

LabCollector

"Your Laboratory management solution"

User's guide

Web Service API

Version 2.0 – Mar. 2021



Summary

1- INTRODUCTION	4
2- LABCOLLECTOR API	5
2-1. API setup.....	5
2-2. Requests	6
2-2-1. API Method.....	6
2-2-2. Headers	6
2-2-3. Tool.....	7
2-3. Uniform Resource Identifier (URI)	7
2-3-1. GET method.....	7
2-3-2. POST method.....	13
2-3-3. PUT method.....	15
2-3-4. DELETE method	17
3- LSM API.....	19
3-1. Available methods in LSM web service	19
3-1-1. Jobs.....	19
3-1-2. Equipment	20
3-1-3. Assays.....	20
3-1-4. Categories	21
3-1-5. Sample types	21
3-1-6. Parameters.....	21
3-1-7. Requesters.....	21
3-1-8. Invoices.....	22
3-1-9. Batches.....	22
3-1-10. Create requesters	23
3-1-11. Create job.....	24

3-1-12.	<i>Append samples to existent job</i>	25
3-1-13.	<i>Save results</i>	25
3-1-14.	<i>Upload files</i>	26
3-1-15.	<i>Link sample to module</i>	26
3-1-16.	<i>Update sample</i>	26
3-1-17.	<i>Update sample status</i>	28
3-1-18.	<i>Start a batch</i>	28
3-2.	Authentication	29
4-	ELN API	30
4-1.	URL	30
4-2.	GET	30
4-3.	POST	31
4-4.	Examples	31
5-	API ERROR MESSAGES	32
6-	LABCOLLECTOR WEB SERVICE API - ANNEX	34

1- INTRODUCTION

The LabCollector Web Service Application Programming Interface (API) allows third-party applications to interact with LabCollector's database (modules) and add-ons (ELN and LSM). Note that both ELN and LSM API have a dedicated manual and will be removed from the present manual at some later revision.

Also WorkFlow add-on also has an API and has a specific manual.

For more examples and information please consult the Knowledge base online at <https://labcollector.com/kb>.

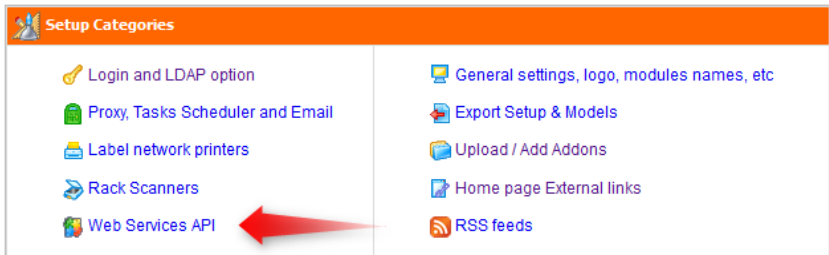
The API is based on a Representational State Transfer (REST) architecture allowing access to resources through Uniform Resource Identifier (URI) and actions on them.

Note: Since June 2017 API v1 was discontinued and all new evolutions are in API v2.

2- LABCOLLECTOR API

2-1. API setup

First of all, you have to declare your application in your LabCollector software. To access the application declaration setup form, log-in to LabCollector with super-administrator rights and go to **Admin > Setup** page. Then select the Web Service API link.



You are now on the Web Service API applications management page. To declare a new application, simply complete this form:

- **Name:** name of your application.
- **Modules connected to this application:** select modules in which the application can access.
- **Default operator:** select the contact that will be the default operator if you don't want to insert this information in each request.
- **IP Restriction:** security option that allows you to declare a list of IP addresses, which will be allowed to perform requests on the API.

The **Application list** shows all the applications for your LabCollector and you can, at any time, modify their scope.

You have also access to the Token which is necessary to identify your application during requests to the API.

Remote applications list:			
Application	Used Modules		IP Restriction
Name: ICollector Default operator: <input type="text" value="Select Contact"/> <input type="button" value="v"/> Token: 1eb2b363bbcedbde113de4fe176253d94202bd8256369de65c71e151f930bb <input type="button" value="Regenerate token"/>	<input type="checkbox"/> Strains & Cells <input type="checkbox"/> Primers <input checked="" type="checkbox"/> Samples <input type="checkbox"/> Sequences <input type="checkbox"/> Equipment <input type="checkbox"/> Documents <input type="checkbox"/> Microarrays <input type="checkbox"/> test pharma <input type="checkbox"/> Parsed data <input type="checkbox"/> storage test <input type="checkbox"/> Hirondele	<input type="checkbox"/> Plasmids <input type="checkbox"/> Reagents & Supplies <input checked="" type="checkbox"/> Antibodies <input type="checkbox"/> Animals <input type="checkbox"/> Chemical Structures <input type="checkbox"/> Address Book <input type="checkbox"/> Clone banks <input type="checkbox"/> Server LC <input type="checkbox"/> Analysis <input type="checkbox"/> Deviation module LSM <input type="checkbox"/> Sample ref	<input type="radio"/> Yes <input checked="" type="radio"/> No <input type="button" value="Update"/> <input type="button" value="Delete app"/>

Note: To use this feature, you need to activate Curl on your PHP preferences.

In Linux, install the PHP-Curl package.

