LabCollector / LSM API v2

Summary

1. Available methods in LSM web service 3
1.1. GET 3
1.1.1. Jobs 3
1.1.2. Equipments 4
1.1.3. Assays 4
1.1.4. Categories 4
1.1.5. Sample Types 4
1.1.6. Parameters 4
1.1.7. Requesters 4
1.1.8. Invoices 5
1.1.9. Batches 5
1.1.10. Token 5
1.1.11. Send token by email 5
1.1.13. Get data trending 5
1.1.14. Models 5
1.1.15. Protocols 6
1.2. POST 7
1.2.1. Create requesters 7
1.2.2. Create job 7
1.2.2.1. Advanced options on job creation 8
1.2.3. Append samples to existent job 8
1.2.4. Save results 8
1.3. PUT 9
1.3.1. Upload files 9
1.3.2. Link sample to module 9
1.3.3. Update sample 9
1.3.4. Update sample status 9
1.3.5. Start a batch 10
1.3.6. Update job 10

1.3.7. Accept/Reject a job 11
1.3.8. Order a job quoted 11
1.3.9. Update requester 11
2. Authentication 11
3. Tool 11
4. Screenshots 12
4.1. GET 12
4.1.1. Get list of jobs in progress 12
4.1.2. Get list of samples received in a period 12
4.1.3. Get list of requesters 13
4.1.4. Get token 13
4.1.5. Send token by email 13
4.2. POST 14
4.2.1. Create jobs 14
4.2.3. Create requesters 15
4.3.PUT 15
4.3.1. Update sample using id 15
4.3.2. Update sample using name 16
4.3.3. Update sample status 16
4.3.4. Update requester 17

1. Available methods in LSM web service

1.1 GET

1.1.1 Jobs

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2
- Optional search parameters:
    - datefrom
    - dateto
    - jobid (id or barcode)
    - sampleid (id or barcode)
    - assayid
    - status (by default only job in progress are returned, use status=all for view all jobs)
    - samplename
    - samplestatus
    - samplestatusdatefrom
    - samplestatusdateto
    - assaystatus
    - assaystatusdatefrom
    - assaystatusdateto
    - startedon
    - completedon
    - validatedon
    - batch
    - jobstatus
    - jobnumber
    - requesterid
    - caserecord
    - caserecordname
    - caserecorddob
    - caserecordgender
    - projectcode

- includeqc (by default QC samples are excluded)
- source
- sourcereport
- parameteronreport (to retrieve only parameters OnReport)
- amended (to filter only amended jobs)

- Additional parameters:
  - parameterid & parametervalue to search by job parameter, sample parameter or assay parameter, supports multiple values
  - parameterdatatype: to retrieve only parameters of a given type
  - searchmode: when searching by jobnumber, samplename, caserecordname, batch or projectcode: searchmode indicates the mode (startswith, endswith, contains, strict-default-, notstartswith, notendswith, notcontains, inlist)
  - samplesearch: searchs by barcode (S+ID or Sample name), regarding the barcode option set in `lsm>admin>setup>job and samples naming`. The search is strict and supports multiple values separated by comma
  - limit: to limit number of results add limit=N, being N the max number of jobs
  - return: by default the tags returned are <job>, <sample>, <assay> and <parameter>. Omit tags or select other tags as sample report in base64 and requeter uses and managers is possible using the parameter return. The list of options for this parameter is return=job,sample,assay,parameter,samplereport,user,manager,jobreport,acceptance,invoice

1.1.2 Equipments

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&getEquipmentList=1
- Optional parameters: id

1.1.3 Assays

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&getAssayList=1
- Optional parameters: id (supports multiple ids separated by comma), search(name), code(supports multiple codes separated by comma)

1.1.4 Categories

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&getCategoryList=1
- Optional parameters: id

1.1.5 Sample Types

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&getSampleTypeList=1
- Optional parameters: id, search (name)

1.1.6 Parameters

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&getParameterList=1
- Optional parameters: id, level (level=J returns job parameters, level=S returns sample parameters and level=A returns test parameters)

1.1.7 Requesters

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&getRequesterList=1
- Optional parameters: id, categoryid

1.1.8 Invoices

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&getInvoiceList=1
- Optional parameters: id, datefrom, dateto, status, format (CSV | IIF), jobid, jobnumber

check also point 1.1.1: method getJobList with optional parameter return=invoice

1.1.9 Batches

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&getBatchList=1
- Optional parameters: batch, status (notfinished –default– | all | assigned | started | completed | approved)

1.1.10 Token

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&getToken
- Required parameters: jobid or sampleid

1.1.11 Send token by email

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&sendToken
- Required parameters: jobid or sampleid

1.1.12. Send results to external sources

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&sendResults
- Required parameters: jobid

Note: Send results back to external source only if the job have been created using our integration API (PWNHealth, 1Health, ORDRS, LabSoft)

1.1.13. Get data trending

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&getDataTrending
- Required parameters: viewid
- Optional parameters: datefrom, dateto (by default last month), separator (by default is ,)

