-10-07T18:40:54Z"
    },
    {
      // Other Smart repo records with higher SPL , higher Build
    },
    {
      // Other Smart repo records with higher SPL , higher Build
    }
  ]
}

```

## Asset Detail Status Request Description

Tag	Required	Type	Boundaries	Description
deviceModel	Yes	String	NA	Device model.
artifactBuildId	Yes	String	NA	Build ID of current artifact.

## Asset Detail Status Response Description

Tag	Description
respCode	Meta data Object
code	Response code indicating any errors. <ul style="list-style-type: none"> <li>code: "5200" , "message": "ALL GOOD"</li> <li>code: "5404" , "message": "Source Record Not Found in Smart Repo"</li> <li>code: "5400" , "message": "corresponding schema errors"</li> </ul>
message	User friendly string
apiversion	Api version in following format: LGE 3.0 , Year , Month , Day , version number within the day
respData	JSON array containing a sorted list of applicable updates. If a newer BSP has a lower SPL, the newer BSP is not presented.   <b>NOTE:</b> The first record is always the data corresponding to the input device and has an additional attribute with following value: <b>"sourceRecord": "matchobj"</b> Recommendations follow from second record onwards.

## Artifacts GET /v1/ota-api/artifacts/latest

Accepts an optional request parameter detailLevel when set to 0, and returns list of latest artifacts per model, and per subType. When set to 1 returns detailed information of latest artifacts per model, and per subType.

### Asset Detail Status Request

1. /v1/ota-api/artifacts/latest?detailLevel=0
2. /v1/ota-api/artifacts/latest?detailLevel=1

### Asset Detail Status Response

1. detailLevel=0

```
{
  "respCode": {
    "code": "5200",
    "message": "ALL GOOD",
    "apiver": "4.2022.02.07.12",
    "operation": "/latest"
  },
  "respData": {
    "GMS": {
      "TC72": "11-35-05.00-RG-U00-STD-HEL-04"
    },
    "AOSP": {
      "TC72": "11-35-05.00-RN-U00-STD-HEL-04"
    }
  },
  "respMetaData": {}
}
```

## 2. detailLevel=1

```

{
  "respCode": {
    "code": "5200",
    "message": "ALL GOOD",
    "apiver": "4.2022.02.07.12",
    "operation": "/latest"},
  "respData": {
    "GMS": {
      "TC72": {
        "apiVersion": "2.00",
        "artifactActive": 1,
        "artifactBaseline": "11-35-05.00-RG-U00",
        "artifactBspVersion": "11.35.05.00",
        "artifactBuildId": "11-35-05.00-RG-U00-STD-HEL-04",
        "artifactChecksum": "40948539ff6ead574943690225db7438",
        "artifactDefaultLanguage": "en-US",
        "artifactDescription": "Full Package Update",
        "artifactDownloadCount": 1,
        "artifactDownloadUrl": "",
        "artifactFingerprint":
        "Zebra/TC72/TC72:11/11-35-05.00-RG-U00-STD-HEL-04/318:user/release-keys;Zebra/TC72/TC72_RU
        :11/11-35-05.00-RG-U00-STD-HEL-04/318:user/release-keys",
        "artifactId": "301712",
        "artifactMimeType": "application/zip",
        "artifactOperatingSystem": "Android",
        "artifactOsVariant": "A11",
        "artifactPublishedDate": "2023-05-26T00:00:00Z",
        "artifactReleaseNotes":
        "https://www.zebra.com/content/dam/zebra_new_ia/en-us/software/operating-system/helios/HE_
        FULL_UPDATE_11-35-05.00-RG-U00-STD.pdf",
        "artifactSize": 1736439348,
        "artifactSubType": "GMS",
        "artifactType": "operating-system",
        "artifactUpdateType": "F",
        "carrier": "FUTURE",
        "criticalSecurityCount": 0,
        "criticalSecurityLevelDate": "NA",
        "cumulativeCveCount": 497,
        "deviceModel": "TC72",
        "highSecurityCount": 0,
        "latestArtifactId": "HE_FULL_UPDATE_11-35-05.00-RG-U00-STD-HEL-04.zip",
        "osVersionNumber": "11",
        "patchNum": "U00",
        "patchNumSimplified": 0,
        "productPageUrl": "FUTURE",
        "protectedDownloadUrl": "NA",
        "releaseNumber": "016.007.000",
        "securityLevelDate": "2023-04-05T00:00:00Z",
        "solutionSupported": "[\"FOTA\"]",
        "targetBaseline": "NA",
        "targetBspVersion": "NA",
        "installType": "AB_STREAMING",
        "packageType": "DYNAMICALLY_PACKAGED",
        "createdOn": "2023-05-29T07:50:00Z",

```





## Asset Detail Status Request Description

Tag	Required	Type	Boundaries	Description
detailLevel	No	Integer	1 or 0	Indicates whether to return the response in compact or expanded form.

## Asset Detail Status Response Description

Tag	Description
respCode	Meta data Object
code	Response code indicating any errors. <ul style="list-style-type: none"> <li>code: "5200" , "message": "ALL GOOD"</li> <li>code: "5404" , "message": "Source Record Not Found in Smart Repo"</li> <li>code: "5400" , "message": "corresponding schema errors"</li> </ul>
message	User friendly string
apiversion	Api version in following format: LGE 3.0 , Year , Month , Day , version number within the day
respData	JSON array containing a sorted list of applicable updates per model and per subtype.

# OS Update Management

This section provides information for managing OS updates on Zebra devices.

## OTA API Data fields

The following tables provide details on the OTA API data fields.

### Release Number Field

The **Release Number** field is available in the OTA API to help applications organize the artifact records of the Zebra OS files.

The release number is composed of three-digit components in the format of: XXX.YYY.ZZZ

**Table 1** Release Number Schema

	Component	Description	Example
XXX	Android OS	See <a href="#">Table 2</a> for details. This number starts from 000.	012 for Nougat
YYY	BSP Number	Indicates the BSP number. Starts from 000 and resets every Android OS change.	000 for BSP 01.12.03 [The very first BSP across M and N for TC 51 GMS] 005 for BSP 21.04.01 [The 5th and the Last BSP on M for TC 51 GMS] 000 for BSP 01.01.49 [ The very first BSP for N on TC51 GMS]
ZZZ	Patch Number	Indicates the absolute patch number. Corresponds to the simplified patch number. The value resets for every BSP but may not start from 000. This number is always increasing but may have gaps in the sequence.	008 for BSP 01.01.49 Patch 08 [ TC51 GMS] 010 for BSP 21.04.01 Patch 10 [ TC51 GMS]

## Android OS Release Numbers

The following table lists the version and OS number for each Android OS release.

**Table 2** Android OS Release Numbers

Code Name	Version Number	OS Number
Petit Four	1.1	000
Cupcake	1.5	001
Donut	1.5	002
Eclair	2.0 - 2.1	003
Frovo	2.2 - 2.2.3	004
Gingerbread	2.3 - 2.3.7	005
Honeycomb	3.0 - 3.2.6	006
Ice Cream Sandwich	4.0 - 4.0.4	007
Jelly Bean	4.1 - 4.3.1	008
KitKat	4.4 - 4.4.4	009
Lollipop	5.0 - 5.1.1	010
Marshmallow	6.0 - 6.0.1	011
Nougat	7.0 - 7.1.2	012
Oreo	8.0 - 8.1	013
Android P	9	014
Android Q	10 (subject to change)	015
Android R	11	016

## OTA Code and Tokens

The following table describes the OTA codes and tokens.

**Table 3** OTA Code and Tokens

Code or Token	Expires	Description
Auth Code	10 mins	Generated by the redirect URL or the user device code OTA-Auth API.
User Code	10 mins	Generated by the user device code OTA-Auth API.
Access token	60 mins	Generated by the <code>/token</code> OTA-Auth API.
downloadURLtoken	Must be configured.	Generated using the <code>/downloadurltoken</code> OTA API.
Client Id: Secret	Perpetual once approved. Can be manually revoked as required.	Generated when an app is created on the authorization server.

# Using a Local Media Server

Zebra OTA API supports the use of a local media server, allowing devices to pull the OS files locally instead of over the Internet from Zebra's media server.

## Local Media Server Requirements

The following are required to use a local media server.

- The local media server must be able to serve files using the http or https protocol.
- Devices pulling the OS update must have direct access the local media server.
- Device administrators must host the required files on the local media server.

## Obtaining the Necessary Files

To host the files on a local media server, administrators must first obtain the required files. The EMM tool provides the list of required files for a deployment by using the simulate feature of the deployment API. The response contains the URLs for the download.

1. Use the simulate feature of the deployment API. For more information, see [POST /v2/deployments](#).

The API response for a simulate deployment provides the following details:

```
{
  "data": {
    "deploymentId": "12345678764",
    "action": "deployments | simulated deployments",
    "message": "Response Message, contains success | Failure messages ",
    "code": "2XX | 4XX | 5XX",
    "timestamp": "2018-07-25T00:00:00+03:00"
    "artifacts":[ //only in case of simulated deployments
      {
        "artifactId":"HE_INCREMENTAL_UPDATE_02-11-01.00-PG-U06-STD.zip ",
        "artifactDownloadURL":"artifact download_url",
        "packageSize":10000
      },
      {
        "artifactId":"HE_FULL_PACKAGE_UPDATE_02-11-01.00-PG-U00-STD.zip ",
        "artifactDownloadURL":"artifact download_url",
        "packageSize":128888123
      }
    ]
  }
}
```

2. Download all the files from the URLs mentioned in the `artifactDownloadURL` attribute. There may be more than one file.
3. Place the downloaded files into one folder on the local media server. For example, the two files in the response example were placed in a folder named `lge_update` on the local media server as:
  - [https://local\\_media\\_server.my\\_company.com/lge\\_update/HE\\_INCREMENTAL\\_UPDATE\\_02-11-01.00-PG-U06-STD.zip](https://local_media_server.my_company.com/lge_update/HE_INCREMENTAL_UPDATE_02-11-01.00-PG-U06-STD.zip)
  - [https://local\\_media\\_server.my\\_company.com/lge\\_update/HE\\_FULL\\_PACKAGE\\_UPDATE\\_02-11-01.00-PG-U00-STD.zip](https://local_media_server.my_company.com/lge_update/HE_FULL_PACKAGE_UPDATE_02-11-01.00-PG-U00-STD.zip)
4. Construct the media server object of the deployment request in the following format. For more information, see [POST /v2/deployments](#).

```
"mediaServer": [  
  [  
    {  
      "artifactUrl": " https://local_media_server.my_company.com/lge_update",  
      "authHeaderName": "Authorization",  
      "authHeaderValue": "type credentials"  
    }  
  ]  
]
```

Where `artifactUrl` is the path to the root folder containing the downloaded files. For example, `lge_update` from the sample.

5. If the local media server is secured with an authorization:
  - For `authHeaderName` enter the auth header. For example, `Authorization:`.
  - For `authHeaderValue` enter the auth value. For example, `Basic YWxhZGRpbjpvGVuc2VzYW11.`
6. Make sure to change the `simulate` attribute to `false` for actual deployment of the OS update.

# Response Codes

For a list of OTA API response codes with descriptions and suggested solutions, refer to the Zebra OTA API Integration and Troubleshooting Guide.

# Auto Update Mode

## Overview

Auto update mode rolls out deployments of recently released artifacts to all the devices with unique serial numbers, similar to updates in consumer devices. The devices must have the auto update mode enabled at the time the updates or artifacts are published. You can configure auto updates to be applied with required constraint overrides such as, download or installation time, through the initial auto deployment configuration request.

Devices with the auto update mode enabled are ensured to be on the latest available OS version that the device supports.



**NOTE:** Auto update mode supports all the GMS models and only Helix Non-GMS model.

## LifeGuard Enablement Auto Update

The LifeGuard Enablement (LG) OTA system enables devices keep updated with the latest OS. Updates are automatically installed on the devices when they become available from Zebra.

Zebra auto updates function similarly to consumer device updates, where no action is required by the device operator or device administrator.

There are two ways to use the Zebra auto update service. Each device can only be enrolled in one auto update method.

- Zebra managed auto-update
- Enterprise managed auto-update

## Zebra Managed Auto Update

Zebra managed auto updates is the more simple solution, however using this method removes the device from any enterprise managed service in which it was previously enrolled.

This option can be enabled in two ways:

- End-users can opt in to the service through Android device settings. Administrators can disable this setting.
- Administrators can opt in to the LG OTA service with OEMConfig using managed configuration settings.



## Enterprise Managed Auto Update

Enterprise managed auto updates are a more complex solution that is only available for customers using an EMM integrated with Zebra's OTA service. This method offers additional controls for setting schedules and other rules.

Using this method requires device administrators to:

- Enroll the device with Zebra's EnrollmentMgr.
- Set an auto update deployment policy for devices.

## Comparing Managed Auto Update Methods

Zebra and enterprise managed auto update methods offer different features and are enabled differently.

**Figure 1** Managed Auto Update Comparison

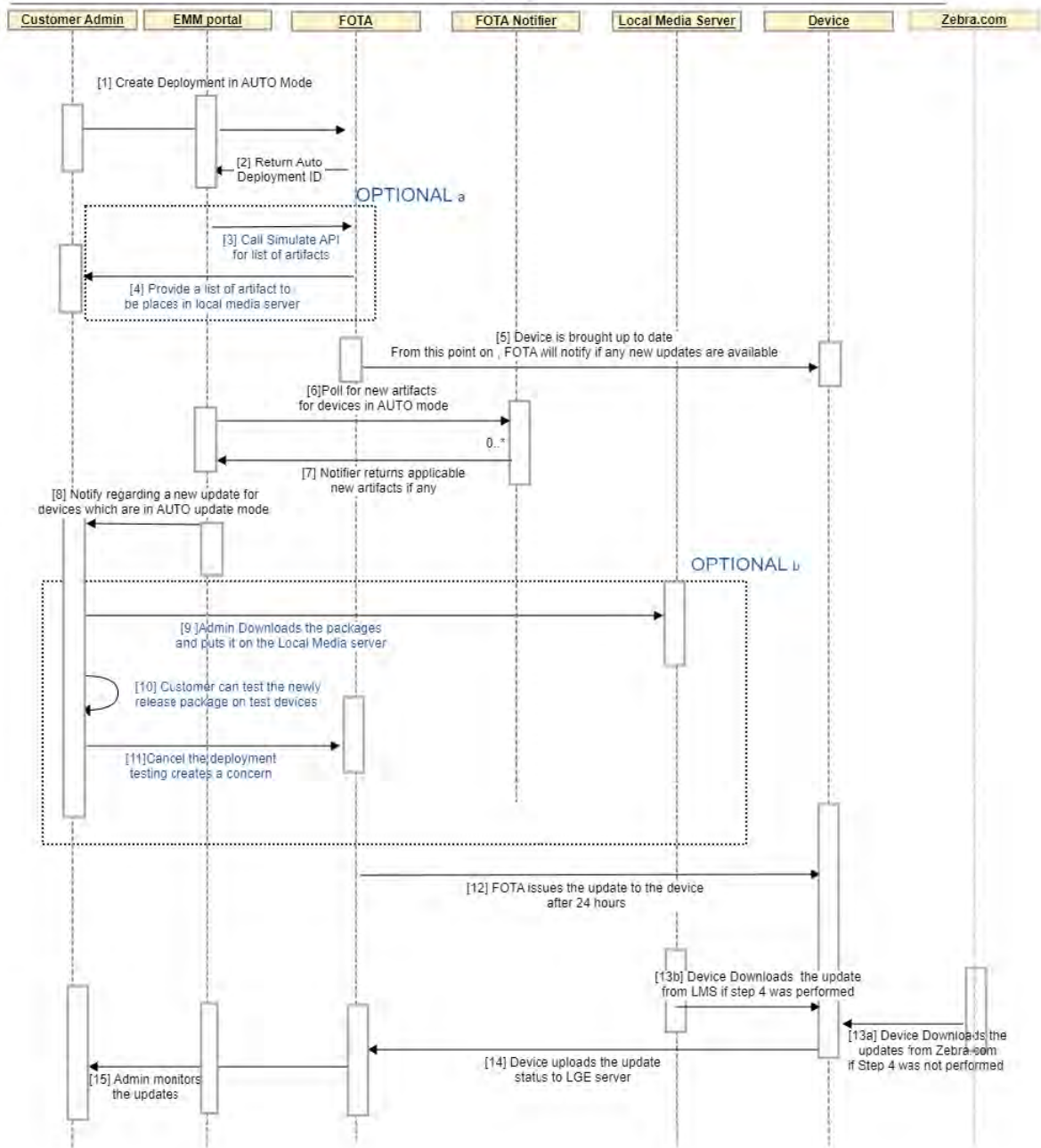
Zebra Managed Auto Update	Enterprise Managed Auto update
Full Automation	Flexible Automation
Zebra Managed Auto Update is fully automated with a fixed set of rules: <ul style="list-style-type: none"> <li>• Perform download and install over WLAN or WAN network connections</li> <li>• No support for Local Media Server</li> <li>• Updates can start at any hour of the day</li> <li>• End-users can postpone the update until a convenient time</li> <li>• Devices are updated one to three day of new update release</li> </ul>	Enables administrators to configure: <ul style="list-style-type: none"> <li>• Network and power rules</li> <li>• Local media server</li> <li>• Scheduled updates</li> <li>• Custom User and postpone messages</li> </ul> Updates can be tested before they are sent to the device by providing an auto update delay.
No EMM is required. Zebra managed auto updates can be enabled through: <ul style="list-style-type: none"> <li>• A staging profile</li> <li>• The system update settings on the device</li> </ul>	Requires an EMM which has integrated with LG OTA
Updates can be cancelled individually through: <ul style="list-style-type: none"> <li>• A staging profile by opting out of Zebra Auto Update</li> <li>• The system update settings on the device</li> </ul>	Updates can be cancelled centrally from a EMM server.
Supported on Android 11 and above.	Supported on Android N patch 19 and above.

## Enterprise Auto Update

The following diagram illustrates the procedure for configuring devices to use the enterprise auto update method. The optional steps (a and b), are applicable only if a local media server is used with auto update mode. When using a local media server with auto updates, the required files must be available on the local media server and accessible by the device. For more information on local media server requirements, see [Using a Local Media Server](#)

# Auto Update Mode

Consumer Mode , Notificaiton and Local Media Server (LMS)



## Step 1: Configure Devices for Auto update

To enable the auto update feature on a single device or set of devices, issue the auto deployment configuration request body for the deployment.

API FTS: POST /v2/deployments

All auto deployments create one master deployment. When auto updates are sent to a device, a sub-deployment is generated in addition to the main deployment.

For example, when sending an auto deployment, both the current release and the next release are generated. The deployment is as follows:

- AUTO-23
  - FOTA-32: Release one. Set when auto deployment was first created.
  - FOTA-33: Release two. Set for the next release.

The auto deployment request differs from a manual deployment request in the following ways:

- Simulate must be set to false in the request body.
- The **updateType** must be set to **AUTO** in the **profile** definition.
- In the **schedule** definition, **StartDate** is forced to +1 day (24 hours) from when the artifact was published.

If the request is valid and successful, then the response is sent containing the following information:

- All auto update deployments contain the prefix **AUTO**, followed by a unique number used to check the deployment status.
- Errors are recorded in the response message.
- If any of the devices sent in request already had auto update enabled, the configuration will fail.

After auto update is enabled on a device, use the Notification API to notify administrators when a new update is available for upload to the local media server. For more information, go to [Notification API](#).

## Auto Deployment Request Example