On windows and with our automatic installer, edit PHP.ini and uncomment extensions for Curl (extension=php_curl.dll).

2-2. Requests

The dialog between third-party applications and the LabCollector web service API is based on the HTTP 1.1 protocol.

2-2-1. API Method

You can send HTTP or HTTPS requests to the web service with a method to act on a resource.

- **GET** method for reading a resource
- **POST** method to create a new resource
- **PUT** method to modify a resource
- **DELETE** method to delete a resource

2-2-2. Headers

A request to the API requires some specific HTTP/HTTPS headers:

- The **Accept** header defines the desired response format of your request, text/XML (default), or application/JSON.
- The **X-LC-APP-Auth** header where you put your application token which is necessary to authorize your request to the API.

- The **X-LC-APP-Charset** header defines the character encoding of your application. It allows the API to send back the response with the appropriate encoding and to correctly convert your POST and PUT requests to the LabCollector's character encoding (ISO 8859-1).

2-2-3. Tool

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

2-3. Uniform Resource Identifier (URI)

2-3-1. GET method

Each LabCollector module data is identified by a unique URI (see annex for a complete list of module's URI):

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]
```

This request replies the list of all data in a module.

You can do searching into module data by adding parameters to your URI. You can pass a parameter with a keyword matching a field value, like:

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]?name=[KEYWORD]
```

e.g.

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]?name=First%20Record
```

This request returns the records where their name value contains the "First Record" keyword.

They are some custom parameters that the API uses to perform searching and filtering actions:

- The **record_id** parameter to specify data by its ID:

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]?record_id=[RECORD_ID]
```

e.g. `[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]?record_id=1,19`

This request returns records with ID 1 and ID 19. You can specify multiple IDs by separating them with a comma.

- The **by_keywords** parameter performs a keyword search:

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]?by_keywords=[KEYWORD]
```

e.g. `[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]?by_keywords=cell`

This request performs a search into all fields of all records and returns matching `cell`. You can specify multiple keywords by separating them with a comma.

- The **fields** parameters, if you want to retrieve only some fields values in the API response:

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]?fields=[FIELD1],[FIELD2]
```

e.g. `[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]?fields=count,name`

This request returns all records from the module but with only `count` and `name` fields. You can specify multiple fields by separating them with a comma.

The request accepts now multiple values separated by a comma, for custom fields of type "select"

http://localhost/lab0/webservice/v2/samples?select_list=68,69

GET http://localhost/lab0/webservice/v2/samples?select_list=68,69

Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> select_list	68,69	
Key	Value	Description

Body Cookies (1) Headers (14) Test Results Status: 200 OK Time: 230 ms Size: 2.83 KB

Pretty Raw Preview XML

```

1 <samples>
2   <data>
56  <data>
103 </samples>

```

- The search_on parameter allows you to search for data. And you can use it to search by date range as follow:

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE] &
search_on=date_field&from=XXXXXX&to=ZZZZZ
```

If you only use FROM, the result will give all dates bigger than the FROM date. If you only use TO, it will return all the value until this date.

GET http://localhost/labcollector-5.5/webservice/index.php?v=2&module=strains&fields=count_name&search_on=date_updated&from=2020-01-30 10:00:00&to=2020-09-09

Params Authorization Headers (13) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> v	2	
<input checked="" type="checkbox"/> module	strains	
<input checked="" type="checkbox"/> fields	count_name	
<input checked="" type="checkbox"/> search_on	date_updated	
<input checked="" type="checkbox"/> from	2020-01-30 10:00:00	
<input checked="" type="checkbox"/> to	2020/09/09	
Key	Value	Description

Body Cookies (1) Headers (13) Test Results Status: 200 OK Time: 268 ms

Pretty Raw Preview Visualize XML

```

1 <strains>
2   <data>
3     <count>1</count>
4     <name>strain1</name>
5   </data>
6   <data>
7     <count>5</count>
8     <name>strain4</name>
9   </data>

```

- The **sort_by** parameter allows you to sort your search:

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]?sort_by=[FIELD1]_DESC  
e.g. [PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]?sort_by=name_DESC
```

This request returns all records sorted in descending order on the **name** field. You can specify multiple **sort_by** separating them with a comma and specified order ascendant “_ASC” or descendant “_DESC” for each field.

- The **limit_to** parameter allows you to limit the number of results:

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]?limit_to=0,10
```

This request returns 10 records beginning at index 0. If you do not specify the index, only the number of results indicated is returned.

The API also returns two custom fields in the header response, “X-LC-QUERY-RESULT” containing the number of results returned in the body response and “X-LC-QUERY-TOTAL” containing the total of records matching your search.

Each record also has a unique URI:

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]/[DATA_ID]
```

This request replies a unique record. [DATA_ID] must match the unique ID of the record you want to retrieve.