Note: it returns a CSV with data trending data

1.1.14. Models

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&getModelList=1
- Optional parameters: id, search(name)

1.1.15. Protocols

- Method: GET
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2&getProtocolList=1
- Optional parameters: id, search(name), code(supports multiple codes separated by comma)

1.2 POST

1.2.1 Create requesters

- Method: POST
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2
- Required parameter: create
- Example of *create*:

```
   <requesters>
    <requester code='R01' name='Requester from API' taxNumber='12346789'>
        <billing address='One street' city='City' state='State' country='US'
zip='99999' phone='555123123' fax='555123123' contactFirstName='Jane'
contactLastName='Doe' contactEmail='jane@gmail.com' ></billing>
        <shipping address='One street' city='City' state='State' country='US'
zip='99999' phone='555123123' fax='555123123' contactFirstName='Jane'
contactLastName='Doe' contactEmail='jane@gmail.com' ></shipping>
    </requester>
</requesters>
```

1.2.2 Create job

- Method: POST
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2
- Required parameter: create
- Optional parameter: notoken (if is set option the token is not sent on job creation, although the automatic token is ON)
- Example of *create*:

```
<jobs>
      <job number='job-api-006' requester='21' project='109' priority='low'
caserecord='1'>
            <parameter id='242' value='text'></parameter>
            <sample name='job-api-006-sample1' type='1' comment='notes1'>
                  <parameter id='247' value='Female'></parameter>
                  <assay id='1'></assay>
                  <assay id='2'></assay>
            </sample>
            <sample name='job-api-006-sample2' type='1' comment='notes2'>
                  <parameter id='247' value='Male'></parameter>
                  <assay id='1'></assay>
                  <assay id='2'></assay>
            </sample>
      </job>
      <job number='job-api-007' requester='21' project="Test new project code">
            <parameter id='242' value='text'></parameter>
            <sample name='job-api-007-sample1' type='1' comment='notes1'>
                  <parameter id='247' value='Female'></parameter>
                  <assay id='1'></assay>
                  <assay id='2'></assay>
            </sample>
            <sample name='job-api-007-sample2' type='1' comment='notes2'>
```

```
                    <parameter id='247' value='Male'></parameter>
                    <assay id='1'></assay>
                    <assay id='2'></assay>
                </sample>
        </job>
</jobs>
```

Multiple jobs and multiple samples are allowed.
Attribute Requester is required, it supports id or text.
Attribute project is required only if Project option is enabled, it supports id or text
Attribute model is required only if Model option is enabled, it supports id or text, if a default model exists and model is not set then default mode is used
Attribute priority is optional, by default is Normal
Attribute caserecord is required only if Case record option is enabled
Attribute source is optional, by default is empty

1.2.2.1. Advanced options on job creation

Instead of add an attribute for requester and caserecord, add a tag with all fields, the system will search the match automatically, and if it does not exist a new record will be created. Indicate the fields to be used to match with an optional attribute keyfields. Examples:

```
<requester name='Agilebio' code='AB' registrationid='9999999999' taxnumber='12345'
keyfields='registrationid,name' />

<caserecord first_name='Jane' last_name='Doe' gender='Female' dob='2000-01-01'
phone='5551234' email='jane@aol.com' address='One Street, 5'
keyfields='first_name,last_name,dob,gender' />
```

The same for attached records. Example:

```
<insurance name='Medic' code'00001' city='NY' zip='555' address='X Street'
keyfields='name,city,zip' />
```

1.2.3 Append samples to existent job

Use the same notation of 1.2.2 Create job adding an extra parameter *appendsamplesifjobexist* = 1.

JobNumber must match and be unique. If the job doesn't exist a new job will be created. If job exists but is already completed it returns an error message.

1.2.4 Save results

- Method: POST
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2
- Required parameter: result
- Example of *result*:

```
<assays><assay uniqueid="1223"><result><parameter id="34" unit="mg"
value="123"/><parameter id="35" unit="ml"
value="456"/></result></assay></assays>
```

Note: results of completed or validated tests cannot be updated

1.3. PUT

1.3.1. Upload files

- Method: PUT
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2
- Parameters:
    - upload (required)
    - file (required)
    - uniqueid (required): identifier of job-sample-assay
    - parameter (optional): supports parameter id or parameter name
- Response: OK 200

1.3.2. Link sample to module

- Method: PUT
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2
- Parameters:
    - sampleid (required): identifier of sample (sampleid or barcode). Supports multiple ids separated by comma

    OR

    - samplename (required): sample name
    - module (required): module name or short name
    - recordid (required): identifier of record in module
- Response: OK 200

1.3.3. Update sample

- Method: PUT
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2
- Parameters:

- sampleid (required): identifier of sample (sample id or barcode). Supports multiple ids separated by comma

OR

- samplename (required): sample name. Supports multiple values separated by comma
- shipmentstatus(optional): Use 0 Submitted, 1 In transit, 2 Received, 3 Returned (supports number of label)
- name (optional): new sample name
- type (optional): new sample type
- batch (optional): batch to be assigned to tests
- Any sample parameter using parameter label or parameter id

