



LDI Plus API Guide



Release Version: 2.0 Release Date: 14 April 2022

Copyright/Disclaimer

Copyright © 2022, Lenovo. All rights reserved.

This document contains proprietary information of Lenovo that is protected by copyright and other intellectual property law which restricts its usage. The content in this document is subject to change without notice. If you find any issues in the documentation, please report to us in writing. Lenovo gives no representations or warranties of any kind regarding its content, including accuracy or completeness. Please do not reproduce or transmit any part of this document in any form or by any means, electronic or mechanical, for any purpose, without a written approval from Lenovo.

Contents

1	RESTful API	3
1.1	Purpose	3
1.2	Audience	3
1.3	Get API Credentials	3
1.4	Learn API Operations	6
1.5	Try APIs	7
1.6	Examples of API Methods	7
1.6.1	<i>Authentication - API token session</i>	7
1.6.1.1	HTTP Samples	8
1.7	Negative API Sample	9
1.7.1	<i>Groovy ACME Test</i>	9
1.8	User Management	10
1.8.1	<i>GET Users</i>	12
1.8.2	<i>Create User</i>	13
1.8.3	<i>Delete User</i>	15
1.9	Devices	16
1.9.1	<i>ACME Client Code</i>	16
1.9.2	<i>HTTP Request Responses</i>	16
1.10	Fleet Management	17
1.11	Insights Tests	18
1.11.1	<i>Request</i>	18
1.12	Issues Filter	19
1.13	Mark Issue as Resolved	20
1.14	Sensors	22

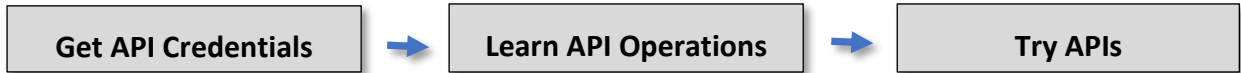
1 RESTful API

1.1 Purpose

The purpose of this guide is to inform you how to generate API credentials from your organization admin account, authenticate APIs and use them to integrate LDI with external platforms or applications.

1.2 Audience

This guide is for IT Administrators, Managers, and Developers.

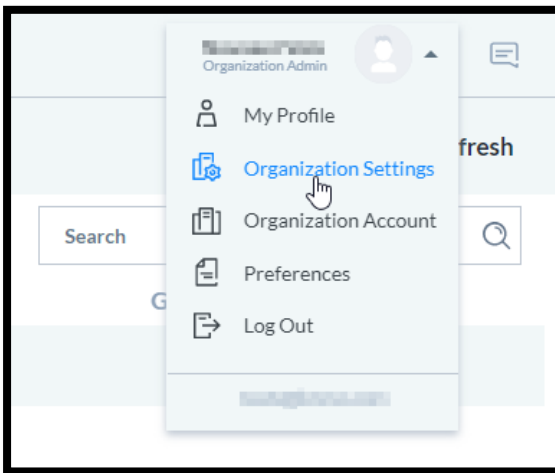


1.3 Get API Credentials

Note: You must have an Organization Administration account in LDI portal to generate API credentials.

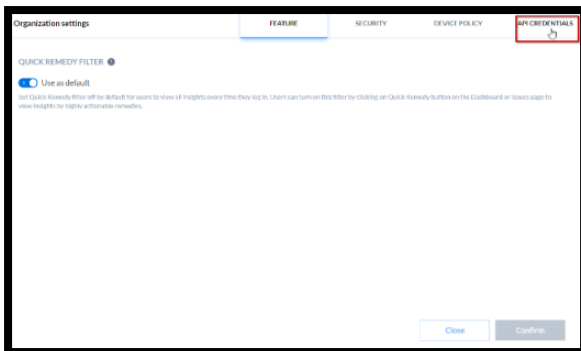
Before you use LDI APIs, you must generate API credentials in the LDI account.

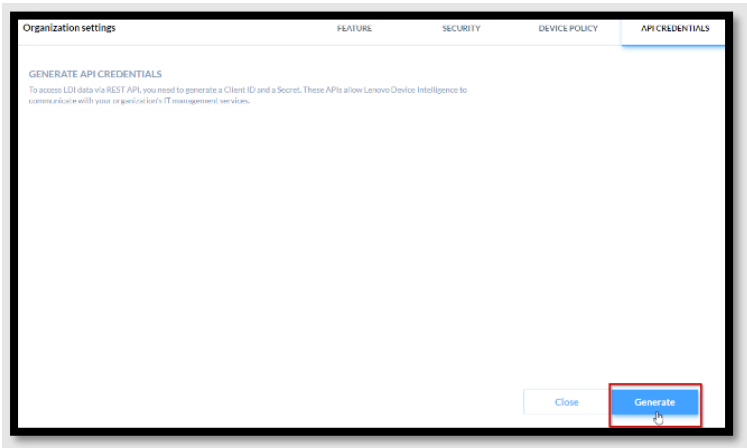
1. Click **Organization Settings** in the Users drop-down list.