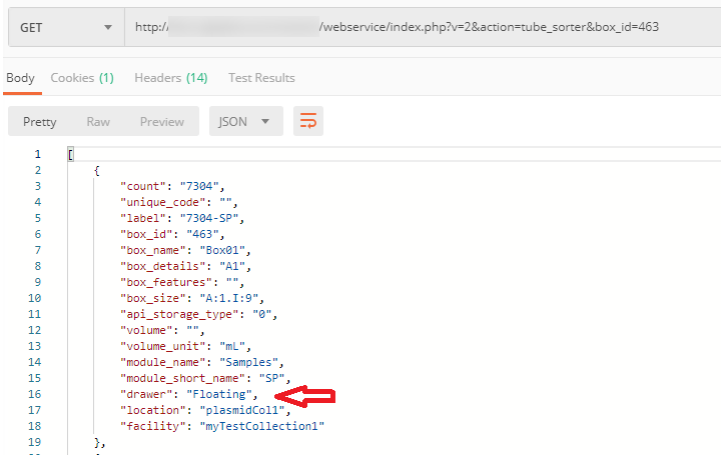
You also have Tube Sorter filtering functions:

```
[PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2&action=tube_sorter&box_id=[BOX_ID]
```

e.g.

```
[PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2&action=tube_sorter&box_id=34
```

This request returns storage info on box ID 34 like tube sorter. You can specify multiple IDs by separating them with a comma.



```
GET http://.../webservice/index.php?v=2&action=tube_sorter&box_id=463

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview JSON

1 {
2   {
3     "count": "7304",
4     "unique_code": "",
5     "label": "7304-SP",
6     "box_id": "463",
7     "box_name": "Box01",
8     "box_details": "A1",
9     "box_features": "",
10    "box_size": "A1.I:9",
11    "api_storage_type": "0",
12    "volume": "",
13    "volume_unit": "mL",
14    "module_name": "Samples",
15    "module_short_name": "SP",
16    "drawer": "Floating",
17    "location": "plasmidColl",
18    "facility": "myTestCollection1"
19  },
20 }
```

```
[PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2&action=tube_sorter&box_id=[BOX_ID]&record_name=[RECORD_NAME]
```

e.g.

```
[PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2&action=tube_sorter&box_id=206&record_name=ST-260
```

```
[PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2&action=tube_sorter&record_name=[RECORD_NAME]
```

e.g.

```
[PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2&action=tube_sorter&record_name=ST-260
```

These requests perform filtering on record named ST-260. You can specify multiple record names by separating them with a comma. You can also specify box ID, here 206.

```
[PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2&action=tube_sorter&box_name=[BOX_NAME]
```

e.g.

```
[PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2&action=tube_sorter&box_name=test-rack_06
```

This request performs filtering on box `test-rack_06`. You can specify multiple box names by separating them with a comma.

Other search parameters to action=tube_sorter can be:

- location_id
- location_name
- facility_id
- facility_name

It will return empty boxes as well.

Other storage actions to consider:

action=getBoxList

action=getBackup

action=getUser

- The **action getLot** allows retrieving lot and reagent info

```
[PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2&action=getLot  
[PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2&action=getLot&lot_  
id=1/LT  
[PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2&action=getLot&chem  
_id=2
```

Optional parameters are lot_id (in format 1 or 1/LT) and chem_id. If it doesn't receive parameters, then it retrieves all active lots.

- The **storage_sec** parameter allows retrieving information about secondary storage.

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]&data_id=[DATA_ID]&fi  
elds=storage_sec
```

LC local secondary storage information

GET http://localhost/Lab_5_422/webservice/index.php?v=2&module=plasmids&data_id=4&fields=storage_sec

Params Authorization Headers (10) Body Pre-request Script Tests

Headers (3)

KEY	VALUE
<input checked="" type="checkbox"/> X-LC-APP-Aush	8e4735b2ff8ab6b0ac3649cc53-d9f
<input checked="" type="checkbox"/> Accept	text/xml
<input checked="" type="checkbox"/> Key	Value

Temporary Headers (7)

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview XML

```

1 <plasmids>
2 <data>
3 <count>4</count>
4 <storage_sec>
5 <box_id>22</box_id>
6 <unique_code>18</unique_code>
7 <box_details>A2</box_details>
8 <box_name>box2</box_name>
9 <keeper>1</keeper>
10 <volume>200</volume>
11 <volume_unit></volume_unit>
12 </storage_sec>
13 <storage_sec>
14 <box_id>12</box_id>
15 <unique_code></unique_code>
16 <box_details>A1</box_details>
17 <box_name>Rack-001-Dr-1</box_name>
18 <keeper>18</keeper>
19 <volume></volume>
20 <volume_unit></volume_unit>
21 </storage_sec>
22 </data>
23 </plasmids>

```

2-3-2. POST method

To create a new resource, simply send a request with the POST method to the desired module URI:

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]
```

Your parameter keys have to match the field's name.

