GLM RESTful APIs

Version 1.0

Contents

[Authentication 1](#_Toc401049233)

[RESTful APIs 2](#_Toc401049234)

[1. Search 2](#_Toc401049235)

[2. Document 3](#_Toc401049236)

[3. Transactions 5](#_Toc401049237)

# Authentication

This authentication is currently not implemented. However, the authentication will be as described when implemented.

API will use Basic Authentication. Each request will have an encoded username/password combination. The server will check the username/password of each request so that only the requests with valid user accounts are processed.

The user accounts can also be useful if metering is needed. A metering capability can be added to control the API request frequency.

Each API request will need to include a username and password that is base64 encrypted and placed in the request header using the following name-value pair entry.

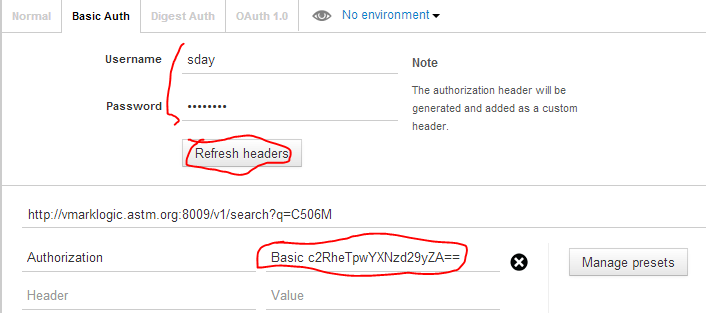
|  |  |
| --- | --- |
| **Name** | **Value** |
| Authorization | Basic c2RheTpwYXNzd29yZA== |

The username/password in the above example is in the encrypted string after the word “Basic”.

In this case, the encrypted string is decrypted to be: sday:password

I recommend using the Google Chrome POSTMAN tool to generate the header as shown in the following image.

**Image 1: POSTMAN Basic Auth Header Generator**



# RESTful APIs

## Search

End Point Info:

|  |  |
| --- | --- |
| End Point | * Port 8030 * /v1/resources/search * http://glm-ml-dev1.amers2.ciscloud:8030/v1/resources/search |
| Request Types | GET |

GET Request Parameters:

|  |  |
| --- | --- |
| **Parameter** | **Notes** |
| rs:start | Page number. Default is 1. Used in combination with rs:pageLength parameter. |
| rs:pageLength | Default is 10. |
| rs:q | search string that supports rich search grammar, such as AND, OR, Exclude (-), NEAR, GE, LE, etc. |
| rs:format | xml or json |

Payload: None

Example Requests:

| **#** | **GET Request URL** |
| --- | --- |
| 1 | [http://glm-ml-dev1.amers2.ciscloud:8030/v1/resources/search?rs:pageLength=30&rs:start=1&rs:q=RU0\*99](http://glm-ml-dev1.amers2.ciscloud:8030/v1/resources/search?rs:pageLength=30&rs:start=1&rs:q=RU0*99) |
| 2 | [http://glm-ml-dev1.amers2.ciscloud:8030/v1/resources/search?rs:pageLength=30&rs:start=1&rs:q=importedUnitCode:RU0\*99](http://glm-ml-dev1.amers2.ciscloud:8030/v1/resources/search?rs:pageLength=30&rs:start=1&rs:q=importedUnitCode:RU0*99) |
| 3 | [http://glm-ml-dev1.amers2.ciscloud:8030/v1/resources/search?rs:pageLength=30&rs:start=1&rs:q=importedUnitCode:RU0\*199](http://glm-ml-dev1.amers2.ciscloud:8030/v1/resources/search?rs:pageLength=30&rs:start=1&rs:q=importedUnitCode:RU0*199) |

## Document

End Point Info:

|  |  |
| --- | --- |
| End Point | * Port 8030 * /v1/resources/document * http://glm-ml-dev1.amers2.ciscloud:8030/v1/resources/document |
| Request Types | GET, PUT, POST, DELETE |