2. Click **API Credentials**. You see API Credentials pane.

Note: If there are no API credentials, you must generate them.



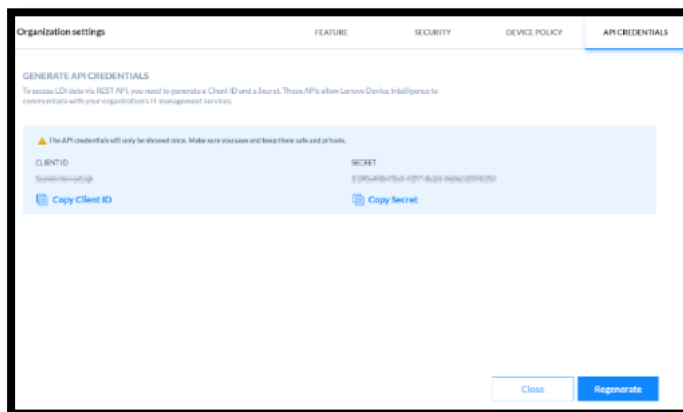


Note: A Client ID and Secret key are generated. You can copy them to the clipboard.

WARNING  **Keep the API credentials in a secure place and regenerate them over the time in accordance with business policies of your organization.**

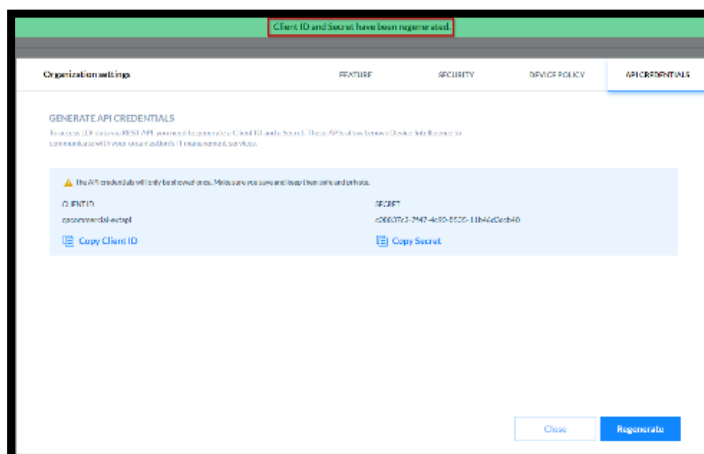
If you want to change the existing API credentials, you must generate a new one.

4. Click **Regenerate**.



5. In the **Regenerate Secret** window, click **Regenerate**.

6. A new Client ID and Secret key is generated.



The Client ID and Secret key do not expire until you regenerate a fresh pair.

After you have generated API credentials, you can use the following URLs to access different API endpoints.

Type the URL : <https://auth.naea1.uds.lenovo.com> or <https://auth.euwe1.uds.lenovo.com>, depending upon the organization region the devices are located.

[Generate LDI API credentials](#) (Client ID and Secret).

LDI API URLs
1. NA
External API: https://api.naea1.uds.lenovo.com
Authentication: https://auth.naea1.uds.lenovo.com
2. EU
External API: https://api.euwe1.uds.lenovo.com
Authentication: https://auth.euwe1.uds.lenovo.com
NA is North American Region and EU is European Union Region.

Note: After you generate a new pair of Client ID and Secret key, the older pair gets invalid.

The bearer token is a type of an access token that uses Auth 2.0 and expires within 30 minutes. You use the bearer token to get a new Access token. To get an access token you send the Authentication server this bearer token along with your client id. This way the server knows that the application using the bearer token is the same application that the bearer token was created for.

Generate API bearer token using External API.

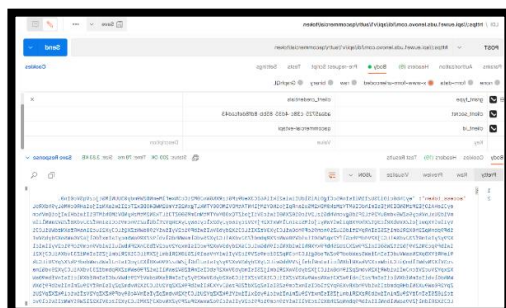
The URL - Base URL + /api/v1/auth/<organization_name>/token

Note: The Base URL depends on your region whether NA or EU.

Body should be x-www-form-urlencoded and should contain:

- grant_type: client_credentials
- client_secret: secret (from api credentials page)
- client_id: id (from api credentials page)