```
{
  "simulate":false
  "profile": {
    "updateType": "AUTO",
  },
  "mediaServer": [
    [
      {
        "artifactUrl": "media_server_url_root",
        "authHeaderName":"Authorization",
        "authHeaderValue":"type credentials"
      }
    ]
  ]
},
"schedule": {
  "mode": "SCHEDULED",
  "duration": "deployment duration in days"
},
"rules": {
  "download": {
    "startDate": "2018-07-25T10:20:32",
    "network": "ANY",
    "autoUpdateDelay" : 10
  },
  "install": {
    "startDate": "2018-07-25T10:20:32",
    "timeWindowStart": "19:00:00",
    "timeWindowEnd": "23:00:00",
    "allowUserPostpone": true,
    "maxPostponeDuration":5,
    "postponeMessage": " message",
    "userMessage": " message",
  },
},
"devices": {
  "model": "TC75",
  "serial": [
    "3453453",
    "353453453"
  ]
}
}
```

## Auto Deployment Response Example

```
{
  "data": {
    "deploymentId": "AUTO-1",
    "message": "AUTO update set for devices successfully"
    "code": "200",
    "timestamp": "2018-07-25T00:00:00+03:00"
  }
}
```

## Step 2: View Status of Auto Deployments

Similar to manual deployments, you can view both the simple and detailed status of deployments rolled out automatically for auto update enabled devices.

### Simple and Detailed Status

The simple and detailed status APIs are similar to manual deployments with the following exceptions:

- The input contains an auto update ID with the prefix **AUTO**.
- The deployment ID identifies the latest FOTA deployment applied on the devices.
- An associated deployments field lists new artifacts that are available for devices with auto update enabled.

## Auto Deployment Request Example

```
{
  "deploymentId": "AUTO-1",
  "state": "Active/All",
}
```

## Auto Deployment Response Example

```

{
  "head": {
    "page": 10,
    "offset": 50,
    "limit": 100,
    "total" : 1000
  },
  "data":
  [{
    "autoUpdateId":"AUTO-1"
    "deploymentTag": "abc",
    "deploymentId": "FOTA-1", //latest deployment id
    "deploymentStatus": "InProgress",
    "totalDevices": 1000,
    "created": 5,
    "scheduled": 5,
    "downloading": 20,
    "awaitingInstall": 10,
    "completed": 30,
    "cancelled": 5,
    "unknown": 5,
    "cancelRequested": "true",
    "failed": {
      "download": 20,
      "install": 10
    }
    "completedOn": "2018-07-25T00:00:00+03:00",
    "cancelledOn": "2018-07-25T00:00:00+03:00",
    "associatedDeployments":["FOTA-1","FOTA-2","FOTA-3"] // applicable only for auto
update
  ]}
}

```

### Step 3: Update Auto Deployment Configurations

Similar to manual deployments, auto deployments cannot be modified or updated. You must cancel the current deployment and configure a new deployment, as explained in [Step 1: Configure Devices for Auto update on page 67](#).

Canceling an auto deployment is similar to canceling a manual deployment, as shown in the following examples.

## Auto Deployment Cancel Request Example

POST /v2/deployments/cancel

```
{
  "deploymentId": "AUTO-1"
}
```

## Auto Deployment Cancel Response Example