Request Parameters:

|  |  |
| --- | --- |
| **Operation** | **Notes** |
| GET | Parameter: rs:uri  Parameter: rs:txid – transaction-id (optional) |
| PUT | Parameter: rs:uri – document identifier  Parameter: rs:txid – transaction-id (optional)  Payload: well-formed XML doc replaces the existing document |
| POST | Parameter: rs:uri  Parameter: rs:txid – transaction-id (optional)  Payload: well-formed XML doc replaces the existing document |
| DELETE | Parameter: rs:uri  Parameter: rs:txid – transaction-id (optional) |

Recommended Payload for PUT and POST Operations:

Payload Example 1:

<inventory xmlns="http://tax.thomsonreuters.com">  
 <meta>  
 <id>1235</id>  
 <type>inventory</type>  
 <salesItem>Bongos</salesItem>  
 <createdAt>2014-10-07T17:14:43.120574-05:00</createdAt>  
 <updatedAt>2014-10-07T17:14:43.120574-05:00</updatedAt>  
 </meta>  
 <feed>  
 <item>  
 <name>Bongo</name>  
 <count>24</count>  
 <price>35.0</price>  
 <version>1.3</version>  
 <notes>This bongo drum has been on sale since 1925.</notes>  
 </item>  
 <supplier>  
 <supplierName>Musical Instruments Inc</supplierName>  
 <address>  
 <street1>One Drum Street</street1>  
 <street2>Suite 22</street2>  
 <city>New York</city>  
 <state>NY</state>  
 <zip>10011-1111</zip>  
 </address>  
 <phones>  
 <phone type="primary">212-111-5555</phone>  
 <phone type="secondary">212-222-5555</phone>  
 <phone type="other">212-333-5555</phone>  
 </phones>  
 </supplier>  
 </feed>  
</inventory>

The root node is used to generate the first part of the document URI.

In this case the URI will look like: /inventory/80366382489486423323950579459703260252.xml

Payload Example 2:

<order xmlns="http://tax.thomsonreuters.com">  
 <meta>  
 <number>1</number>  
 <type>order</type>  
 <retailer>Amazon</retailer>  
 <createdAt>2014-10-06T14:27:30.170996-05:00</createdAt>  
 <updatedAt>2014-10-06T14:27:30.170996-05:00</updatedAt>  
 </meta>  
 <feed>  
 <customer>  
 <id>2</id>  
 <firstName>Rex</firstName>  
 <lastName>Shepard</lastName>  
 <email>rex@dogsrus.com</email>  
 <street>1 Milk Bone Lane</street>  
 <city>Anytown</city>  
 <postalCode>12345</postalCode>  
 </customer>  
 <lineItems>  
 <lineItem>  
 <item>artichoke</item>  
 <quantity>3</quantity>  
 <unitPrice>0.50</unitPrice>  
 <total>1.50</total>  
 </lineItem>  
 <lineItem>  
 <item>bongo</item>  
 <quantity>1</quantity>  
 <unitPrice>25.00</unitPrice>  
 <total>25.00</total>  
 </lineItem>  
 <lineItem>  
 <item>airplane</item>  
 <quantity>1</quantity>  
 <unitPrice>749999.99</unitPrice>  
 <total>749999.99</total>  
 </lineItem>  
 </lineItems>  
 <notes>Customer is a talking dog.</notes>  
 <total>750026.49</total>  
 </feed>  
</order>

As before, the root node is used to generate the first part of the document URI.

In this case the URI will look like: /order/71389246786399328231559708831309729441.xml

## Transactions

End Point Info:

|  |  |
| --- | --- |
| End Point | * Port 8030 * /v1/resources/transactions * http://glm-ml-dev1.amers2.ciscloud:8030/v1/resources/transactions |
| Request Types | GET, POST, DELETE |

Parameters:

| **Operation** | **Notes** |
| --- | --- |
| GET | No parameters  Returns Transaction Status of all transactions in the system at the time of the request. |
| POST | Parameter: rs:txid – transaction-id  Parameter: rs:result=commit or rs:result=rollback  Payload: none |
| DELETE | Parameter: rs:txid  DELETE is not yet implemented |