Note: Generate a new bearer token when it expires after 30 minutes.



```

private synchronized Tuple2<String, Instant> fetchNewToken() {
    Instant now = Instant.now();
    var response = authClient
        .POST_FORM("/api/v1/auth/" + realm + "/token",
            Map.of("grant_type", "client_credentials",
                "client_id", clientId,
                "client_secret", clientSecret));
    if (response.code >= 400) throw new RuntimeException("Token not
retrieved: " + response.asString());
    var body = response.asMap();
    String accessToken = (String) body.get("access_token");
    int expiredIn = ((Number) body.get("expires_in")).intValue();
    return new Tuple2<>(accessToken, now.plusSeconds(expiredIn - 15));
}

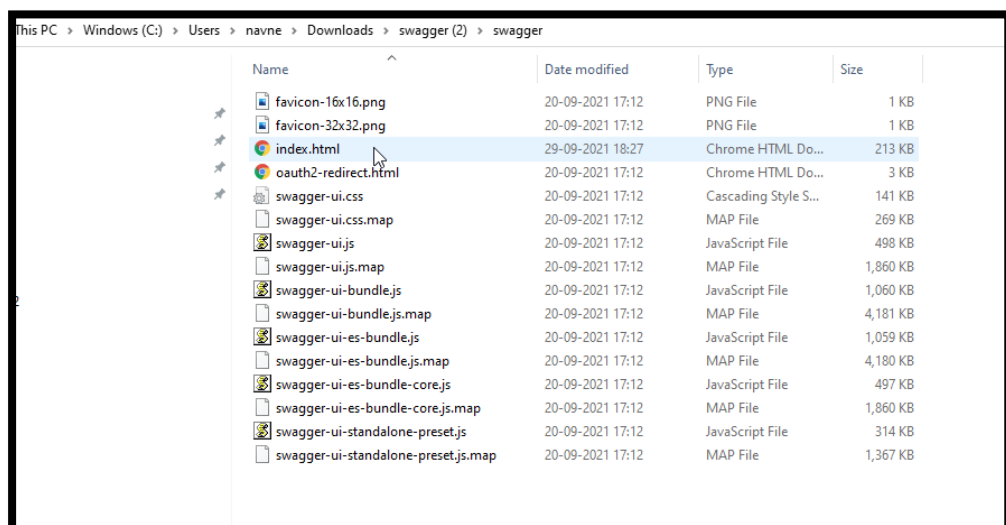
public synchronized String getToken() {
    if ((token == null) || (token.getV2().isAfter(Instant.now())))

```

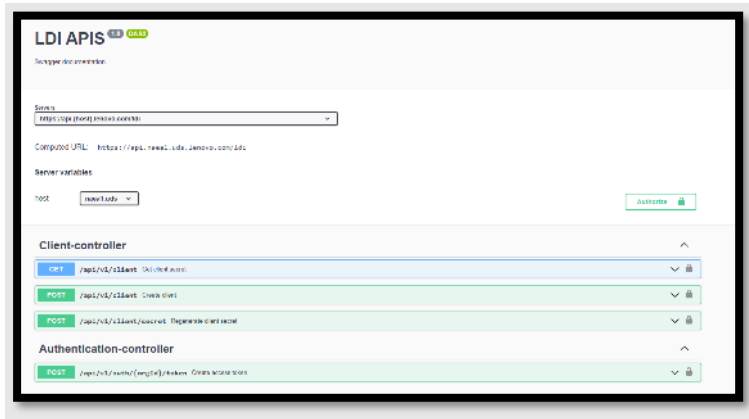
1.4 Learn API Operations

Swagger specification archive contains a folder with an index.html file and some other JavaScript files.

1. Download Swagger Specification zip file from the [support site](#) which provides you details about each API.
2. Extract the files in the folder.



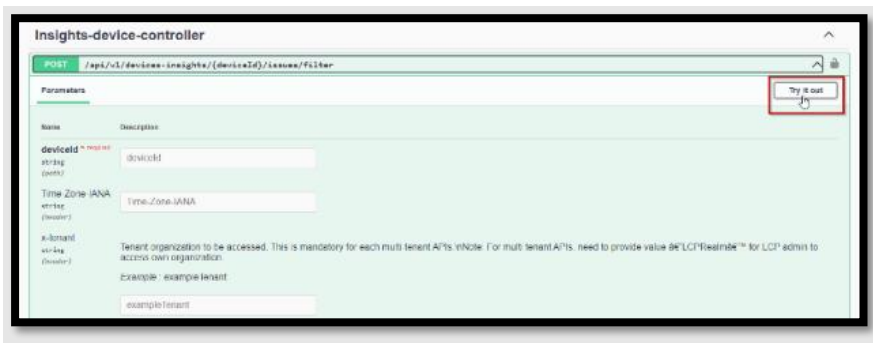
3. Double-click to open the index.html. You see the LDI APIs home page in Swagger.