Check for uniqueness fields have been added, when creating new records (POST) or update (PUT)

If exists a different record with the same value for a field Uniqueness, API will not complete the action and will return code 409 (Conflict), and the text: Value for field 'XXX' must be unique. Value 'YYY' already exists in table 'ZZZ'. (see screenshot)

http://localhost/lab0/webservice/v2/strains

POST http://localhost/lab0/webservice/v2/strains Send

Params Authorization Headers (2) **Body** Pre-request Script Tests Cookies Co

none
 form-data
 x-www-form-urlencoded
 raw
 binary

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	name	new_strain	
<input checked="" type="checkbox"/>	unico	abc	
	Key	Value	Description

Body Cookies (1) Headers (12) Test Results Status: 409 Conflict Time: 365 ms Size: 552 B

Pretty Raw Preview XML

```
1 Value for field unico must be unique. Value 'abc' already exists in table 'strains'
```

- The **action addBox** allows you to create a box

```
[PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2&action= addBox
```

- Required parameters:

- o name
- o type (must be a valid type: box, box_nogrid, plate, microplate, visotube, bag, shelfpart)
- o equipment (supports id or name and must exist in LabCollector storage).
- o size (depends on the type of box: should be numeric for a visotube, and the format (A:1.H:8) for a box, a plate, and a microplate)

- Optional parameters:

- o description
- o rack
- o keeper

Other action available:

Action=addLot

To specify what the POST returns after a successful request you can pass a return option. By default you get the internal ID of the newly created resource. Alternatives are:

return=name

return=unique_code

return=id|name

return=id|unique_code

2-3-3. PUT method

To modify a resource, simply send a request with the PUT method to the desired record URI:

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]/[DATA_ID]
```

Your parameter keys have to match the field's name you want to modify.

For the following actions, note that for PUT request, parameters must be on the body (not in URL).

The URL is [PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2

The headers are: X-LC-APP-Auth, Accept.

- **Remove Volume**

- Parameters:

- removeVolume (mandatory)
- barcode, unique_code or aliquot_barcode (one of them must be present)
- quantity (mandatory)

- Response: OK

- **Remove Storage**

- Parameters:

- removeStorage (mandatory)
- barcode, unique_code or aliquot_barcode (one of them must be present)

- Response: OK

- **Add Registry Book**

- URL:

[PATH_TO_LABCOLLECTOR]/webservice/index.php?v=2&module=[module]

- Parameters:

- `addRegistryBook` (mandatory)
- `record_id` (mandatory)
- `date` (mandatory, format yyyy/mm/dd or yyyy-mm-dd)
- `comments` (mandatory)
- `operator` (optional, if it does not send the API default operator will be used)
- `action` (optional, must be a valid 'Storage Action Type' defined in [LC > Admin > Preferences > Process & Actions Type](#))

- Response: OK

- **Add Secondary storage**

- Parameters:

- `addSecondaryStorage` (mandatory)
- `barcode` (mandatory)
- `box_id` (mandatory)
- `box_details` (mandatory only for the box with grid divider, tube tray, and microplate. If the box is without grid, a bag, a visotube or a shelf part, it is not required)
- `unique_code` (optional)
- `volume` (optional)
- `comments` (optional)
- `cap_color` (optional)

Note: An error message is returned if mandatory parameters are not present; if the barcode does not exist; if the `unique_code` is present, but it isn't unique; and, if the color is present but it doesn't exist.

If the parameter `box_details` is not received and the type of box needs position (box with grid, tube tray, or microplate), an error message is returned.

- **Return definitions**

To specify what the PUT returns after a successful request you can pass a return option. By default you get the internal ID of the newly created resource. Alternatives are:

return=name
return=unique_code
return=id|name
return=id|unique_code

A screenshot of a REST client interface showing a PUT request to `http://localhost/labcollector-5.5/webservice/index.php?v=2`. The 'Body' tab is selected, and the 'x-www-form-urlencoded' radio button is chosen. The body is a form with the following data:

KEY	VALUE
<input checked="" type="checkbox"/> addSecondaryStorage	1
<input checked="" type="checkbox"/> unique_code	test_comil_by_api_3
<input checked="" type="checkbox"/> barcode	84/ST
<input checked="" type="checkbox"/> volume	100
<input checked="" type="checkbox"/> box_id	481
<input checked="" type="checkbox"/> box_details	C7,C8
<input checked="" type="checkbox"/> comments	3nd test comment inserting secondary storage via api
<input checked="" type="checkbox"/> cap_color	a00e0e
Key	Value

A screenshot of a REST client interface showing a PUT request to `http://localhost/labcollector-5.5/webservice/index.php?v=2`. The 'Body' tab is selected, and the 'x-www-form-urlencoded' radio button is chosen. The body is a form with the following data:

KEY	VALUE
<input checked="" type="checkbox"/> addSecondaryStorage	1
<input checked="" type="checkbox"/> barcode	1/ST
<input checked="" type="checkbox"/> box_id	3
Key	Value

Below the body, the 'Test Results' tab is active, showing a single message:

```
1 Position in required for this type of box. Please, provide a value for box_details
```

2-3-4. DELETE method

To delete a resource, simply send a request with the DELETE method to the desired record URI:

```
[PATH_TO_LABCOLLECTOR]/webservice/v2/[MODULE]/[DATA_ID]
```

3- LSM API

A dedicated manual for LSM API exists. This section will be removed from the LabCollector API at some point. Visit our website to get the latests LSM API manual.

3-1. Available methods in LSM web service

Note: URL works with HTTP and HTTPS

3-1-1. Jobs

- Method: GET
- URL: [PATH_TO_LSM]/webservice/index.php?v=2
- Optional parameters:
 - datefrom
 - dateto
 - jobid (id or barcode)
 - sampleid (id or barcode)
 - assayed
 - status (by default only jobs in progress are returned, use status=all to view all the jobs)
 - samplename
 - samplestatus
 - samplestatusdatefrom
 - samplestatusdateto
 - assaystatus
 - batch
 - assaystatusdatefrom
 - assaystatusdateto
 - startedon
 - completedon
 - validatedon

GET `http://[redacted]extra_modules/lsm/webservice/index.php?v=2` Send Save

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	BU
<input checked="" type="checkbox"/> v	2			
Key	Value	Description		

Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 384ms Size: 106.74 KB Save Respor

Pretty Raw Preview Visualize XML ⌵ ⌵