```
{
  "data": {
    "deploymentId": "AUTO-1",
    "action": "cancel",
    "message": "The auto-update configuration was cancelled successfully",
    "code": "200",
    "timestamp": "2018-07-25T00:00:00+00:00"
  }
}
```

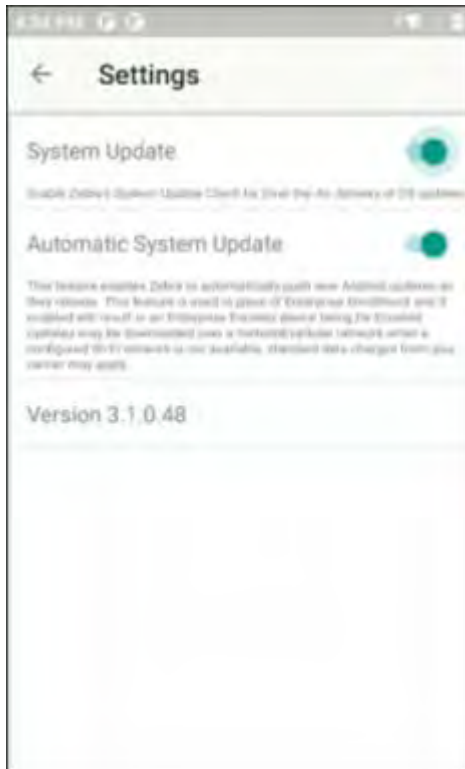
## Zebra Managed Auto Update

Enable Zebra Managed Auto Update on a device from device settings or StageNow.

### Enabling from Device Settings

1. Swipe up from the bottom of the screen. The All Apps screen displays.

2. Go to **Settings > System > System Update > Settings**.



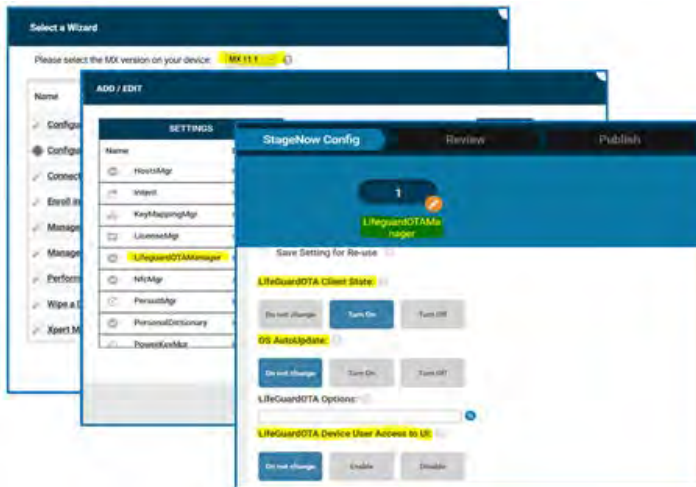
3. Select **Automatic System Update** to enable it.

### Enabling from StageNow

1. Install the latest Android 11 supported StageNow desktop tool from [zebra.com/stagenow](https://zebra.com/stagenow).
2. From StageNow, select **Create a new Profile**.
3. From MX version drop-down box, select MX 11.1 or later.  
MX 11.1 or later is required to enable auto updates from StageNow.
4. Select **Xpert Mode > Create**.
5. Enter a profile name and select **Start**.
6. From the SETTINGS tab, scroll down and select **LifeguardOTAManager**.  
LifeguardOTAManager appears in the CONFIG tab.
7. Select **Add**.



- From the StageNow Config tab, set LifeGuardOTA Client State and OS AutoUpdate to **Turn On**.

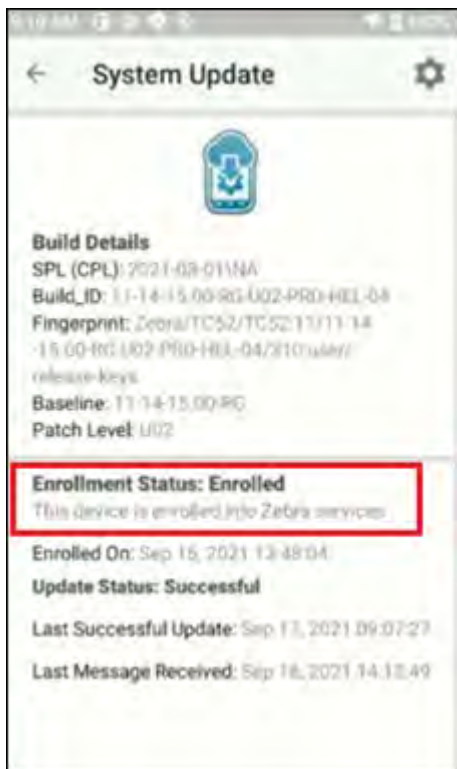


- Select **Continue** > **Complete Profiles**.

## Confirming Auto Update is Enabled

Confirm that Zebra Managed Auto Update is enabled from the System Update screen on the device.

- On the device, go to **Settings** > **System** > **System Update**.
- Check that Enrollment Status is set to **Enrolled** and that the enrollment status description is **This device is enrolled into Zebra services**, indicating that device updates are Zebra managed.



### Restrictions and Behavior

- When opted in to Zebra auto update, the device is removed from enterprise enrollment, and is not available for updates through EMM/ZDNA.
- The only way to get back to the enterprise tenant is via re-enrollment.
- Before the device opts in or opts out of Zebra Auto Update, make sure there are no on-going updates.
  - A Notification is presented on the device whenever a deployment is in progress or scheduled.
  - Wait for the update to complete, restart the device, and then opt in or out of the Zebra auto update.
- It is good practice to restart a device after opting in or out of Zebra managed auto update.
- New updates are automatically installed on the device about three days after they are released on Zebra.com. This gives administrators time to test updates.
- Approximately 200 MB of free disk space is required on the end-user device for a successful update.
- Updates use any available Internet connection, including wireless, wired, or cellular. Mobile data provider charges may apply.
- To cancel an update that has not started, switch off the Lifeguard System update client. Once an OS update has already started, it cannot be cancelled.
- When a device is enrolled in enterprise managed auto updates, it is automatically opted out of Zebra managed auto updates. After successful enrollment, the transition to enterprise manage auto updates takes about 15 minutes. During this time it is not recommended to send any deployments to the device.
- Zebra Data Service is required for LG OTA. Make sure that Zebra Data service is enabled on the device. For more information about Zebra data service and how to check if it is enabled, refer [techdocs.zebra.com/zds/setup/](https://techdocs.zebra.com/zds/setup/).

### Retry Mechanism in Zebra Auto Updates

- The FOTA system checks for the latest failed deployment of devices and performs a retry every day.
- If deployments continue to fail, then the FOTA system will retry only 3 times per device.
- If the device does not successfully update after the retries are exhausted, it will only retry again on the next OS update release.



**NOTE:** To force a device to retry, the user or admin can toggle the Automatic system update setting within the device settings or via OEMConfig / Mx controls.

### Skip Auto Updates

Administrators can stop auto updates for a preset amount of time or skip updates for a particular month by disabling the Lifeguard OTA client. When the OTA client is disabled, no new update deployments are sent to the device.

To skip update for particular month, disable the OTA client immediately after the new artifact notification is received and wait until the delay period specified in the auto deployment has ended. By default, the delay period is one day. Refer to the deployment API for more details.

When the OTA client is enabled, the auto deployment resumes and the device begins receiving updates for newly released artifacts.

Use the following methods for disabling and enabling the OTA client.

- StageNow and Mx - Enable or disable the OTA client through the Lifeguard Over the Air Manager. Refer to [techdocs.zebra.com/mx/fotamgr/](https://techdocs.zebra.com/mx/fotamgr/) for more details on configuration.
- OEMConfig - Push the **Turn On** or **Turn Off** Mode value to the device. Refer to [techdocs.zebra.com/oemconfig/11-1/mc/#firmwareovertheairconfiguration](https://techdocs.zebra.com/oemconfig/11-1/mc/#firmwareovertheairconfiguration).
- Device UI - Go to **Settings > System > Advanced > System updates > Settings** and turn the System Update toggle to the Off position.