4. Generate a bearer token (Refer [Get API Credentials](#) and Postman example).

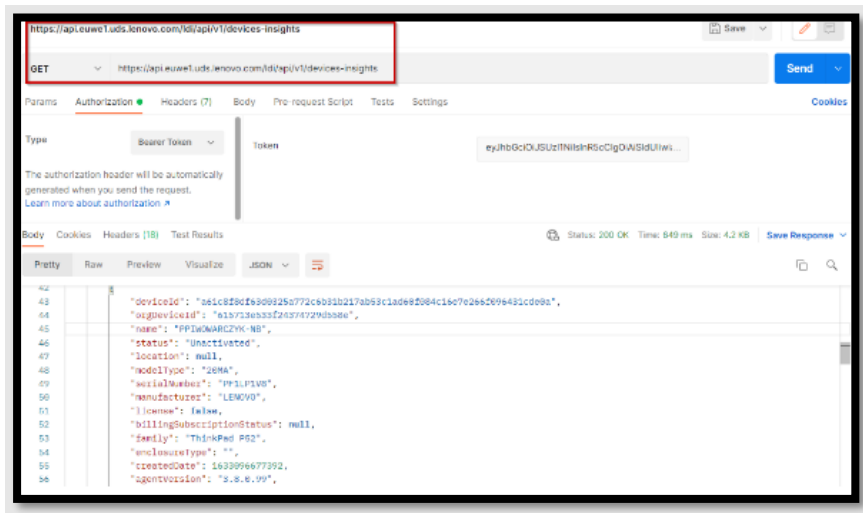
1.5 Try APIs

1. Select an API method in Swagger.



2. Use the bearer token in the call.

3. Click Try it out.



1.6 Examples of API Methods

1.6.1 Authentication - API token session

ACME JAVA CODE

```
package com.acme.lidi.test;

public class LdiClientTest {

    @org.junit.Test
    void tokenTest() {

        // parameters depends on geography
        String authUrl = "https://api.uds-qa.lenovo.com";
        String apiUrl = "https://api.uds-qa.lenovo.com";
        // realm name supplied by sales team
        String realm = "autoticketing";
        // client id to be supplied by integration support team
        String clientId = "autoticketing-extapi";
        // client secret gotten via UI self-service
        String clientSecret = "0bf4c041-a9b1-4133-9045-73795a254439";

        LdiClient client = new LdiClient(apiUrl, realm, clientId,
clientSecret);
    }
}
```

Lenovo SDK snapshot

POST <https://api.uds-qa.lenovo.com/lidi/api/v1/auth/autoticketing/token>
Content-Type: application/x-www-form-urlencoded

client_id=autoticketing-extapi&client_secret=0bf4c041-a9b1-4133-9045-73795a254439&grant_type=client_credentials

```
private synchronized Tuple2<String, Instant> fetchNewToken() {
    Instant now = Instant.now();
    var response = apiClient
        .POST_FORM("/ldi/api/v1/auth/" + realm + "/token",
        Map.of("grant_type", "client_credentials",
            "client_id", clientId,
            "client_secret", clientSecret));
    if (response.code >= 400) throw new RuntimeException("Token not
retrieved: " + response.asString());
    var body = response.asMap();
    String accessToken = (String) body.get("access_token");
    int expiredIn = ((Number) body.get("expires_in")).intValue();
}
```


HTTP Response

```
200
access-control-allow-credentials: true
cache-control: no-store
content-length: 3269
content-security-policy: frame-src 'self'; frame-
ancestors 'self' https://portal.uds-qa.lenovo.com https://developer.naea1.uds-
qa.lenovo.com; object-src 'none';
content-type: application/json
date: Tue, 28 Sep 2021 11:47:12 GMT
pragma: no-cache
referrer-policy: no-referrer
server: Lenovo
set-cookie: KC_RESTART=; Version=1; Expires=Thu, 01-Jan-1970 00:00:10 GMT;
Max-Age=0; Path=/auth/realms/autoticketing/; Secure; HttpOnly
strict-transport-security: max-age=31536000; includeSubDomains
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block
{"access_token":"eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUiiwia2lkIiA6ICJwZG1iME41
R
...(truncated text)
zU0Nk1CX0RBOFJUOyKRZ-H716fn1Fwk54eCn2vouFmKFz2frAuR9kE-
bqp3AhTSOuT6nlb4HGmSrMNNkYbg","expires_in":15552000,"refresh_expires_in":0,"to
ken_type":"Bearer","not-before-policy":0,"scope":"email profile"}
```

1.7 Negative API Sample

1.7.1 Groovy ACME Test

```
@Test
void tokenNegativeTest() {
    var client = new com.lenovo.ldi.client.LdiClient(authUrl, apiUrl, realm,
clientId, clientSecret + "_INVALID");
    var response = client.authClient
        .POST_FORM("/auth/realms/" + realm + "/protocol/openid-
connect/token",
        [grant_type : "client_credentials",
client_id : clientId,
client_secret: clientSecret])
    assert response.code == 401
}
```

HTTP Request

```
POST https://auth.uds-qa.lenovo.com/auth/realms/autoticketing/protocol/openid-connect/token
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&client_secret=0bf4c041-a9b1-4133-9045-73795a254439-INVALID&client_id=autoticketing-extapi
```

HTTP Response

```
401
access-control-allow-credentials: true
cache-control: no-store
content-length: 75
content-security-policy: frame-src 'self'; frame-ancestors 'self' https://portal.uds-qa.lenovo.com https://developer.naea1.uds-qa.lenovo.com; object-src 'none';
content-type: application/json
date: Tue, 28 Sep 2021 13:29:06 GMT
pragma: no-cache
referrer-policy: no-referrer
server: Lenovo
strict-transport-security: max-age=31536000; includeSubDomains
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
```