```

1  <?xml version="1.0" encoding="UTF-8" ?>
2  <jobs>
3  <job id="491" name="20200109-003" barcode="J491" date="2020-01-09" status="In progress" requester="22" requestername="Claire
4  Requester" user="1" username="Super Administrator">
5  <parameter id="242" value="1"></parameter>
6  </job>
7  <job id="408" name="20200106-002" barcode="J408" date="2020-01-06" status="In progress" requester="29"
  requestername="ghRequest3" user="1" username="Super Administrator">
  <parameter id="4" value="test"></parameter>
  <parameter id="8" value=""></parameter>

```

GET `http://[redacted]extra_modules/lsm/webservice/index.php?v=2&status=all&samplestatus=Received&samplestatusdatefrom=2019-10-08&samplestatusdateto=2019-10-10`

Params Authorization Headers (10) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> v	2	
<input checked="" type="checkbox"/> status	all	
<input checked="" type="checkbox"/> samplestatus	Received	
<input checked="" type="checkbox"/> samplestatusdatefrom	2019-10-08	
<input checked="" type="checkbox"/> samplestatusdateto	2019-10-10	
Key	Value	Description

Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 208ms Si

Pretty Raw Preview Visualize XML ⌵ ⌵

```

24  <sample id="1291" name="0H-1570606628501" barcode="S1291" comment="" lc_link="" status="Received" statusdate="2019-10-09 09:59:39">

```

3-1-2. Equipment

- Method: GET
- URL: `[PATH_TO_LSM]/webservice/index.php?v=2&getEquipmentList=1`
- Optional parameter: id

3-1-3. Assays

- Method: GET

- URL:[PATH_TO_LSM]/webservice/index.php?v=2&getAssayList=1
- Optional parameter: id

3-1-4. Categories

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

3-1-5. Sample types

- Method: GET
- URL:[PATH_TO_LSM]/webservice/index.php?v=2&getSampleTypeList=1
- Optional parameter: id

3-1-6. Parameters

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

3-1-7. Requesters

- Method: GET
- URL:
[PATH_TO_LSM]/webservice/index.php?v=2&getRequesterList=1

- Optional parameter: `id`

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `http://localhost/lab/extra_modules/lsm/webservice/index.php?v=2&getRequesterList=1`
- Headers (3):**
 - X-LC-APP-Auth: `8ab3f3d1287ac10a059869f8b4660099'`
 - Accept: `text/xml`
 - X-LC-APP-Charset: `ISO 8859-1`
- Body:**

```

1 <requesters>
2   <requester id='1' code='' name='Req1' taxNumber=''>

```

3-1-8. Invoices

- Method: GET
- URL: `[PATH_TO_LSM]/webservice/index.php?v=2&getInvoiceList=1`
- Optional parameters:
 - `id`
 - `datefrom`
 - `dateto`
 - `status`
 - `format` (CSV | IIF)

3-1-9. Batches

- Method: GET
- URL: `[PATH_TO_LSM]/webservice/index.php?v=2&getBatchList=1`
- Optional parameters:
 - *Batch*
 - *status: notfinished –default– | all | assigned | started | completed | approved*

3-1-10. Create requesters

- Method: POST
- URL: [PATH_TO_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>
```

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost/lab/extra_modules/lsm/webservice/index.php?v=2
- Authorization:** Headers (4)
- Body:** Selected tab, showing form-data selected.
- Key:** create
- Value:** XML request body:

```
<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>
```

3-1-11. Create job

- Method: POST
- URL: [PATH_TO_LSM]/webservice/index.php?v=2
- **Required parameter: create**

Example of create:

```
<jobs>
  <job number='XP-001' requester='1'>
    <parameter id='1' value='text'></parameter>
    <sample name='sample01' type='1' comment='notes'>
      <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 supports id or text.

Attribute project supports id or text.

KEY	VALUE
<input checked="" type="checkbox"/> create	<jobs> <job number='job-api-006' requester='21' project='109'> <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> </jobs>

3-1-12. Append samples to existent job

Use the same notation of [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 it is already completed it returns an error message.

3-1-13. Save results

- Method: POST
- URL: [PATH_TO_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>
```

3-1-14. Upload files

- Method: PUT
- URL: [PATH_TO_LSM]/webservice/index.php?v=2
- **Required parameters:**
 - **upload**
 - **file**
 - **uniqueid** (identifier of job-sample-assay)
- Optional parameters: supports parameter *id* or *name*
- Response: OK 200

3-1-15. Link sample to module

- Method: PUT
- URL: [PATH_TO_LSM]/webservice/index.php?v=2
- **Required parameters:**
 - **sampleid** (identifier of the sample, ID or barcode) or *samplename*
 - **module** (module name or short name)
 - **recordid** (identifier of record in the module)
- Response: OK 200

3-1-16. Update sample

- Method: PUT
- URL: [PATH_TO_LSM]/webservice/index.php?v=2
- Required parameters:
 - **sampleid** (identifier of the sample – ID or barcode – , supports multiple ids separated by a comma) or *samplename*
- Optional parameters:

- shipmentstatus (supports the following numbers or labels: 0|Submitted, 1|In transit, 2|Received, 3|Returned)
- name
- type
- batch

* Note: sampleid supports multiple ids separated by a comma.