1.3.4. Update sample status

- Method: PUT
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2
- Parameters:
  - sampleid (required): identifier of sample (sampleid or barcode). Supports multiple ids separated by comma

  OR

  - samplename (required): sample name
  - status (optional): started | completed | approved
  - startedby (optional): valid User ID in LSM. If status is set to 'started' and this parameter is not received, the default operator selected in API settings will be used instead
  - startedon (optional): datetime (YYYY-MM-DD hh:mm:ss). If status is set to 'started' or 'startedby' is set, and this parameter is not received, current date will be used
  - completedby (optional): valid User ID in LSM. If status is set to 'completed' and this parameter is not received, the default operator selected in API settings will be used instead
  - completedon (optional): datetime (YYYY-MM-DD hh:mm:ss).). If status is set to 'started' or 'startedby' is set and this parameter is not received, current date will be used
  - validatedby (optional): valid User ID in LSM. If status is set to 'approved' and this parameter is not received, the default operator selected in API settings will be used instead
  - validatedon (optional): datetime (YYYY-MM-DD hh:mm:ss).). If status is set to 'approved' or 'validatedby' is set and this parameter is not received, current date will be used
  - assayid (optional): if it is set, the update is applied only to this assay

- Response: OK 200

Note: startedby, completedby and validatedby also accepts LabCollector Contact ID if it is preceded by 'LC', example: LC58

1.3.5. Start a batch

- Method: PUT
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2
- Parameters:
    - batch (required): batch name
    - startedby (optional): user id valid in LSM (or 1 for super admin).
    - startedon (optional): if not set, current date is used

Note: startedby also accepts LabCollector Contact ID if it is preceded by 'LC', example: LC58

1.3.6. Update job

- Method: PUT
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2
- Parameters:
    - jobid (required): identifier of job (job id or barcode). Supports multiple ids separated by comma

    OR

    - jobsamplename (required): sample name. Supports multiple values separated by comma
    - Any job parameter using parameter label or parameter id
    - source (optional): only allowed if job source is empty
    - sourcereport (optional)
    - sourcereportdts (optional)

1.3.7. Accept/Reject a job

- Method: PUT
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2
- Parameters:
    - jobid (required): identifier of job (job id or barcode). Supports multiple ids separated by comma
    - acceptance (required): accept | reject | hold
    - comment (optional): a text comment to the acceptance/rejection
    - sendnotification (optional): 0 | 1 (send notification by email)

1.3.8. Order a job quoted

- Method: PUT
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2
- Parameters:
    - jobid (required): identifier of job (job id or barcode). Supports multiple ids separated by comma
    - order (required): 1
    - purchase order (optional): purchase order

1.3.9. Update requester

- Method: PUT
- URL: http://[my_labcollector_url]/extra_modules/lsm/webservice/index.php?v=2
- Parameters:
    - requesterid (required): identifier of requester (requester id or code or name)
    - any requester field (optional): code, name, taxNumber, discount, company, prefix, registrationid, onPortal, reportPassword, category, reportPreferred, notes
    - any requester billing field (optional): billing.address, billing.city, billing.state, billing.country, billing.zip, billing.phone, billing.fax, billing.contactFirstName, billing.contactLastName, billing.contactEmail

- any requester shipping field (optional): shipping.address, shipping.city, shipping.state, shipping.country, shipping.zip, shipping.phone, shipping.fax, shipping.contactFirstName, shipping.contactLastName, shipping.contactEmail

2. Authentication

One HEADER parameter is mandatory:
-Key: X-LC-APP-Auth
-Value: The token you can view/generate in LabCollector -> Admin -> Others -> Setup -> Web services API

