



Cisco Unity Connection APIs -- Troubleshooting

[Cisco Unity Connection APIs -- Troubleshooting](#) 2

[Troubleshooting Connection APIs \(Applies to All Connection APIs\)](#) 2

[GET Example](#) 2

[PUT Example](#) 3

[POST Example](#) 4

[DELETE Example](#) 4

[Example of Making a Call](#) 5

Revised: December 28, 2018,

Cisco Unity Connection APIs -- Troubleshooting

Links to Other API pages: [Cisco_Unity_Connection_APIS](#)

Troubleshooting Connection APIs (Applies to All Connection APIs)

The VMREST micro traces provide diagnostic information on the use of the API. When troubleshooting problems, enable all levels of the VMREST micro trace. Since the API is built upon the same foundation as the administrative interface, certain levels of the CUCA micro trace might also provide additional information for a particular problem.

The VMREST micro traces are written to the Tomcat diagnostic files. These files can be identified by their name, which starts with `diag_Tomcat`. If root access is available and you would rather not use RTMT to retrieve the logs, they can be found in `/var/opt/cisco/connection/log`.

Much of the API code is generated at build time. Usually if the class name begins with the word 'Generated', then it is generated code. For example, 'com.cisco.connection.rest.impl.GeneratedUserRestImpl' is generated.

All examples are taken from internal build 8.5.0.198.

GET Example

Below are the diagnostics that the VMREST micro traces generate on a "GET /vmrest/users" request. These diagnostics are annotated, and annotations begin with a # and applies to the diagnostic line after the annotation.

This result code tells us if authentication was successful or not, and alerts us of things such as the password needs to be changed.

```
09:26:53.473 \1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.SecurityFilter - IMS result code: 0
```

This is the actual request that came over and is being processed.

```
09:26:53.629 \1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.RequestFilter - REQUEST GET users/
```

In this case the request was very simple, and had no rows per page, page number, query, or sorting parameters.

```
09:26:53.660 \1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - setting rows per page to default: 20001
```

```
09:26:53.660 \1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - setting page number to default: 1
```

```
09:26:53.660 \1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - query: null
```

```
09:26:53.660 \1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - parse sort: null
```

Information on who is actually being, or has been, authenticated.

```
09:26:53.663 \1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - user object ID retrieved from security context: objectid=257f39e9-cf29-4e37-b8bf-d74e479e9f8a
```

```
09:26:53.664 \1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - created authentication information: username=CCMAdministrator, alias=CCMAdministrator, id=257f39e9-cf29-4e37-b8bf-d74e479e9f8a
```

```
09:26:53.925 \1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - returning session cached authentication information: alias=CCMAdministrator, id=257f39e9-cf29-4e37-b8bf-d74e479e9f8a
```

```

# Date and time data from and to the database have to go through a transformation to be in the correct format.
09:26:53.972 \|1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.helper.Utilities - databaseDateTime passed in
as 2010-11-05 22:44:28.425
09:26:53.972 \|1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.helper.Utilities - Replacement results in
2010-11-05T22:44:28Z

# In this case, these fields not existing in the transfer object are benign. The transfer object holds all the various pieces of data for the
object in question, and in this case, the transfer object being used does not have a few of the fields that the generated code is expecting.
These fields would be there if you were to do a GET on a specific user since the transfer object used for retrieving a specific user is
different than the one used for retrieving a list of users.
09:26:53.975 \|1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - skipping field
that does not exist in transfer object: field=exitTargetConversation
09:26:53.975 \|1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - skipping field
that does not exist in transfer object: field=exitAction
09:26:53.976 \|1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - skipping field
that does not exist in transfer object: field=exitTargetHandlerObjectId

# The generated code can be extended through extension classes. This is a call to the afterGet method of the extension.
09:26:53.984 \|1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - Calling
restImplExtension.afterGet
...
# Found 13 users in total.
09:26:54.070 \|1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - found user:
count=13

# This is simply denoting that the response filter has been called. Currently, the only time this method does any actual work is when
dealing with JSON...in this case we are using XML.
09:26:54.088 \|1685,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.ResponseFilter - RESPONSE filter IN