## Special Considerations

The **StartDate** of auto deployment requests are forced to have a delay of at least one day from the date the artifact was published.

When multiple artifacts are released for the same device model before the completion date of the current deployment, only the last artifact that was released is deployed on the device. Earlier incomplete updates in the queue are cancelled.

If an installation is in progress and a new auto deployment is triggered, the new auto update fails and the older installation that is in progress continues. If this happens, the device does not have the latest version, but continues to receive future updates. This is typically the only time an auto update fails. Administrators can wait for the next auto update or create a manual deployment to update the device sooner.

## Deprecation of Simulate

The simulate tag in the Deployment API is deprecated in LGE OTA version 3.3 and later. It is recommended to use the Notification API to download artifacts.

The following table compares the deprecated simulate tag with the Notification API.

**Figure 2** Simulate and Notification API Comparison

	Simulated Deployment (simulate flag in Deployment API)	Notification API
Type	A simulate tag is passed during a deployment API request.  The default value of the simulate tag is false.	A separate dedicated API that is used to publish various events.
Availability	Has been available in Deployment API since January 2022.	<b>NEW_ARTIFACT_AUTO_UPDATE:</b> Available since January 2022 <b>NEW_DEPLOYMENT:</b> Available since March 2022.
Purpose	Simulate a deployment to obtain the following: <ul style="list-style-type: none"> <li>• Syntax issues</li> <li>• Check boundary conditions</li> <li>• Obtain the artifacts required to update devices</li> </ul>	Receive notifications regarding the following: <ul style="list-style-type: none"> <li><b>NEW_ARTIFACT_AUTO_UPDATE:</b> New artifacts for devices in auto update mode.</li> <li><b>NEW_DEPLOYMENT:</b> New artifacts required to upgrade a device to a specific Build ID.</li> </ul>
Pull / Push	Pull using http REST API	Pull using http REST API

**Figure 2** Simulate and Notification API Comparison

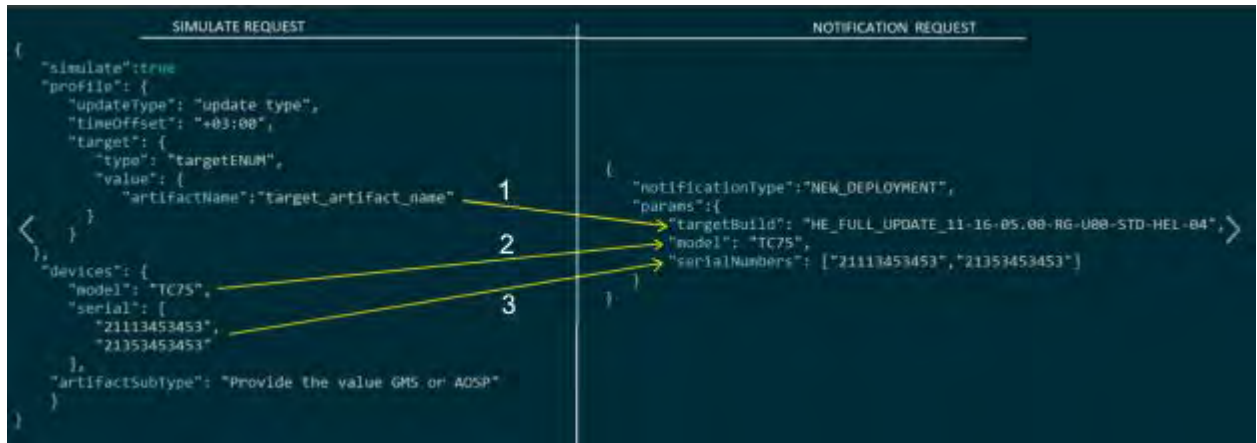
	Simulated Deployment (simulate flag in Deployment API)	Notification API
URL	https://api.zebra.com/devices/fts/deployment/v2/deployments	https://api.zebra.com/devices/fts/deployment/v1/notification
Input	Device serial number Model Target artifact	Input depends on the notification type. Auto update mode updates: <ul style="list-style-type: none"> <li>• Time range</li> <li>• Page size and offsets</li> </ul> Custom updates: <ul style="list-style-type: none"> <li>• Device serial number</li> <li>• Model</li> <li>• Target artifact</li> </ul>
Supported Deployment Types	Custom Deployment	Custom Deployment Auto update mode Deployment

## Migrating to Notification API

This section shows the request and response differences between creating simulated deployment and using the Notification API.

A request using the simulate tag had three parts, where a notification request only has one.

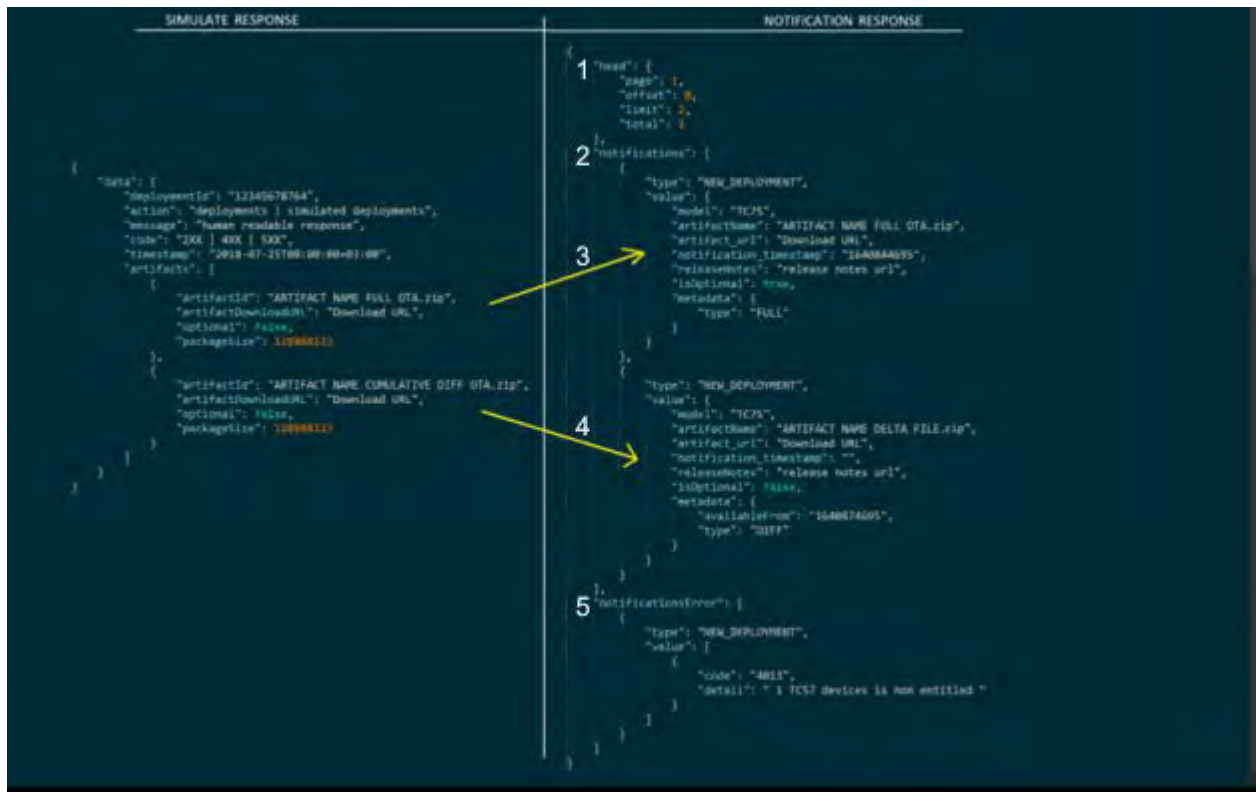
**Figure 3** Request Mapping



Item	Description
1	<b>artifactName</b> becomes <b>targetBuild</b> . You can obtain the <b>targetBuild</b> value from the same record of the OTA Update Access API as the file name.
2	<b>model</b> does not change.
3	<b>serial</b> array does not change.

A simulated deployment response contains one part, while a notification response is split into three main parts: head, notifications, and notificationErrors.

**Figure 4** Response Mapping



Item	Description
1	head contains a summary of the response.
2	<p>notifications contains response details.</p> <p>This section contains same elements as the simulated deployment response. However, the Notification API supports true delta packages, which are not supported by a simulated deployment. True delta packages are smaller and do not require a Full OTA package.</p> <p>Refer to the <b>optional</b> tag in the deployment package request to determine which packages are optional.</p>
3	The Notification API includes all the elements of a simulated deployment and additional parameters that provide enhanced features.
4	The Notification API includes all the elements of a simulated deployment and additional parameters that provide enhanced features.
5	<p>notificationErrors contains errors that were encountered during generation of the response. For example, if a device is not entitled, it returns 4013 error with a text message describing the error.</p> <p>The Notification API error codes are same as the Deployment API error codes. For error code descriptions, refer to the response codes listed in the Zebra OTA API Integration and Troubleshooting Guide.</p>

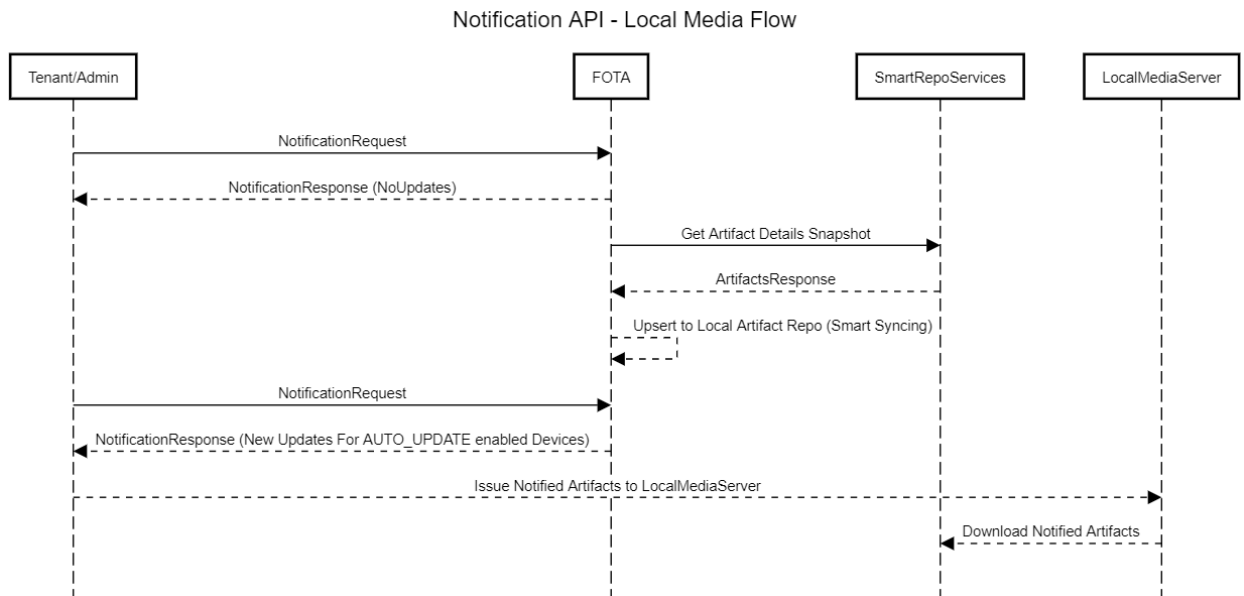
## Notification API

The Notification API gets a list of recently released OS versions and details for devices with auto update mode enabled. In OTA API version 3.3 and later, this feature is used to make recently released or updated artifacts available for download on a local media server.

Notification API requests must be made in the requested time window and preceded by the request validation.

## Notification API Flow

The following diagram illustrates the Notification API flow.



The following example shows a typical request body.