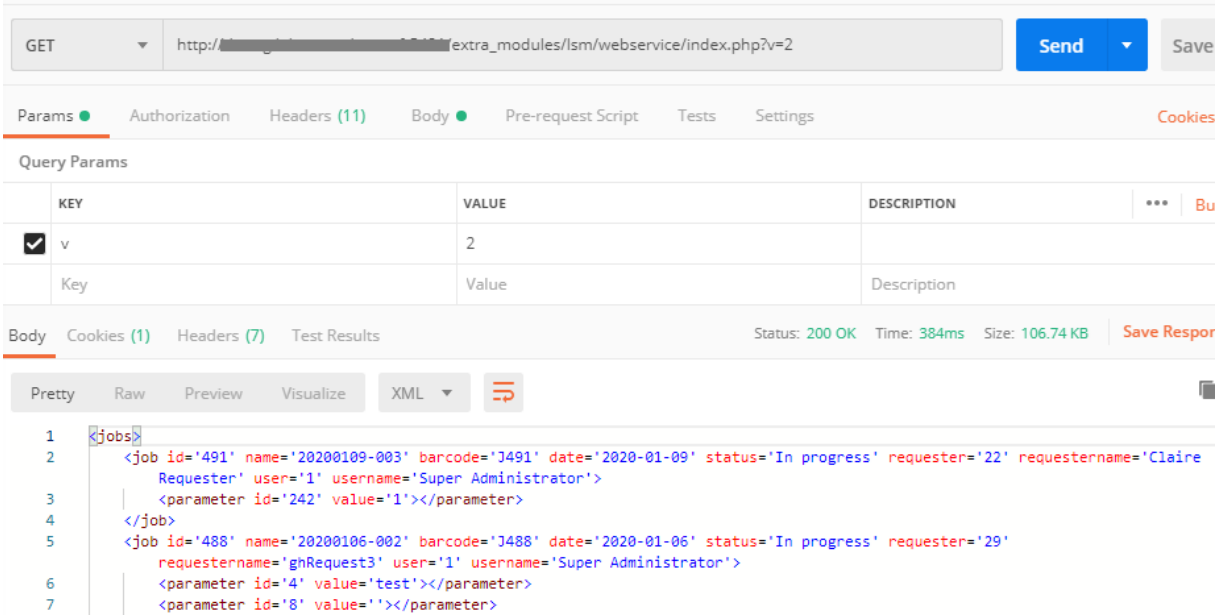
3. Tool

You can try to retrieve data from or send data to the API with some software app as Postman (https://www.getpostman.com/).