```

PUT Example

Below are the diagnostics that the VMREST micro traces generate on a "PUT /vmrest/users/<objectId>" request where the DisplayName and LastName of the User are being changed. These diagnostics are very similar to the GET example.

```

10:49:06.410 \|28433,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.SecurityFilter - IMS result code: 0
10:49:06.540 \|28433,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.RequestFilter - REQUEST PUT
users/58a3839b-0827-4cde-9acd-ce33ae64bc27
10:49:07.015 \|28433,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - update user:
ObjectId=58a3839b-0827-4cde-9acd-ce33ae64bc27
10:49:07.166 \|28433,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - user object ID
retrieved from security context: objectId=257f39e9-cf29-4e37-b8bf-d74e479e9f8a
10:49:07.167 \|28433,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - created
authentication information: username=CCMAAdministrator, alias=CCMAAdministrator, id=257f39e9-cf29-4e37-b8bf-d74e479e9f8a
10:49:07.167 \|28433,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - returning
session cached authentication information: alias=CCMAAdministrator, id=257f39e9-cf29-4e37-b8bf-d74e479e9f8a

```

10:49:07.172 \28433,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - Calling restImplExtension.beforeModify

10:49:07.989 \28433,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.ResponseFilter - RESPONSE filter IN

POST Example

This is an example of creating a new user with an Alias of NewUser and a DtmfAccessId of 4001. The only new diagnostic here is the one printing out the subscriber transfer object, which contains the fields that will be set on the new user. The other fields that are not set on the transfer object will be set according to the template provided, which in this case was voicemailuseremplate.

08:34:13.486 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.SecurityFilter - IMS result code: 0

08:34:13.487 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.RequestFilter - REQUEST POST users

08:34:13.983 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - user object ID retrieved from security context: objectid=f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad

08:34:13.983 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - created authentication information: username=CCAdministrator, alias=CCAdministrator, id=f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad

08:34:14.048 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - returning session cached authentication information: alias=CCAdministrator, id=f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad

08:34:14.052 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - Calling restImplExtension.beforeCreate

This is a dump of the transfer object that will be used to create the new user.

08:34:15.368 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - created subscriber:

08:34:15.368 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection

DELETE Example

These diagnostics are from deleting the same user that was created in the POST example above.

08:44:14.795 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.SecurityFilter - IMS result code: 0

08:44:14.796 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.RequestFilter - REQUEST DELETE users/22a27b16-752a-4b14-adbf-3ccf3aa4c9d9

08:44:14.800 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - user object ID retrieved from security context: objectid=f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad

08:44:14.800 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - created authentication information: username=CCAdministrator, alias=CCAdministrator, id=f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad

08:44:14.800 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - returning session cached authentication information: alias=CCAdministrator, id=f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad

08:44:14.934 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - returning session cached authentication information: alias=CCAdministrator, id=f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad

08:44:14.934 \7372,,,VMREST,3,DEBUG \[http-8443-2\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - Calling restImplExtension.beforeDelete

Here a check is being made to see if the user is imported (determined by a non-null ccmid and type). If the user is imported, then the user would not be deleted and a 405 response would be generated. In this case, the user is not imported, and so it can be deleted.

08:44:14.960 \|7372,,,VMREST,3,DEBUG \| [http-8443-2\] com.cisco.connection.rest.helper.DataHelper - executeQuery: query='select CcmId,CcmIdType from vw_user where objectid=?'

