A simple search portal for both the template (workbook) and user data is available here. => <http://glm-ml-dev.amers1.cis.trcloud:8010/>

There are 2 REST API Endpoints:

* /template – Workpaper Template API
* /wpaper – Workpaper User Data API

**Template API Tests**

| # | Type | Action |
| --- | --- | --- |
| 1 | POST | Uploads a new template file that physically resides on the local client directory. |
| 2 | PUT | Updates (overwrites) an existing Template file. |
| 3 | GET File | Returns specific template spreadsheet file. |
| 4 | GET - List | Returns the list of IDs and URIs of all Template Excel files. Pagination parameters are not yet implemented. |
| 5 | GET - Search | Searches templates using terms especially Ontology terms. Returns list of templates with search relevance scores. |
| 6 | DELETE | Deletes the Template file and its metadata. |

**Data Request API Tests**

| # | Type | Action |
| --- | --- | --- |
| 1 | POST | Creates a new User Data file. |
| 2 | PUT | Updates (overwrites) an existing Template file. |
| 3 | GET Merged File | Return merged spreadsheet file. Returns binary xlsx file. |
| 4 | GET Specific User Doc | Return User Data File. Returns specific user data file. |
| 5 | GET - List | Returns the list of IDs and URIs of all User Data files. This will eventually be restricted by client, by user, and by role. |
| 6 | GET - Search | Searches user data using terms especially Ontology terms. Returns list of user data IDs with search relevance score. |
| 7 | DELETE | Deletes the User Data file. |

**Template API Test Sequence**

| # | Type | Parameters | Action |
| --- | --- | --- | --- |
| 1 | POST | **Parameters:**  rs:filename – required  **Examples:**  /template  /template?fs:filename=C590300 | Uploads a new template file that physically resides on the local client directory.  If no filename is provided then a filename is auto generated.  The payload is the binary Excel file.  Returns the template ID and URI.  **Curl Commands:**  In this case, the template file is indicated by “./C5903000.xlsx”.   1. POST Template File with desired filename.   curl -X POST --data-binary @./C5903000.xlsx \      --header "Content-Type:application/vnd.openxmlformats-officedocument.spreadsheetml.sheet" \  <http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/template?rs:filename=C5903000> |
| 2 | PUT | **Parameters:**  rs:id – template ID  **Example:**  /template?rs:id=7940446531123017941 | Updates (overwrites) an existing Template file.  The payload is the binary Excel file.  Be sure to appropriate template id (rs:id). If necessary, use the following GET APIs to find the appropriate id.  Returns the template ID and URI.  **Curl Commands:**   1. PUT Template File – update existing template file if one exists.   curl -X PUT --data-binary @./C5903000.xlsx \      --header "Content-Type:application/vnd.openxmlformats-officedocument.spreadsheetml.sheet" \  <http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/template?rs:id=7940446531123017941> |
| 3 | GET File | **Parameters:**  rs:id – Template ID  **Example:**  /template?rs:id=8216640002390149622 | Returns specific template spreadsheet file.  **Curl Commands:**   1. GET Binary by Template ID 8216640002390149622   curl --anyauth --user grusso:password -o templateC5903000.xlsx -X GET \    -H "Accept: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet" \    -r "0-511999" \  <http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/template?rs:id=8216640002390149622> |
| 4 | GET - List | **Parameters:**  rs:pg – page (not yet implemented)  rs:ps – page size  **Example:**  /template?rs:pg=2&rs:ps=10 | Returns the list of IDs and URIs of all Template Excel files. Pagination parameters are not yet implemented.  **Curl Commands:**   1. GET List of Templates using a search term:   BEPS AND juris:Brazil   curl --anyauth --user grusso:password -X GET \      -H "Accept: application/xml" \  <http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/template> |
| 5 | GET - Search | **Parameters:**  rs:q – Search Term  rs:pg – page (not yet implemented)  rs:ps – page size  **Example:**  /template?rs:q=BEPS AND juris:Brazil | Searches templates using terms especially Ontology terms.  Returns list of templates with search relevance scores.  **Curl Commands:**   1. GET List of Templates using a search term:   BEPS AND juris:Brazil   curl --anyauth --user grusso:password -X GET \      -H "Accept: application/xml" \      "[http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/template?rs:q=BEPS AND juris:Brazil](http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/template?rs:q=BEPS%20AND%20juris:Brazil)" |
| 6 | DELETE | **Parameters:**  rs:id – template Id  **Examples:**  /template?rs:id=7940446531123017941 | Deletes the Template file and its metadata.  **Curl Commands:**   1. DELETE Template File   curl --anyauth --user grusso:password -X DELETE \      -H "Accept: application/xml" \  <http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/template?rs:id=7940446531123017941> |