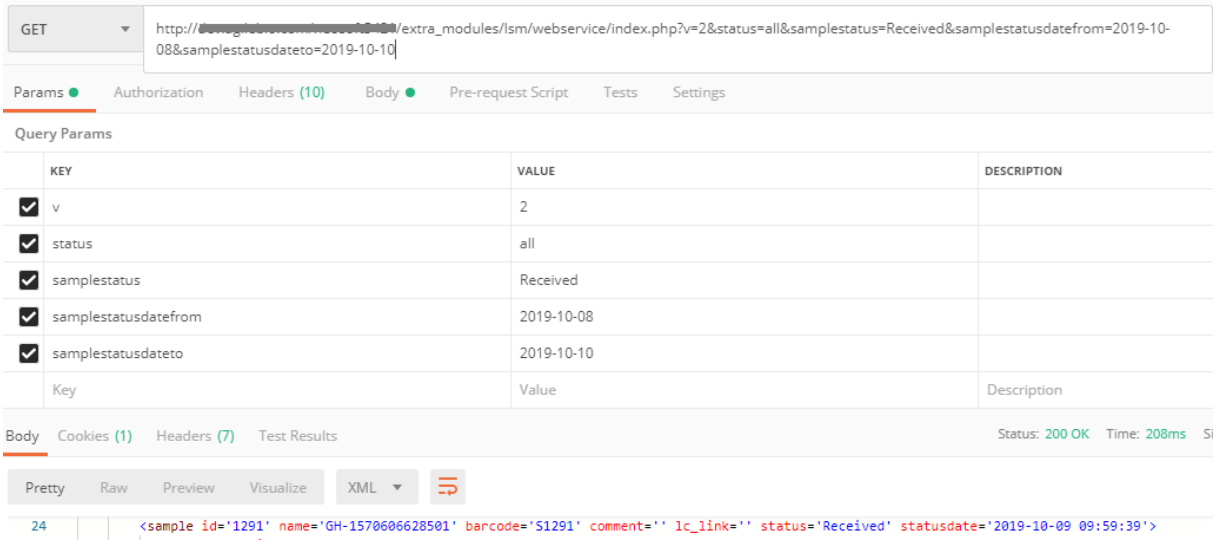
4. Screenshots

4.1. GET

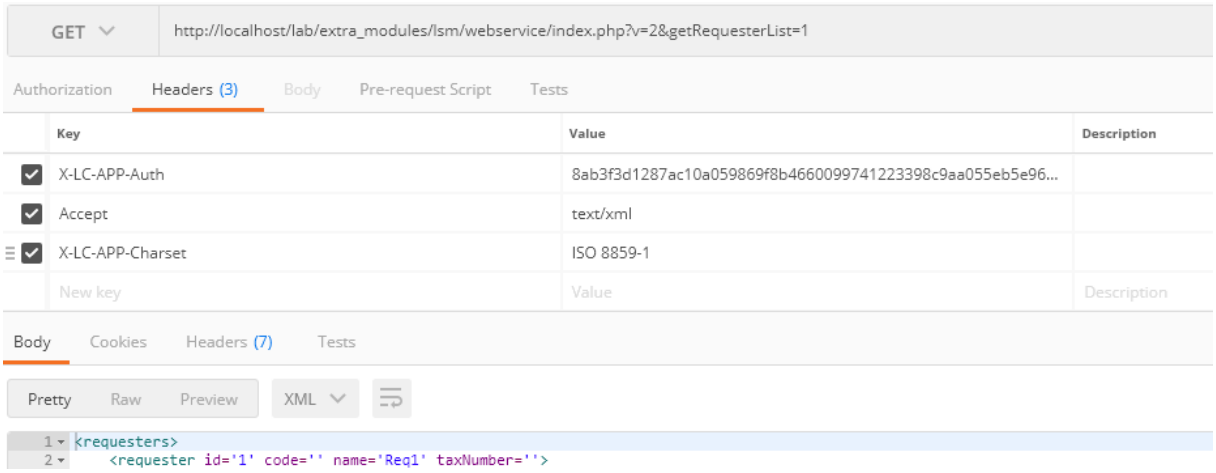
4.1.1. Get list of jobs in progress



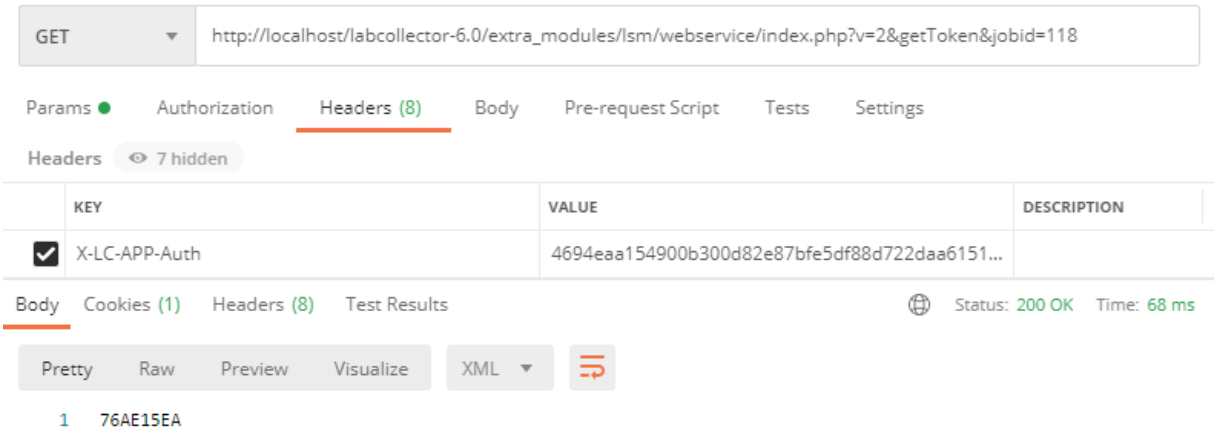
4.1.2. Get list of samples received in a period