08:44:14.960 \|7372,,,VMREST,3,DEBUG \| [http-8443-2\] com.cisco.connection.rest.helper.DataHelper - arg='22a27b16-752a-4b14-adbf-3ccf3aa4c9d9'

08:44:14.965 \|7372,,,VMREST,3,DEBUG \| [http-8443-2\] com.cisco.connection.rest.helper.DataHelper - getValuesFromQuery: ccmid=null

08:44:14.965 \|7372,,,VMREST,3,DEBUG \| [http-8443-2\] com.cisco.connection.rest.helper.DataHelper - getValuesFromQuery: ccmidtype=null

08:44:15.641 \|7372,,,VMREST,3,DEBUG \| [http-8443-2\] com.cisco.connection.rest.ResponseFilter - RESPONSE filter IN

Example of Making a Call

This is an example of making a phone call through CUTI. In this case, a call is being established to the extension 1017.

08:56:48.698 \|20660,,,VMREST,3,DEBUG \| [http-8443-5\] com.cisco.connection.rest.SecurityFilter - IMS result code: 0

08:56:48.699 \|20660,,,VMREST,3,DEBUG \| [http-8443-5\] com.cisco.connection.rest.RequestFilter - REQUEST POST calls

08:56:48.886 \|20660,,,VMREST,3,DEBUG \| [http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - user object ID retrieved from security context: objectid=f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad

08:56:48.886 \|20660,,,VMREST,3,DEBUG \| [http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - created authentication information: username=CCMAdministrator, alias=CCMAdministrator, id=f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad

08:56:48.886 \|20660,,,VMREST,3,DEBUG \| [http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - returning signed in user id: f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad

08:56:48.886 \|20660,,,VMREST,3,DEBUG \| [http-8443-5\] com.cisco.connection.rest.helper.AuthorizationHelper - searching for assigned roles: data association = com.cisco.unity.datatools.DataAssociation@15b110e

08:56:48.891 \|20660,,,VMREST,3,DEBUG \| [http-8443-5\] com.cisco.connection.rest.helper.AuthorizationHelper - user has role: user id=f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad, role name=System Administrator

System administrators are not limited by a class of service.

08:56:48.891 \|20660,,,VMREST,3,DEBUG \| [http-8443-5\] com.cisco.connection.rest.impl.callcontrol.CallControlRestImpl - do not enforce restriction table for administrators

08:56:48.891 \|20660,,,VMREST,3,DEBUG \| [http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - returning session cached authentication information: alias=CCMAdministrator, id=f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad

08:56:48.891 \|20660,,,VMREST,3,DEBUG \| [http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - returning signed in user id: f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad

An attempt is made to find the media switch for the user, which will get used to make the call from. In this case the user does not have a media switch assigned, so the fallback is the default. This is perfectly fine and an expected case in many situations (it should not be an error, that will be changed to a DEBUG level diagnostic in the future).

08:56:48.923 \|20660,,,VMREST,3,ERROR \| [http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - subscriber not found: id=f8a8e0b3-4366-4bb6-b25b-1d39e1ae58ad

08:56:48.933 \|20660,,,VMREST,3,DEBUG \| [http-8443-5\] com.cisco.connection.rest.impl.GeneratedUserRestImpl - returning media switch from default trap switch: 2120b8e0-638e-4734-9dfa-659b2a4433f3

These two lines note which call id is mapped to which trap connection id. The call id is how the user of the API will reference this particular call, while the trap connection id is what is used internally by the trap component to keep track of the call.

08:56:53.076 \20660,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.callcontrol.CallControlRestImpl - native trap connection id: 1

08:56:53.406 \20660,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.impl.callcontrol.CallControlRestImpl - mapped call id to trap connection: call id=5, trap connection id: 1

08:56:53.407 \20660,,,VMREST,3,DEBUG \[http-8443-5\] com.cisco.connection.rest.ResponseFilter - RESPONSE filter IN



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA 95134-1706
USA

Asia Pacific Headquarters
CiscoSystems(USA)Pte.Ltd.
Singapore

Europe Headquarters
CiscoSystemsInternationalBV
Amsterdam,TheNetherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.