PUT http://.../extra_modules/lsm/webservice/index.php?v=2

Params Authorization Headers (11) **Body** Pre-request Script Tests Settings

none form-data **x-www-form-urlencoded** raw binary GraphQL

KEY	VALUE
<input checked="" type="checkbox"/> sampleid	1556
<input checked="" type="checkbox"/> gender	Male
<input checked="" type="checkbox"/> batch	baxc-0123
Key	Value

Body Cookies (1) Headers (7) Test Results

Pretty Raw Preview Visualize XML

1 OK

PUT http://dev.agilebio.com/neosoft5421/extra_modules/lsm/webservice/index.php?v=2

Params Authorization Headers (11) **Body** Pre-request Script Tests Settings

none form-data **x-www-form-urlencoded** raw binary GraphQL

KEY	VALUE
<input checked="" type="checkbox"/> samplename	pr000120200204-001-010
<input checked="" type="checkbox"/> shipmentstatus	Received
Key	Value

Body Cookies (1) Headers (7) Test Results

Pretty Raw Preview Visualize XML

1 OK

3-1-17. Update sample status

- Method: PUT
- URL: [PATH_TO_LSM]/webservice/index.php?v=2
- Required parameters:
 - `sampleid` (identifier of the sample – ID or barcode –, supports multiple ids separated by a comma) or `samplename`
- Optional parameters:
 - `status` supports the following value: `started`|`completed`|`approved`
 - `startedby` (valid User ID in LSM)
 - `startedon`
 - `completedby`
 - `completedon`
 - `validatedby`
 - `validatedon`
- Response: OK 200

* Note: the datetime fields are in the form of YYYY-MM-DD hh:mm:ss.

** Note: If the status is set to 'started' and the parameters `Startedby`, `startedon`, `completedby`, `completedon`, `validatedby`, `validatedon` are not received, the default operator selected in API settings or the current date will be used instead.

The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** http://[redacted]/extra_modules/lsm/webservice/index.php?v=2
- Params:** Authorization, Headers (13), Body (selected), Pre-request Script, Tests, Settings
- Body Type:** x-www-form-urlencoded (selected)
- Body Parameters:**

KEY	VALUE
<input checked="" type="checkbox"/> sampleid	56
<input checked="" type="checkbox"/> status	completed
<input checked="" type="checkbox"/> completedby	2
Key	Value
- Response:** Body, Cookies (1), Headers (11), Test Results
- Response Format:** Pretty (selected), Raw, Preview, Visualize, XML
- Response Content:** 1 OK

3-1-18. Start a batch

- Method: PUT
- URL: [PATH_TO_LSM]/webservice/index.php?v=2
- Required parameters:
 - batch (batch name)
 - startedby (user id valid in LSM (or 1 for super admin))
- Optional parameters:
 - *startedon*

3-2. 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](#)

4- ELN API

4-1. URL

- URL: [PATH_TO_ELN]/webservice/index.php?v=2
- HEADER PARAMETERS:
 - X-LC-APP-Auth
 - Accept=application\json or Accept = text\xml

Note: URL works with HTTPS and HTTPS

4-2. GET

Recover data of books, experiments & pages.

- Mandatory parameter: UID (user identifier), supports numeric value or string value (=login). Example UID=1 or UID=admin
- Optional parameters:
 - Bookid: book identifier, example: bookid=1
 - Datefrom: books created after of, example: datefrom=2010-01-01
 - Dateto: books created before of, example: datefrom=2016-01-01
- Action and examples:

- action= getBooks

[PATH_TO_ELN]/webservice/index.php?v=2&UID=1&action=getBooks

- action= getExperiments

[PATH_TO_ELN]/webservice/index.php?v=2&UID=1&action=getExperiments&bookid=1

- action= getPages

[PATH_TO_ELN]/webservice/index.php?v=2&UID=1&action=getPages&experimentid=7

- action= getPagelimages

[PATH_TO_ELN]/webservice/index.php?v=2&UID=1&action=getPageImages&pageid=3

- action= getPageContent

[PATH_TO_ELN]/webservice/index.php?v=2&UID=1&action=getPageContent&pageid=58

Note: field *page_img_status* means if the page has an image or not: '1' means the page has an image, '0' means the page has not an image

4-3. POST

Create or update pages and send images and/or content

- Mandatory parameter: UID (user identifier) example UID=1
- Optional parameters:
 - image: image file -form data-
 - content: html content
 - pageid: page identifier , example: pageid=134
 - experimented: experiment identifier, example: experimentid=8
 - name: page name, example: name=page from api

Note: If the page is not created you can also create page, leaving out parameter pageid and adding parameters experimentid and name:

- Response: the response is formatted as JSON or XML (not just a text OK), example:

```
{
  "@attributes": {
    "message": "OK"
  }
}
```

4-4. Examples

- To create a new page, use parameters: experimentid and name and optionally content and/or image.