```

{
  "notificationType": "NEW_ARTIFACT_AUTO_UPDATE",
  "offset": 0,
  "limit": 100,
  "params": {
    "from_time": "2020-11-15T00:00:00+00:00",
    "to_time": "2020-11-17T00:00:00+00:00"
  }
}

```

The FOTA server responds with the recently available artifacts, which are constantly synced with the SmartRepo Services. The following example shows a typical pagination enabled response.

```
{
  "head": {
    "page": 10,
    "offset": 50,
    "limit": 100,
    "total" :1000
  },
  "notifications":[
    {
      "type":"NEW_ARTIFACT_AUTO_UPDATE",
      "value":{
        "model":"TC57",
        "artifactName":"HE_FULL_UPDATE_10-12-13.00-QG-U00-STD-HEL-04.zip",
        "artifact_url": ""
        "notification_timestamp":"",
        "releaseNotes":"release notes url"
      }
    },
    {
      "type":"NEW_ARTIFACT_AUTO_UPDATE",
      "value":{
        "model":"TC77",
        "artifactName":"HE_FULL_UPDATE_10-12-13.00-QG-U00-STD-HEL-04.zip",
        "artifact_url": ""
        "notification_timestamp":"",
        "releaseNotes":"release notes url"
      }
    }
  ],
  "notificationsError": [
    {
      "type": "NEW_ARTIFACT_AUTO_UPDATE",
      "value": [
        {
          "code": "4013",
          "value": [
            ],
          "detail":"3400 TC52 devices are non entitled ";
        },
      ],
    }
  ]
}
```

Using the details provided in the response, the customer administrator is able to download and keep the files on the local media server.



## Notification API Consumption

When there is a large number of devices or large packages, it is recommended to download the packages to a local media server. Using a local media server avoids issues with download speed and Internet connectivity. It also ensures that artifacts are downloaded as they become available. For more information, see [Using a Local Media Server](#).

This section provides pseudo code examples that describe the logic required to download the necessary artifacts to a local media server for LGE auto updates and custom deployments.

Terms used in the pseudo code are as follows:

- LMSartifcats - A map containing a list of artifacts for download.
- LastSuccessfulSyncTime - Time until the artifacts are processed.
- GetArtifacts scheduler - Scheduler that runs once in a day. It calls the Notification API to get new artifacts published in last 24 hours and adds to the LMSartifcats map. LGE decides if new artifacts are relevant or not based on auto updates created by the account that is currently logged in.
- DownloadArtifact - Runs once every two hours to download artifacts that are present in the LMSartifcats map. The scheduler also retries a failed downloads and artifacts in the building state, which is why it runs once every two hours.

Each artifact has a unique name and can be downloaded to a single directory (for example, `media`). The path to the local media server is entered as input for the auto update rule.

You can create another scheduler to delete unused artifacts that are not part of pseudo code. For example, you could delete files older than three months since they are probably not needed.

## Pseudo Code for Auto Updates

The following pseudo code demonstrates how an application can automate artifact downloads to the customer's local media server. In this code, OS update consumption happens automatically for devices that are in auto update mode.

The pseudo code assumes the following:

- The workflow for auto updates is fully automated.
- The developer is aware of the OAuth flow.
- All time references in the pseudo code are in UTC.

### Pseudo Code

The following pseudo code describes what happens from the time the application starts.

```
Create the media folder if does not exist.
```

```
Create an empty map named LMSartifcats (first time start) or load the LMSartifcats map from permanent storage. This map holds artifacts that will be downloaded to the local media server.
```

```
Create a variable named LastSuccessfulSyncTime and set the value between the current time and one month in the future (first time start), or load it from permanent storage.
```

```
Login with 3 legged oAuth
//OAuth flow start
Get the device code.
Authenticate with the URL.
```

## Auto Update Mode

Get AccessToken and RefreshToken (used to call notification API later).

```
//OAuth flow start
```

Create the GetArtifacts scheduler that starts at a random time of the day, everyday.

Create the DownloadArtifact scheduler which runs once every two hours.

GetArtifacts scheduler:

Execution starts

Call the RefreshToken API to get a new AccessToken.

Call the Notification API with a time window of LastSuccessfulSyncTime until the current time.