1.8 User Management

```
package com.acme.ldi.test
import org.junit.jupiter.api.Test
class UserTests extends BaseLdiTestClass {
    @Test
    void createNewUser() {
        var response =
this.client.authenticatedRestConnector.POST("/ldi/api/v1/users",
```

```

        [loginId          : "ccretoiu@lenovo.com",
         compositeRoleName: "pm_org_admin",
         firstName       : "ABC",
         creatorId       : "autoticketing",
         country         : "USA",
         email           : "ccretoiu@lenovo.com",
         lastName        : "ABC"])
        assert response.code == 409 // Expected conflict
    }
    @Test
    void getAllUsers() {
        var response =
this.client.authenticatedRestConnector.GET("/ldi/api/v1/users")
        assert response.code == 200
        var body = response.asMap()
        assert body.keySet() == ['_embedded', 'page', 'responseType'] as Set
        var userList = body._embedded.userList
        var loginIds = userList*.loginId
        println "${loginIds.size()} users found: ${loginIds}"
    }
    @Test
    void createAndDeleteUser() {
        var seed = Math.random().toString().replaceAll(/[^\d]/, '')
        var response =
this.client.authenticatedRestConnector.POST("/ldi/api/v1/users",
        [loginId          : "sdragos.${seed}@lenovo.com",
         compositeRoleName: "pm_org_admin",
         firstName       : "ABC",
         creatorId       : "autoticketing",
         country         : "USA",
         email           : "sdragos.${seed}@lenovo.com",
         lastName        : "ABC"])
        assert response.code == 201
        var searchResult =
this.client.authenticatedRestConnector.GET("/ldi/api/v1/users",
[freeText: "sdragos.${seed}@lenovo.com"])

```

```

        assert searchResult.asMap()._embedded.userList*.loginId ==
["sdragos.${seed}@lenovo.com"]
        var userDetails = searchResult.asMap()._embedded.userList[0]

        var deleteResponse =
this.client.authenticatedRestConnector.PATCH("/ldi/api/v1/users",
            ["userList" : [userDetails.userId],
            "operation": "DELETE"]
        )
        var secondSearchResult =
this.client.authenticatedRestConnector.GET("/ldi/api/v1/users",
[freeText: "sdragos.${seed}@lenovo.com"])
        var usersList=secondSearchResult.asMap()?._embedded?.userList
        assert (usersList?.collect { it['loginId'] } ? : []).empty
    }
}

```

1.8.1 GET Users

HTTP Request

```
GET https://api.uds-qa.lenovo.com/ldi/api/v1/users
```

```
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTUyZDZlY509LuBw
... (truncated text) vxilXFgr4gKVbCfnnVUScQXPkcGF2aqifGEQNaLiIXsBWM8iAX8smq-

```

API Response

```

200
cache-control: no-cache, no-store, max-age=0, must-revalidate
content-security-policy: default-src 'self'; connect-src *.uds-qa.lenovo.com;
style-src 'self' 'unsafe-inline'; img-src 'self' data:; script-
src 'self' 'unsafe-inline'; object-src 'none';
content-type: application/json
date: Wed, 29 Sep 2021 10:31:13 GMT
expires: 0
pragma: no-cache
referrer-policy: no-referrer
server: Lenovo
strict-transport-security: max-age=31536000; includeSubDomains

```

```
transfer-encoding: chunked
x-content-type-options: nosniff
x-envoy-upstream-service-time: 349
x-frame-options: DENY
x-xss-protection: 1; mode=block
```

```
{"_embedded":{"userList":[{"userId":"7204a566-1495-4be8-8c76-19899adf508d"...
(truncated
text)...{"number":0,"size":8,"totalElements":8,"totalPages":1},"responseType":"P
AGE"}
```

Stdout

```
8 users found: [shuma@lenovo.com, sdragos@lenovo.com, penghong2@lenovo.com,
penghong221062911034172@lenovo.com, penghong2+21063007144784@lenovo.com,
otsiupa@lenovo.com, ccretoiu@lenovo.com, ladamestean1@lenovo.com]
```

1.8.2 Create User

POST <https://api.uds-qa.lenovo.com/ldi/api/v1/users>

```
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IiJpIiwiaXNjaXkiOiA6IiwiaWF0IjoiYXZyN2ZlME41RzU0Nk1CX0RBOFJUOW
...(truncated text)... -4NQ
Content-Type: application/json; charset=utf-8
{
  "loginId": "ccretoiu@lenovo.com",
  "compositeRoleName": "pm_org_admin",
  "firstName": "ABC",
  "creatorId": "autoticketing",
  "country": "USA",
  "email": "ccretoiu@lenovo.com",
  "lastName": "ABC"
}
```

409

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
content-length: 99
content-security-policy: default-src 'self'; connect-src *.uds-qa.lenovo.com;
style-src 'self' 'unsafe-inline'; img-src 'self' data:; script-
src 'self' 'unsafe-inline'; object-src 'none';
content-type: application/json
date: Wed, 29 Sep 2021 10:31:15 GMT
```