- To update an existing page, use parameters: pageid and optionally content and/or image.

5- API ERROR MESSAGES

Message	Response code	Description
'Requires application authentication to access the Web Service'	401 Unauthorized	The request either does not have the header parameter X-LC-APP-Auth or does not have a valid value
'Invalid Action xxx'	400 Bad Request	Parameter action has a different value to 'tube_sorter' or 'getBackup'
Missing Search Parameters!	400 Bad Request	The request contains the parameter Action=tube_sorter but it is missing at least one of the following parameters: box_id, box_name, record_name, unique_code, barcode, aliquot_barcode
'Module "XXX" does not exist!'	400 Bad Request	The value of the parameter 'module' is not a LabCollector module
'Module "XXX" does not share this data!'	403 Forbidden	The value of the parameter 'module' is not checked on LabCollector > Admin > Setup > Web service
'The format of the request is not accepted!'	415 Unsupported Media Type	The parameter Accept is used, but the value is not one of the accepted values: application/XML or application/JSON
(Empty)	406 Not Acceptable	The method must be one of the following: GET, POST, PUT, DELETE
'No data found.'	404 Not Found	No data found with this request's parameters
'OK.'	200 OK	Record updated successfully

'Conflict.'	409 Conflict	The record could not be updated because there is a conflict in data
'No organisms value for this module'	404 Not Found	Only the modules "strains", "samples" and "microarrays" have an organism value – you have chosen an incorrect module
'No categories value for this module'	404 Not Found	Only the module 'docs' has categories – you have chosen an incorrect module
'Webservice requires user authentication'	401 Unauthorized	Deprecated
'Your IP is not allowed to access this Web Service'	401 Unauthorized	The client IP is not in the list of authorized IPs for this webservice (LC > Admin > Setup > Web service)
'Error during your request, the following information is mandatory to create a new record: X, Y, Z '	400 Bad Request	Attempt to post new data without mandatory fields X, Y, Z
'An error has occurred during your request, the following information is mandatory to remove volume: unique_code or barcode or aliquot_barcode, quantity, quantity'	400 Bad Request	Attempt to remove volume without mandatory parameters: unique_code or barcode or aliquot_barcode, quantity
'An error has occurred during your request, the following information is mandatory to remove storage: unique_code or barcode or aliquot_barcode, quantity '	400 Bad Request	Attempt to remove storage without a mandatory parameter: unique_code or barcode or aliquot_barcode
''	200 OK	Data requested was returned successfully

6- LABCOLLECTOR WEB SERVICE API - ANNEX

The URI system of the API uses a simple and clean URL. Be sure to enable the rewrite engine from Apache to use the URI referenced in the following table.

If the LabCollector server does not support the rewrite engine please use the full URL pattern for your request (secondary URL of each line).

URI	Module	Description
webservice/v2/strains webservice/index.php?v=2&module=strains	GET POST	Strains & Cells List of all records
webservice/v2/strains/[DATA_ID] webservice/index.php?v=2&module=strains&data_id=[DATA_ID]	GET PUT	Strains & Cells Unique record
webservice/v2/strains/customfields webservice/index.php?v=2&module=strains&getModuleCustomFields=1	GET	Strains & Cells List of custom fields
webservice/v2/strains/organisms webservice/index.php?v=2&module=strains&getModuleOrganisms=1	GET	Strains & Cells List of organisms
webservice/v2/plasmids webservice/index.php?v=2&module=plasmids	GET POST	Plasmids List of all records
webservice/v2/plasmids/[DATA_ID] webservice/index.php?v=2&module=plasmids&data_id=[DATA_ID]	GET PUT	Plasmids Unique record
webservice/v2/plasmids/customfields webservice/index.php?v=2&module=plasmids&getModuleCustomFields=1	GET	Plasmids List of custom fields
webservice/v2/primers webservice/index.php?v=2&module=primers	GET POST	Primers List of all records
webservice/v2/primers/[DATA_ID] webservice/index.php?v=2&module=primers&data_id=[DATA_ID]	GET PUT	Primers Unique record
webservice/v2/primers/customfields	GET	Primers List of custom fields

LABCOLLECTOR WEB SERVICE API - ANNEX

webservice/index.php?v=2&module=pri mers&getModuleCustomFields=1			
webservice/v2/chemicals webservice/index.php?v=2&module=che micals	GET POST	Reagents & Supplies	List of all records
webservice/v2/chemicals/[DATA_ID] webservice/index.php?v=2&module=che micals&data_id=[DATA_ID]	GET PUT	Reagents & Supplies	Unique record
webservice/v2/chemicals/customfields webservice/index.php?v=2&module=che micals&getModuleCustomFields=1	GET	Reagents & Supplies	List of custom fields
webservice/v2/samples webservice/index.php?v=2&module=sam ples	GET POST	Samples	List of all records
webservice/v2/samples/[DATA_ID] webservice/index.php?v=2&module=sam ples&data_id=[DATA_ID]	GET PUT	Samples	Unique record
webservice/v2/samples/customfields webservice/index.php?v=2&module=sam ples&getModuleCustomFields=1	GET	Samples	List of custom fields
webservice/v2/samples/organisms webservice/index.php?v=2&module=sam ples&getModuleOrganisms=1	GET	Samples	List of organisms
webservice/v2/samples/types webservice/index.php?v=2&module=sam ples&getModuleTypes=1	GET	Samples	List of sample types
webservice/v2/antibodies webservice/index.php?v=2&module=anti bodies	GET POST	Antibodies	List of all records
webservice/v2/antibodies/[DATA_ID] webservice/index.php?v=2&module=anti bodies&data_id=[DATA_ID]	GET PUT	Antibodies	Unique record
webservice/v2/antibodies/customfields webservice/index.php?v=2&module=anti bodies&getModuleCustomFields=1	GET	Antibodies	List of custom fields
webservice/v2/sequences webservice/index.php?v=2&module=seq uences	GET POST	Sequences	List of all records