```
//notification API params:
```

```
//1. fromTime=LastSuccessfulSyncTime
```

```
//For to_time and notificationType, the default values for currentTime and NEW_ARTIFACT_AUTO_UPDATE are taken respectively.
```

Update LastSuccessfulSyncTime to the current time with a five minute buffer.

For each object in notification array:

If the file is not present in media folder,

add artifactName as the key and downloadURL as the value to the LMSartifcats map.

//Some artifacts may be optional (where isoptional and true). These artifacts are good to have and it is recommended to download them in case true diff packages are not available for a device source.

DownloadArtifact scheduler:

Execution starts.

Call the RefreshToken API to get an AccessToken.

Call the getDownloadToken API to get a DownloadToken.

For each entry in the LMSartifcats map:

If the file is not in the media folder,

download the file using download URL. //Make sure to use the same name provided in the artifactName response () while saving file.

Note: The API to download the artifact is a GET request and takes "token" as the header with a value of the generated downloadToken. Download tokens are short lived tokens and it is recommended to generate a new one each time. Refer to the OTA Update Access API documentation at [developer.zebra.com/apis/ota-update-access](http://developer.zebra.com/apis/ota-update-access).

On a success, delete the entry with artifactName from the LMSartifacts map.

On failure (error code 503), continue with the next download.

//503 is a rare error that may happen if a particular diff package was not built in LGE before. It may take up to two hours to build a diff package. 503

indicates that artifact is not ready for download. Do not remove the entry from the map so that it downloads during the next execution in two hours.

On Application stop:

- Abort any in progress downloads.

- Save LMSartifcats map to permanent storage.

- Save LastSuccessfulSyncTime to permanent storage.

//It is necessary that this application is running all the time to keep the new artifacts updated in local media server. In case it stops due to errors, better to have auto restart implemented so that auto deployments are not affected

### Pseudo Code for Manual Deployments

For administrators to manage files in local media servers, EMM vendors provide a user interface by manually downloading the necessary package for custom deployments. This option requires the EMM vendor to provide a user interface for administrators to view and download artifacts and a way to refresh the download when the artifact is in building state or failed to download.

#### Pseudo Code

The following pseudo code allows an administrator to download an artifact or refresh a download through a user interface button.

- Call the `getDownloadToken` API to get a `downloadToken` //used for downing artifacts

- Call the `Notification` API with the target artifact, serial numbers, and notification type set to `NEW_DEPLOYMENT`.

- //Response may contain one or more than one artifacts

- For each object in the notification array:

  - If the file is not present in the media folder,

    - download the file using `downloadURL`. //make sure to use same name provided in `artifactName` response () while saving file and do not change the file name.

    - Note: Download artifact API is get request and takes "token" as header with value as generated "downloadToken". download tokens are short lived tokens and it is recommended to generate new one each time.Refer to "OTA Update Access" API documentation - <https://developer.zebra.com/apis/ota-update-access>

  - On success, save the file to media folder

  - On failure (error code 503), display the error text: "The artifact is not ready, try again after two hours." Make sure to provide a refresh button to retry downloads.

    - //503 is a rare error that may happen if a particular diff package was not built in LGE before. It may take up to two hours to build a diff package. 503 indicates that artifact is not ready for download. Do not remove the entry from the map so that it downloads during the next execution in two hours.

# Appendix

This section provides ENUM values, deployment and job status triggers, and a description of the date and time format.

## ENUMS

This section lists the ENUM values available for the OTA API.

### Reset Types

```
[  
  "NONE",  
  "REBOOT",  
  "ENTERPRISE_RESET"  
]
```

### Network Types

```
[  
  "ANY",  
  "WIFI",  
  "WIFI_CELLULAR"  
]
```

### Deployment Query Status

```
[  
  "ALL",  
  "ACTIVE",  
  "CREATED",  
  "INPROGRESS",  
  "COMPLETED"  
]
```

**updateType**

```
[  
  "CUSTOM",  
  "LATEST",  
  "AUTO",  
]
```

**target**

```
[  
  "AN",  
  "AV",  
  "RN"  
]
```

**Deployment Status**

```
[  
  "CREATED",  
  "INPROGRESS",  
  "COMPLETED"  
]
```

**Job Status**

```
[  
  "CREATED",  
  "INPROGRESS",  
  "FAILED",  
  "COMPLETED",  
  "CANCELLED"  
]
```

## Detailed Job Status

```
[
  INCOMING_JSON_RCVD,
  INCOMING_JSON_INVALID,
  MX_SETTINGS_FAILED,
  DOWNLOAD_PENDING,
  DOWNLOAD_IN_PROGRESS,
  DOWNLOAD_COMPLETE,
  DOWNLOAD_FAILED,
  MANIFEST_IN_PROGRESS,
  MANIFEST_FAILED,
  INSTALL_AW,
  INSTALL_TRIGGER,
  INSTALL_SUCCESS,
  INSTALL_FAILURE,
  INSTALL_FAILED,
  INSTALL_IN_PROGRESS,
  EXECUTE_ENTERPRISE_RESET,
  DATA_CLEARED,
  CANCELLED,
  WAITING_FOR_REBOOT,
  DOWNLOAD_FAILED_RETRY
]
```

## Request Status

```
[
  "SUCCESS",
  "FAIL"
]
```

## authType

```
[
  "BASIC",
  "TOKEN"
]
```

**Device Model**

"TC75x"  
"TC52"  
"TC56"  
"VC80x"  
"TC72"  
"ET55"  
"CC600"  
"TC51"  
"L10A"  
"PS20J"  
"TC57"  
"TC25"  
"ET50"  
"TC70x"  
"TC77"  
"CC6000"  
"TC8300"  
"VC8300"  
"ET51"  
"ET56"  
"MC33"  
"WT6000"  
"MC93"  
"EC30"  
"TC20"  
"TC21"  
"TC26"  
"CC605"  
"CC610"  
"L10AW"  
"MC33X"  
"TC57X"  
"MC93"  
"MC2200"  
"MC2700"  
"TC52X"  
"WT6300"  
"EC50"  
"EC55"  
"MC3300x"  
"MC3300a"  
"TC52AX"  
"RZ-270"  
"MC20"

## Android Desserts

```
[  
  "KitKat",  
  "Lollipop",  
  "Marshmallow",  
  "Nougat",  
  "Oreo",  
  "Pie"  
  "A10"  
  "A11"  
]
```

## installMode

