Cisco Unity Connection Imaging Interface (CUII) API -- Error Handling

Links to Other API pages: Cisco_Unity_Connection_APIs

Cisco Unity Connection Imaging Interface (CUII) API -- Error Handling

Links to Other API pages: Cisco_Unity_Connection_APIs

Possible Errors

The HTTP status codes themselves provide information about many typical errors. See the following list for some of the status codes returned by CUII:

Possible Errors	Description	Cause	Troubleshooting
401	This means there has been an error in authentication and the response text will have a "Not Authorized" message.	The cause of this error can be that the user credentials provided in the request are not authorized to view the information requested in the API.	As the error clearly specifies the user does not have necessary permissions to access the information, so either provide admin credentials or admin should assign the user the necessary role.
503	This means "Server Busy". It signifies that the connection server is busy and cannot handle this request at that time.	The cause of this error can be that the connection server is processing multiple simultaneous API requests so it is not able to take up that request at that particular time.	This is actually not a problem. The throttling is implemented to save connection's tomcat server from taking up multiple requests which can cause issues in memory, DB etc.

Possible Errors	Description	Cause	Troubleshooting
400	This means "Bad request". The error signifies that the data provided in the request is not valid.	The causes of this error can be that the input data which is required for processing the API request is either incomplete or incorrect.	This error would contain the details of the missing or incorrect or error data in the response itself as described above but for further debugging the traces can be analyzed.
500	This means "Internal Server Error". The error signifies that there has been an error while processing the request and this error is not handled by the application.	The causes of this error can be multiple. Generally those are the scenarios which should have been handled in the application but not done.	To troubleshoot this type of error, one has to look into the stack trace returned in the response, which will clearly specify the exception. For further debugging, one must look into the traces for errors as the errors are quite prominently logged with a lot of details.