LABCOLLECTOR WEB SERVICE API - ANNEX

webservice/v2/sequences/[DATA_ID] webservice/index.php?v=2&module=sequences&data_id=[DATA_ID]	GET PUT	Sequences	Unique record
webservice/v2/sequences/customfields webservice/index.php?v=2&module=sequences&getModuleCustomFields=1	GET	Sequences	List of custom fields
webservice/v2/animals webservice/index.php?v=2&module=animals	GET POST	Animals	List of all records
webservice/v2/animals/[DATA_ID] webservice/index.php?v=2&module=animals&data_id=[DATA_ID]	GET PUT	Animals	Unique record
webservice/v2/animals/customfields webservice/index.php?v=2&module=animals&getModuleCustomFields=1	GET	Animals	List of custom fields
webservice/v2/equipments webservice/index.php?v=2&module=equipments	GET POST	Equipment	List of all records
webservice/v2/equipments/[DATA_ID] webservice/index.php?v=2&module=equipments&data_id=[DATA_ID]	GET PUT	Equipment	Unique record
webservice/v2/equipments/customfields webservice/index.php?v=2&module=equipments&getModuleCustomFields=1	GET	Equipment	List of custom fields
webservice/v2/structures webservice/index.php?v=2&module=structures	GET POST	Chemical Structures	List of all records
webservice/v2/structures/[DATA_ID] webservice/index.php?v=2&module=structures&data_id=[DATA_ID]	GET PUT	Chemical Structures	Unique record
webservice/v2/structures/customfields webservice/index.php?v=2&module=structures&getModuleCustomFields=1	GET	Chemical Structures	List of custom fields
webservice/v2/docs webservice/index.php?v=2&module=docs	GET POST	Documents	List of all records
webservice/v2/docs/[DATA_ID] webservice/index.php?v=2&module=docs&data_id=[DATA_ID]	GET PUT	Documents	Unique record

LABCOLLECTOR WEB SERVICE API - ANNEX

webservice/v2/docs/customfields webservice/index.php?v=2&module=docs &getModuleCustomFields=1	GET	Documents	List of custom fields
webservice/v2/docs/categories webservice/index.php?v=2&module=docs &getModuleCategories=1	GET	Documents	List of categories
webservice/v2/abook webservice/index.php?v=2&module=abook	GET POST	Address Book	List of all records
webservice/v2/abook/[DATA_ID] webservice/index.php?v=2&module=abook &data_id=[DATA_ID]	GET PUT	Address Book	Unique record
webservice/v2/abook/customfields webservice/index.php?v=2&module=abook &getModuleCustomFields=1	GET	Address Book	List of custom fields
webservice/v2/abook/categories webservice/index.php?v=2&module=abook &getModuleCategories=1	GET	Address Book	List of categories
webservice/v2/microarrays webservice/index.php?v=2&module=microarrays	GET POST	Microarrays	List of all records
webservice/v2/microarrays/[DATA_ID] webservice/index.php?v=2&module=microarrays &data_id=[DATA_ID]	GET PUT	Microarrays	Unique record
webservice/v2/microarrays/customfields webservice/index.php?v=2&module=microarrays &getModuleCustomFields=1	GET	Microarrays	List of custom fields
webservice/v2/microarrays/organisms webservice/index.php?v=2&module=microarrays &getModuleOrganisms=1	GET	Microarrays	List of organisms
webservice/v2/[CUSTOM_MODULE_NAME] webservice/index.php?v=2&module=[CUSTOM_MODULE_NAME]	GET POST	Custom Module	List of all records
webservice/v2/[CUSTOM_MODULE_NAME]/[DATA_ID] webservice/index.php?v=2&module=[CUSTOM_MODULE_NAME] &data_id=[DATA_ID]	GET PUT	Custom Module	Unique record

webservice/v2/[CUSTOM_MODULE_NAME]/customfields webservice/index.php?v=2&module=[CUSTOM_MODULE_NAME]&getModuleCustomFields=1	GET	Custom Module	List of custom fields
---	------------	---------------	-----------------------



<http://www.labcollector.com>
sales@agilebio.com

AgileBio USA

5473 Kearny Villa Road Suite 255
San Diego, CA 92123
USA

Tel: 347 368 1315
Fax: (800) 453 9128

<http://www.agilebio.com>

AgileBio Headquarters

75 rue de Lourmel
75015 Paris
FRANCE

Tel: 01 41 79 15 85
Fax: 01 72 70 40 22