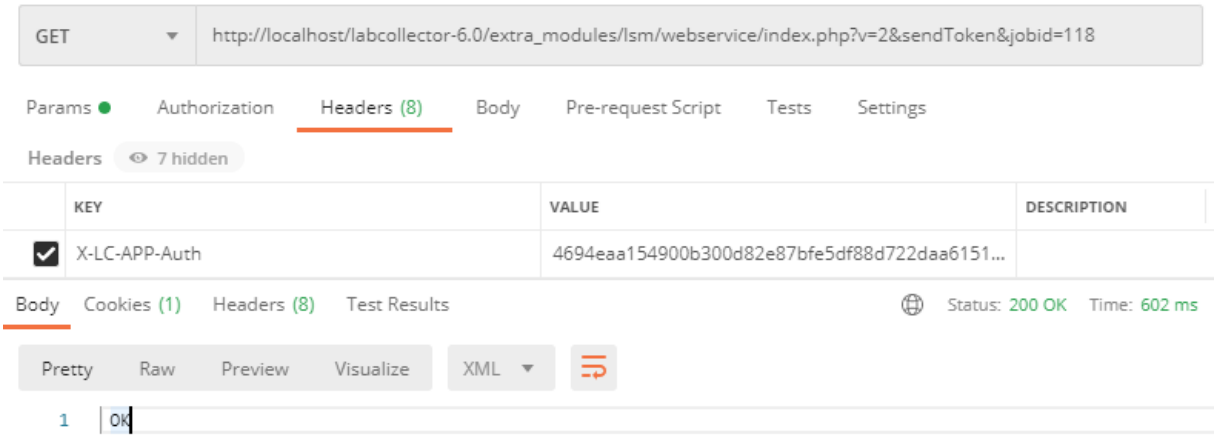
4.1.3. Get list of requesters



4.1.4. Get token

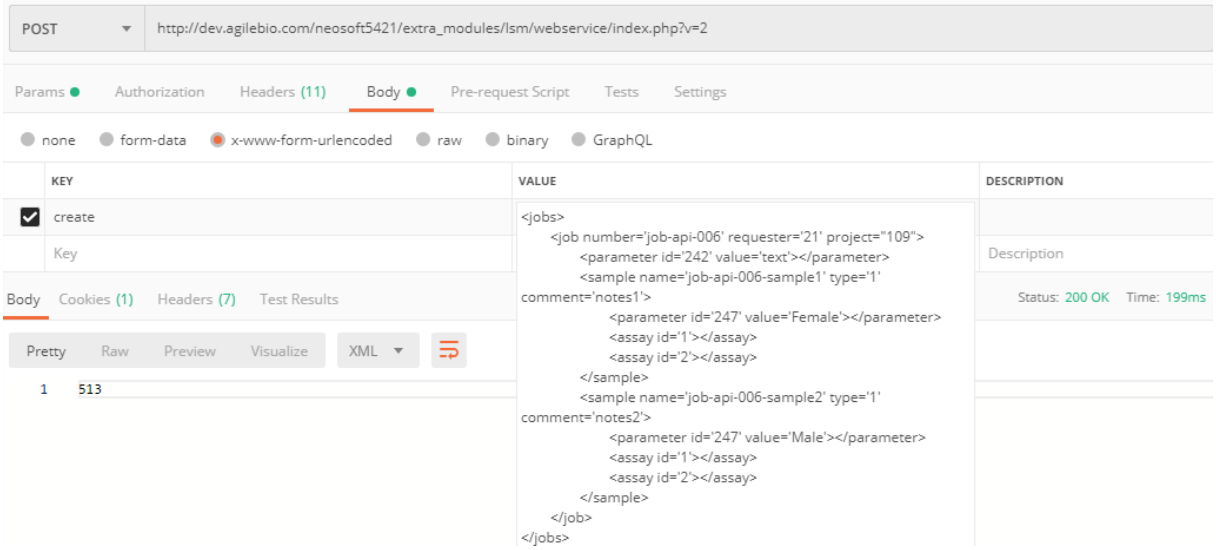


4.1.5. Send token by email

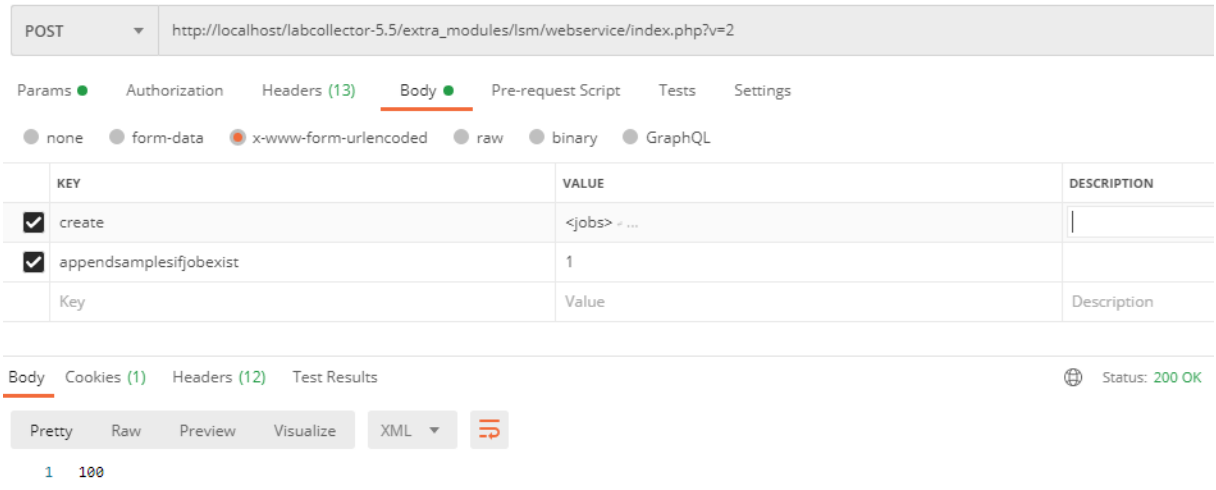


4.2. POST