```
expires: 0
pragma: no-cache
referrer-policy: no-referrer
server: Lenovo
strict-transport-security: max-age=31536000; includeSubDomains
x-content-type-options: nosniff
x-envoy-upstream-service-time: 54
x-frame-options: DENY
x-xss-protection: 1; mode=block
{"messages":["User with login id ccretoiu@lenovo.com already exist in organization autoticketing"]}
```

Email

```
GET https://api.uds-qa.lenovo.com/ldi/api/v1/users?freeText=sdragos.02879781611736808%40lenovo.com
```

```
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpzZW50Lm15IiwiaWF0IjoiYXZlbnV1ZG1iME41RzU0Nk1CX0RBOFJUOW...(truncated text)... SMHqjNONc5SeugT5C4dINKENr0Mlh933i6qw
```

```
200
cache-control: no-cache, no-store, max-age=0, must-revalidate
content-length: 1269
content-security-policy: default-src 'self'; connect-src *.uds-qa.lenovo.com; style-src 'self' 'unsafe-inline'; img-src 'self' data:; script-src 'self' 'unsafe-inline'; object-src 'none';
content-type: application/json
date: Wed, 29 Sep 2021 10:31:18 GMT
expires: 0
pragma: no-cache
referrer-policy: no-referrer
server: Lenovo
strict-transport-security: max-age=31536000; includeSubDomains
x-content-type-options: nosniff
x-envoy-upstream-service-time: 183
x-frame-options: DENY
x-xss-protection: 1; mode=block
```

```
{"_embedded":{"userList":[{"userId":"555fcf91-9dca-4fc6-9df5-...(truncated text)...number":0,"size":1,"totalElements":1,"totalPages":1},"responseType":"PAGE"}
```

1.8.3 Delete User

```
PATCH https://api.uds-qa.lenovo.com/ldi/api/v1/users
```

```
Authorization: Bearer
```

```
eyJhbGciOiJIJSUzI1NiIsInR5cCI6IkpzZW50Lm54IiwiaWF0IjoiYXZlICJwZG1iME41RzU0Nk1CX0RBOFJUOW...(truncated text)... KBLewSCisibDjTmr8Rnt4w
```

```
Content-Type: application/json; charset=utf-8
```

```
{
  "userList": [
    "555fcf91-9dca-4fc6-9df5-4d9f38841e2d"
  ],
  "operation": "DELETE"
}
```

```
200
```

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
```

```
content-length: 95
```

```
content-security-policy: default-src 'self'; connect-src *.uds-qa.lenovo.com;
style-src 'self' 'unsafe-inline'; img-src 'self' data:; script-
src 'self' 'unsafe-inline'; object-src 'none';
```

```
content-type: application/json
```

```
date: Wed, 29 Sep 2021 10:31:20 GMT
```

```
expires: 0
```

```
pragma: no-cache
```

```
referrer-policy: no-referrer
```

```
server: Lenovo
```

```
strict-transport-security: max-age=31536000; includeSubDomains
```

```
x-content-type-options: nosniff
```

```
x-envoy-upstream-service-time: 601
```

```
x-frame-options: DENY
```

```
x-xss-protection: 1; mode=block
```

```
{"operationSuccessfulUsers":["555fcf91-9dca-4fc6-9df5-4d9f38841e2d"],"operationFailedUsers":[]}
```

1.9 Devices

1.9.1 ACME Client Code

Acme Groovy Code

```
package com.acme.lidi.test
import org.junit.jupiter.api.Test

class DevicesTests extends BaseLdiTestClass {
    @Test
    void getDevices() {
        var response =
client.authenticatedRestConnector.GET("/ldi/api/v1/devices/")
        assert response.code == 200
        var body = response.asMap()
        assert body.keySet() ==
['content', 'pageable', 'last', 'totalElements', 'totalPages', 'sort', 'first'
, 'number', 'numberOfElements', 'size', 'empty'] as Set
        assert body['content'] instanceof List
    }
    @Test
    void export() {
        var deviceId =
client.authenticatedRestConnector.GET("/ldi/api/v1/devices/").asMap()['content
'][0]['deviceId']
        var response =
client.authenticatedRestConnector.POST("/ldi/api/v1/devices/bulk/export",
            [ids: [deviceId]])
        assert response.code == 200
        var file = response.asFile()
        assert file.name =~ /\.*.csv/
        assert file.text.split(/\v/)[0] == 'DEVICE NAME,MACHINE TYPE,SERIAL
NUMBER,GROUP'
    }
}
```

1.9.2 HTTP Request Responses

Get Devices

Request

```
GET https://api.uds-qa.lenovo.com/ldi/api/v1/devices/
```



```
Authorization: Bearer
eyJhbGciOiJIJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJwZG1iME41RzU0Nk1CX0RBOFJUOW
...(truncated text)...N0LFHsvE7Q09QXeGzoU0IS86P1DF16BhEEcXzN5Pow
```