**Data Request API Test Sequence**

| # | Type | Parameters | Action |
| --- | --- | --- | --- |
| 1 | POST | **Parameters:**  rs:templateid – associates the json name/value pairs with the specific spreadsheet template file.  **Example:**  /wpaper?rs:templateId= 3857790183476763686  Input   * URI parameter: template Id * Payload: Name/value pairs | Creates a new User Data file.  The payload is the User Data file.  Returns the User Data File ID and maybe the URI.  JSON Format:  { "dnames": [   {"name":"I\_0001",      "value":"1001.25"},    {"name":"I\_0002",      "value":"1002.25"},    {"name":"BegMnA.block","value":"44442.98"},   {"name":"EndMnA.block","value":"17772.34"},   {"name":"EndMnA.block","value":"27772.34"} ] }  **Curl Commands:**   1. POST User Data   curl --anyauth --user grusso:password -X POST \    -H "Content-Type: application/json" \    -d '{"dnames":[{"name":"I\_0001","value":"1111.25"},{"name":"I\_0002","value":"2222.25"},  {"name":"EndMnA.block","value":"27772.34"}]}' \  <http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/wpaper?rs:templateId=3857790183476763686>   1. POST User Data File – Used above json but is stored in file called ./data1.txt   curl --anyauth --user grusso:password --data @./data1.txt -X POST \    -H "Content-Type: application/json" \  <http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/wpaper?rs:templateId=3857790183476763686> |
| 2 | PUT | **Parameters:**  rs:dataid – User Data Document Id  payload – Use same json document format as shown in the above POST operation.  **Example:**  /wpaper?rs:dataId= 1526690289281135790 | Updates (overwrites) an existing Template file.  The payload is the User Data file.  Use the same User Data format as described in the POST request description.  **Curl Commands:**   1. PUT User Data   curl --anyauth --user grusso:password -X PUT \    -H "Content-Type: application/json" \    -d '{"dnames":[{"name":"I\_0001","value":"1111.25"},{"name":"I\_0002","value":"2222.25"},  {"name":"EndMnA.block","value":"27772.34"}]}' \  <http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/wpaper?rs:dataid=8708286384647449580>   1. PUT User Data File – json is stored in file called ./data1.txt   curl --anyauth --user grusso:password --data @./data1.txt -X PUT \    -H "Content-Type: application/json" \  <http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/wpaper?rs:dataid=8708286384647449580> |
| 3 | GET Merged File | **Parameters:**  rs:id – User Data ID  **Example:**  /wpaper?rs:id=9806729890807106386 | Return merged spreadsheet file.  Returns binary xlsx file.  **Curl Commands:**   1. GET Merged Spreadsheet File   curl --anyauth --user grusso:password -o userWorkpaper994.xlsx -X GET \      -H "Accept: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet" \  <http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/wpaper?rs:id=9806729890807106386> |
| 4 | GET Specific User Doc | **Parameters:**  rs:id – User Data ID  rs:merge – set to false  **Example:**  /wpaper?rs:id=9806729890807106386&rs:merge=false | Return User Data File  Returns specific user data file.  **Curl Commands:**   1. GET User Data File Only   curl --anyauth --user grusso:password -X GET \      -H "Accept: application/xml" \      "<http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/wpaper?rs:id=9806729890807106386&rs:merge=false>" |
| 5 | GET - List | **Parameters:**  rs:pg – page (not yet implemented)  rs:ps – page size  **Example:**  /wpaper | Returns the list of IDs and URIs of all User Data files.  This will eventually be restricted by client, by user, and by role.  **Curl Commands:**   1. GET User Data File Only   curl --anyauth --user grusso:password -X GET \      -H "Accept: application/xml" \  <http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/wpaper> |
| 6 | GET - Search | **Parameters:**  rs:q – Search Term  rs:pg – page (not yet implemented)  rs:ps – page size  **Example:**  /wkpaper?rs:q=BEPS AND juris:Brazil | Searches user data using terms especially Ontology terms.  Returns list of user data IDs with search relevance score and snippet.  **Curl Commands:**   1. GET List of User Data documents using Search Term   curl --anyauth --user grusso:password -X GET \      -H "Accept: application/xml" \      "[http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/wpaper?rs:q=BEPS AND juris:Brazil](http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/wpaper?rs:q=BEPS%20AND%20juris:Brazil)" |
| 7 | DELETE | **Parameters:**  rs:dataid – User Data Document Id to be deleted  **Example:**  /wpaper?rs:id=9806729890807106386 | Deletes the User Data file.  **Curl Commands:**   1. DELETE User Data File   curl --anyauth --user grusso:password -X DELETE \      -H "Accept: application/xml" \  <http://glm-ml-dev.amers1.cis.trcloud:8010/v1/resources/wpaper?rs:id=9806729890807106386> |