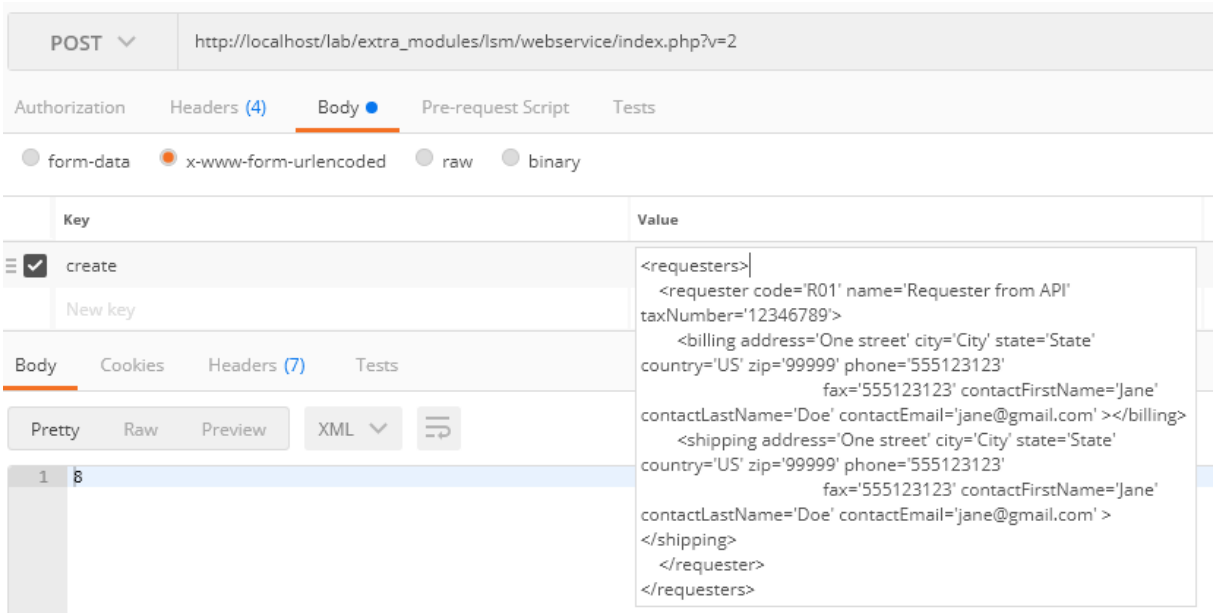
4.2.1. Create jobs



4.2.2. Append samples to job

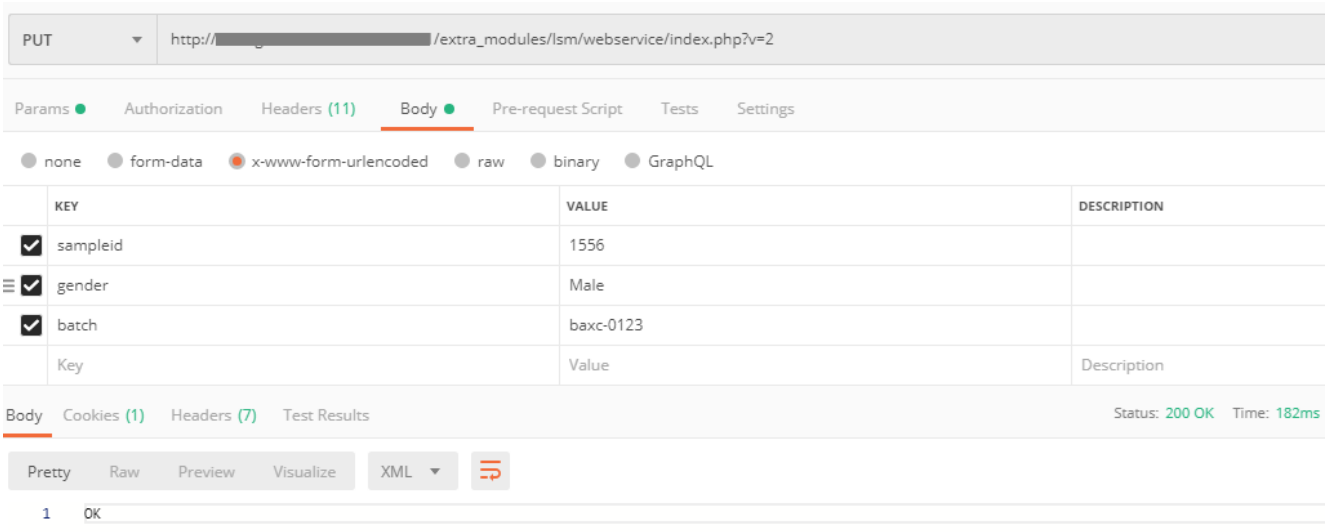


4.2.3. Create requesters

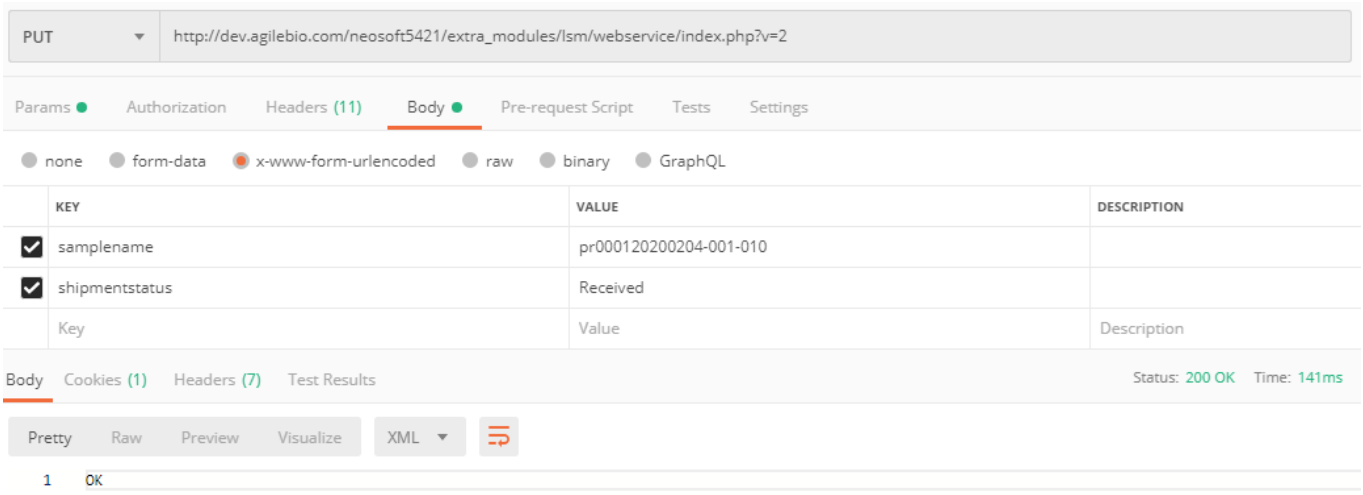