200

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
content-security-policy: default-src 'self'; connect-src *.uds-qa.lenovo.com;
style-src 'self' 'unsafe-inline'; img-src 'self' data:; script-
src 'self' 'unsafe-inline'; object-src 'none';
content-type: application/json
date: Tue, 28 Sep 2021 15:37:56 GMT
expires: 0
pragma: no-cache
referrer-policy: no-referrer
server: Lenovo
strict-transport-security: max-age=31536000; includeSubDomains
transfer-encoding: chunked
x-content-type-options: nosniff
x-envoy-upstream-service-time: 543
x-frame-options: DENY
x-xss-protection: 1; mode=block
{"content":[{"orgDeviceId":"6112d2990e9e6a202b99effc","deviceId":""},"orgId":
...(truncated
text)...{"sorted":true,"unsorted":false,"empty":false},"number":0,"first":true,"
numberOfElements":20,"size":20,"empty":false}
```

1.10 Fleet Management

ACME Code

Fleet Status

```
package com.acme.ldi.test
import org.junit.jupiter.api.Test
class FleetManagement extends BaseLdiTestClass {
    @Test
    void fleetStatus() {
        var response =
client.authenticatedRestConnector.GET("/ldi/api/v1/fleethealth")
        assert response.asMap().keySet() ==
["latestJobRuntime", "timestamp", "fleetHealthScore", "fleet
```

```
StorageScore", "fleetBatteryScore", "fleetWdmScore", "fleetPerformanceScore"]
as Set
    }
}
```

1.11 Insights Tests

```
package com.acme.lidi.test

import org.junit.jupiter.api.Test

class InsightsTests extends BaseLdiTestClass {
    @Test
    void insightsTest() {
        var devices =
client.authenticatedRestConnector.GET('/ldi/api/v1/devices').asMap().content
        var deviceId = devices[0].deviceId
        var response =
client.authenticatedRestConnector.POST("/ldi/api/v1/devices-
insights/$deviceId/issues/filter", [:])
        var body = response.asMap()
        assert body.keySet() ==
['content', 'pageable', 'last', 'totalElements', 'totalPages', 'sort', 'first'
, 'number', 'numberOfElements', 'size', 'empty'] as Set
        assert body['size'] == 20
    }
}
```

1.11.1 Request

```
GET https://api.uds-qa.lenovo.com/ldi/api/v1/fleethealth
```

```
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWZG1iME41RzU0Nk1CX0RBOFJUOW
...(truncated
text)...SWWahdAVE1l0wYgQRmbPNnDEoAq_ajmCyTPDb3SRR2C1JqIjF0za2Yr796vj5xgoyycLOLl
4ydadQ
```

Response

```
200
cache-control: no-cache, no-store, max-age=0, must-revalidate
content-length: 223
content-type: application/json
```

```
date: Wed, 29 Sep 2021 10:30:58 GMT
expires: 0
pragma: no-cache
server: Lenovo
strict-transport-security: max-age=31536000 ; includeSubDomains
x-content-type-options: nosniff
x-envoy-upstream-service-time: 314
x-frame-options: DENY
x-xss-protection: 1; mode=block

{"latestJobRuntime":"2021-09-24T07:49:42.196","timestamp":"2021-09-10T05:06:01.585638","fleetHealthScore":98,"fleetBsodScore":100,"fleetStorageScore":91,"fleetBatteryScore":94,"fleetWdmScore":99,"fleetPerformanceScore":100}
```

1.12 Issues Filter

Issues Tests

```
package com.acme.ldi.test
import org.junit.jupiter.api.Test
class IssuesFilter extends BaseLdiTestClass {
    @Test
    void filterIssues() {
        var response =
client.authenticatedRestConnector.POST('/ldi/api/v1/issues/filter')
        assert response.code == 200
        var body = response.asMap()
        assert body.keySet() ==
['content', 'pageable', 'last', 'totalElements', 'totalPages', 'sort', 'number',
', 'first', 'numberOfElements', 'size', 'empty'] as Set
        assert body['pageable']['pageNumber'] == 0
        assert body['pageable']['pageSize'] == 20
        assert body['content'] instanceof List
        println "Found ${body['content'].size()} issues."
    }
    @Test
    void markIssueAsResolved() {
        var response =
client.authenticatedRestConnector.POST('/ldi/api/v1/issues/mark-as-resolved',
["issuesUuids": ["fake-issue-uuid"],
```

```

        "comment"    : "Would like to resolve an issue that does NOT
exist."
    ])
    assert response.code == 404
}
}

```

Request

POST <https://api.uds-qa.lenovo.com/ldi/api/v1/issues/filter>

Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJwZG1iME41RzU0Nk1CX0RBOFJUOW
...(truncated text)...
bzhfSJCvsw8pTti6yVbvHLJVaw1eNkqVkvXJ2DXwDMoAyibXc70CUMLX0JfH2fU9tfERhYWwd7A

Response