```
[  
  "INSTALL_NOW",  
  "SCHEDULED"  
]
```

## notificationType

```
[  
  "NEW_ARTIFACT_AUTO_UPDATE"  
]
```

## deployment\_query status

```
[  
  "ALL",  
  "ACTIVE"  
]
```



## Deployment Status Triggers

This section describes the triggers for each deployment status change.

### CREATED

This is the initial state of the deployment. When it is first created, the deployment remains in this state for a very short period of time.

### INPROGRESS

The deployment status changes to **INPROGRESS** when jobs are created for each device and messages are sent to the devices with the deployment dates.

The deployment remains in this state until the schedule end date.

### COMPLETED

The **COMPLETED** status is triggered when the deployment end date has passed or when the total device count equals the installed device count plus the failed device count.

A **COMPLETED** deployment status does not guarantee that the deployment was successful. It only indicates that no new activity will be initiated. Use Job Status to view how many device upgrades succeeded or how many device upgrades failed.

## Job Status Triggers

This section describes the status triggers for each job status change.

### CREATED

This state is triggered when a job is created. Normally, this is immediately after the deployment is created.

### INPROGRESS

The job status changes to **INPROGRESS** when the device reports the following sub-states.

Device internal State (not visible in FOTA API)	Contributes To	Triggers
<b>INCOMING_JSON_RCVD</b>	totalInProgress	The deployment message is received by the device.
<b>DOWNLOAD_PENDING</b>	totalInProgress	Waiting for download to being (download start date).
<b>DOWNLOAD_IN_PROGRESS</b>	totalInProgress	Download is in progress.
<b>DOWNLOAD_COMPLETE</b>	totalDownload	Download is in complete.
<b>MANIFEST_IN_PROGRESS</b>	totalDownload totalInProgress	Device is doing a manifest check to determine if the supplied artifact is appropriate for the device.
<b>INSTALL_AW</b>	totalDownload totalInProgress	Waiting for install.

Device internal State (not visible in FOTA API)	Contributes To	Triggers
<b>INSTALL_TRIGGER</b>	totalDownload totalInProgress	Installation has started.
<b>INSTALL_IN_PROGRESS</b>		Installation is in progress.
<b>DOWNLOAD_FAILED_RETRY</b>	totalInProgress	Download failed. FOTA client will reattempt the download.

## COMPLETED

The job status changes to **COMPLETED** when the device reports the following sub-states.

Device internal State (not visible in FOTA API)	Contributes To	Triggers
<b>INSTALL_SUCCESS</b>	totalInstall	The package and all intermediary packages are installed.

## CANCELLED

The job status changes to **CANCELLED** when the device reports the following sub-states.

Device internal State (not visible in FOTA API)	Contributes To	Triggers
<b>CANCELLED</b>	totalCancel	Job cancelled by the application.

## UNKNOWN

The job status changes to **UNKNOWN** when the deployment end date passes.

Device internal State (not visible in FOTA API)	Contributes To	Triggers
<b>UNKNOWN</b>	totalUnknown	Device did not report a status. State is unknown.

## FAILED

The job status changes to **FAILED** when the device reports the following sub-states.

Details for the error are available in the **Detailed Status** message attribute.

Device internal State (not visible in FOTA API)	Contributes To	Triggers
<b>INCOMING_JSON_INVALID</b>	totalDownloadFail, totalFailedCount	The Deployment JSON supplied to device has errors. These errors are mostly filtered out at the FOTA end. This is an additional security net and is important if direct device API becomes supported.
<b>MANIFEST_FAILED</b>	totalInstallFail	The provided artifact is not suitable for the selected device.
<b>INSTALL_FAILURE</b>	totalInstallFail, totalFailedCount	The installation failed.
<b>MX_SETTINGS_FAILED</b>	totalDownloadFail, totalFailedCount	MX System is not available or down on the device.
<b>DOWNLOAD_FAILED</b>	totalDownloadFail, totalFailedCount	The download failed.
<b>DA_DISABLED</b>	totalFailedCount	Deployment failed because the ZDS client was disabled on the device. ZDS is required for FOTA .
<b>FOTA_DISABLED</b>	totalFailedCount	Deployment failed because FOTA client was disabled on the device. This is the last message that FOTA sends before disabling itself.
<b>SPL_CHECK_FAILED</b>	totalFailedCount	The deployment failed because the device contains a newer SPL compared to the requested update.
<b>INVALID_SESSION_ID</b>	totalFailedCount	The deployment failed because the device has incorrect enrollment information.

## Date Time Format

All date and time formats must be in ISO-8601.

### Examples

- Date time in UTC will be represented as **2019-02-13T11:33:18+00:00**.
- Date time in device local time zone will be represented as **2019-02-13T11:33:18**.
- Start Date in UTC will be represented as **2019-02-13** and sent to device as **2019-02-13T00:00:00+00:00**.
- Start Date in device time will be represented as **2019-02-13** and sent to device as **2019-02-13T00:00:00**.
- End Date in UTC will be represented as **2019-02-13** and sent to device as **2019-02-13T23:59:59+00:00**.
- End Date in device time will be represented as **2019-02-13** and sent to device as **2019-02-13T23:59:59**.
- Time will be represented as 20:23:30.
- Time zone of the deployment will be represented as **+05:30 (india time zone)**.

ISO-8601 Date is represented in following UTC formats:

- 2019-07-12T10:55:48+00:00
- 2019-07-12T10:55:48Z

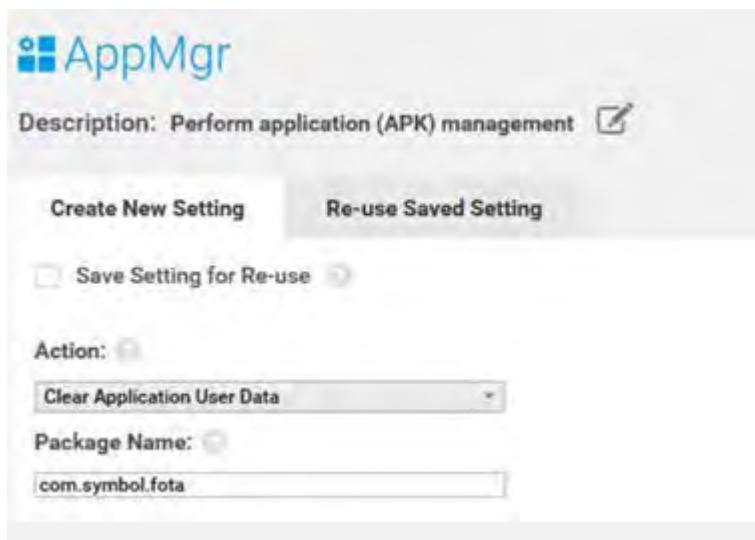
## Clearing Enrollment Information

LGE OTA API provides a way to de-enroll devices on the server. Before performing a new enrollment, devices that have been de-enrolled should have previous enrollment information cleared from the device by performing a factory or enterprise reset.

In cases where the device will not be reset, enrollment information can be cleared from the device by clearing the FOTA and enrollment manager application cache using the Application Manager in StageNow.

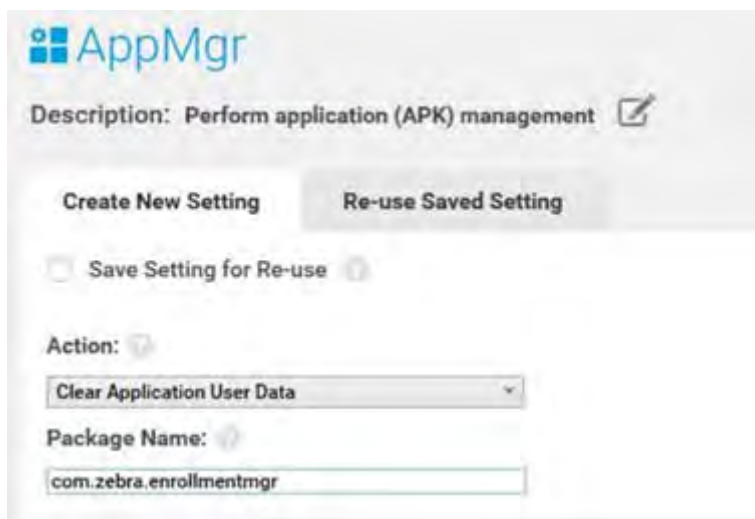
The screen shots below display the exact values to be used.

- To clear enrollment information from FotaClient:



The screenshot shows the AppMgr interface. At the top, there is a description: "Perform application (APK) management". Below this, there are two buttons: "Create New Setting" and "Re-use Saved Setting". A checkbox labeled "Save Setting for Re-use" is present. The "Action:" dropdown menu is set to "Clear Application User Data". The "Package Name:" text box contains the value "com.symbol.fota".

- For Clearing Enrollment information from enrollment Manager:



The screenshot shows the AppMgr interface. At the top, there is a description: "Perform application (APK) management". Below this, there are two buttons: "Create New Setting" and "Re-use Saved Setting". A checkbox labeled "Save Setting for Re-use" is present. The "Action:" dropdown menu is set to "Clear Application User Data". The "Package Name:" text box contains the value "com.zebra.enrollmentmgr".

## A13 Behaviour on SDM660 Models

Android 13 presents some unique challenges for organizations planning to migrate their [Zebra SDM660 platform devices](#) from earlier Android versions. Updating the devices to Android 13 requires conversion from Full-Disk Encryption to File-Based Encryption, will result in loss of applications, data, and settings when the partition is rebuilt using the new encryption.

For detailed information, please refer [A13 Guide for SDM660](#).

FOTA offers a special upgrade mechanism to prevent data loss while attempting to upgrade SDM660 devices from A11 and below. It requires prerequisite actions by the administrator.

**IMPORTANT:** The below steps are required for the admin before attempting updates to A13.

- Deploy the input config XML file to the device
- Install the [File Guardian](#) application on the device (manually upload by the admin).



**IMPORTANT:**

- Enterprise Auto Update - Updates only up to A11. A13 support is coming soon.
- Deployment - "updateType": "LATEST" if the above prerequisites are met, the device will upgrade to A13. Otherwise, the upgrade will only be the latest A11.
- Deployment - "updateType": "CUSTOM" if the prerequisites are not met, and the target is A13, the upgrade will fail with error code 4021.

## A13 Behaviour in EMMs

1. Enrollment Manager should be upgraded from Playstore to accept managed configuration.
2. CTL installation via Playstore should be skipped for A13.