4.3.PUT

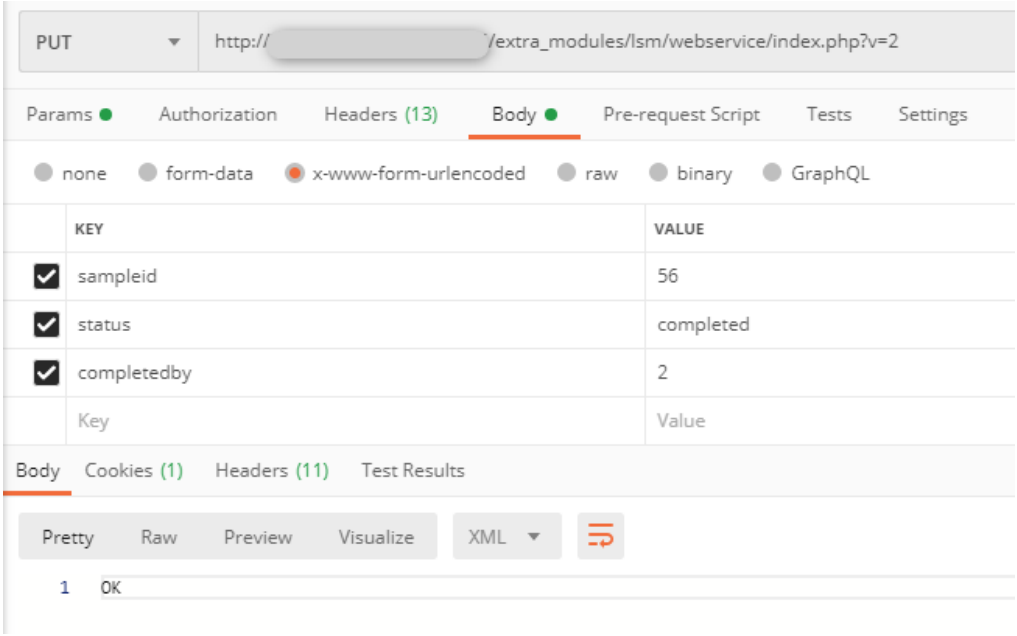
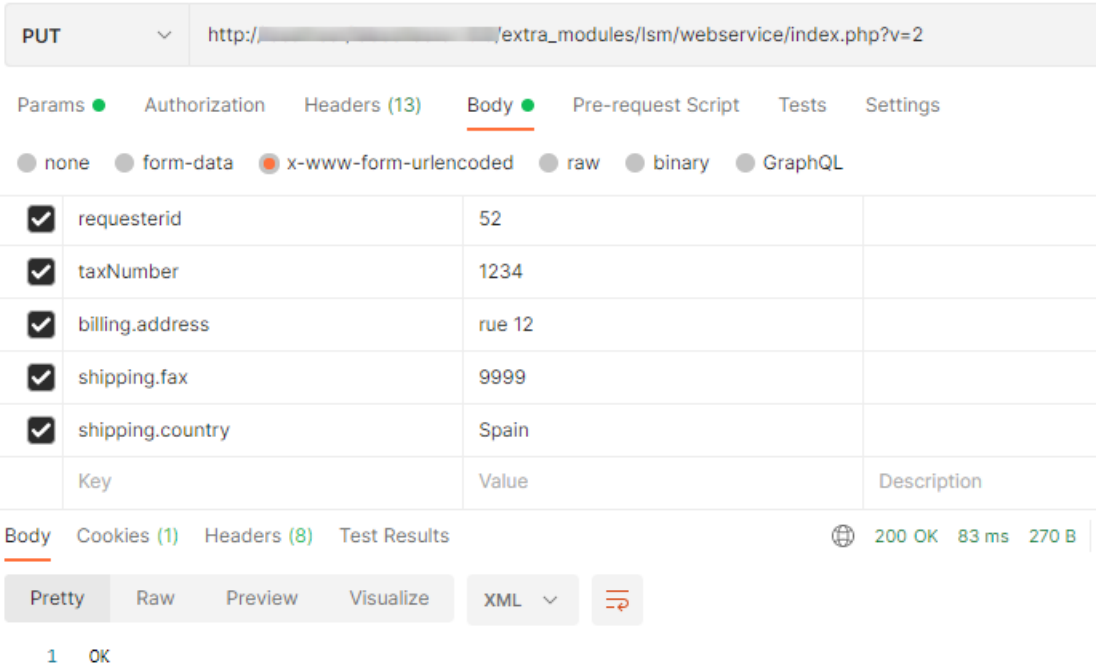
4.3.1. Update sample using id



4.3.2. Update sample using name



4.3.3. Update sample status



4.3.4. Update requester