```

200
cache-control: no-cache, no-store, max-age=0, must-revalidate
content-type: application/json
date: Wed, 29 Sep 2021 10:31:07 GMT
expires: 0
pragma: no-cache
server: Lenovo
strict-transport-security: max-age=31536000 ; includeSubDomains
transfer-encoding: chunked
x-content-type-options: nosniff
x-envoy-upstream-service-time: 2066
x-frame-options: DENY
x-xss-protection: 1; mode=block
{"content":[{"bucketId":"app_performance_impact","category":"Excel.exe","code":null...(truncated text)...
2,"first":true,"numberOfElements":20,"size":20,"empty":false}

```

1.13 Mark Issue as Resolved

```

@Test
void markIssueAsResolved() {
    var response =
client.authenticatedRestConnector.POST('/ldi/api/v1/issues/mark-as-resolved',
    ["issuesUuids": ["fake-issue-uuid"],
    "comment"    : "Would like to resolve an issue that does NOT
exist."

```

```
    ])  
    assert response.code == 404  
}
```

Request

```
POST https://api.uds-qa.lenovo.com/ldi/api/v1/issues/mark-as-resolved  
Authorization: Bearer  
eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJwZG1iME41RzU0Nk1CX0RBOFJUOW  
...(truncated text)...  
_ZD3p_pmiikTMzOJyQQ64CDSxE7DmRjS_zirs0XAnDQ5nIm716XVxn9bqgCriHNoqSERg8CyRWJixL  
BRPIe1P5K6Zd184A  
Content-Type: application/json; charset=utf-8  
{  
  "issuesUids": [  
    "fake-issue-uuid"  
  ],  
  "comment": "Would like to resolve an issue that does NOT exist."  
}  
400  
cache-control: no-cache, no-store, max-age=0, must-revalidate  
content-length: 1355  
content-type: application/json  
date: Wed, 29 Sep 2021 10:31:03 GMT  
expires: 0  
pragma: no-cache  
server: Lenovo  
strict-transport-security: max-age=31536000 ; includeSubDomains  
x-content-type-options: nosniff  
x-envoy-upstream-service-time: 71  
x-frame-options: DENY  
x-xss-protection: 1; mode=block  
{  
  "timestamp": "2021-09-29T10:31:03.761+00:00",  
  "status": 400,  
  "error": "Bad Request",  
  "message": "400 BAD_REQUEST \\\"JSON parse error: Cannot deserialize value of type `java.util.UUID` from String `fake-issue-uuid`: UUID has to be represented by standard 36-char representation; nested exception is com.fasterxml.jackson.databind.exc.InvalidFormatException: Cannot deserialize value of type `java.util.UUID` from String `fake-issue-uuid`: UUID has to be represented by standard 36-char representation\\n at [Source:"
```

```
(PushbackInputStream); line: 3, column: 9] (through reference chain:
com.lenovo.iss.graphql.rest.request.MarkAsResolvedRequest[\"issuesUids\"]-
>java.util.HashSet[0])\""}, "path": "/iss-insights-api/api/issues/mark-as-
resolved", "errors": {"defaultMessage": "400 BAD_REQUEST \\\"JSON parse error:
Cannot deserialize value of type `java.util.UUID` from String \\\"fake-issue-
uuid\\\": UUID has to be represented by standard 36-char representation; nested
exception is com.fasterxml.jackson.databind.exc.InvalidFormatException: Cannot
deserialize value of type `java.util.UUID` from String \\\"fake-issue-uuid\\\":
UUID has to be represented by standard 36-char representation\\n at [Source:
(PushbackInputStream); line: 3, column: 9] (through reference chain:
com.lenovo.iss.graphql.rest.request.MarkAsResolvedRequest[\"issuesUids\"]-
>java.util.HashSet[0])\""}}}
```

Response

```
Assertion failed:
  assert response.code == 404
    |           |      |
    |           400  false
    com.lenovo.ldi.util.RestResponse@f6497e3
  at
org.codehaus.groovy.runtime.InvokerHelper.assertFailed(InvokerHelper.java:436)
  at
org.codehaus.groovy.runtime.ScriptBytecodeAdapter.assertFailed(ScriptBytecodeA
dapter.java:670)
  at
com.acme.ldi.test.IssuesFilter.markIssueAsResolved(IssuesFilter.groovy:25)
  at java.base/java.lang.Thread.run(Thread.java:834)
```

1.14 Sensors

```
package com.acme.ldi.test
import org.junit.jupiter.api.Test
class SensorsTest extends BaseLdiTestClass {
    @Test
    void getDefinedSensors() {
        // Test not functional, it's expected to fail
        var response =
client.authenticatedRestConnector.GET("/api/v1/ldiplus/definedsensors?descript
ionApp=1")
        assert response.code == 200 // Test not implemented
    }
}
```

```
@Test
void getSensorActions() {
    // Test not functional, it's expected to fail
    var response =
client.authenticatedRestConnector.GET("/api/v1/ldiplus/sensoractions?descripti
onApp=1")
    assert response.code == 200 // Test not implemented
}
}
```