# LiveAction API Quick Start Guide

## Contents

## What is the API?

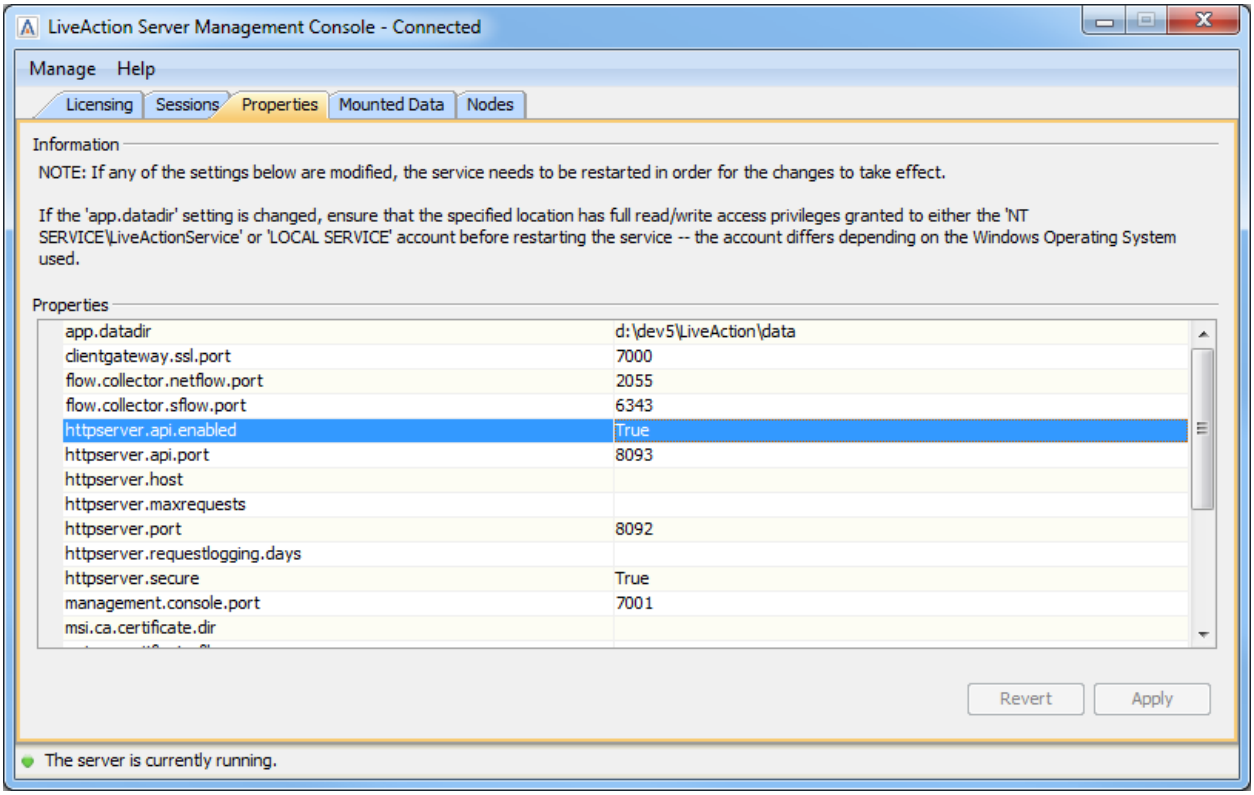
The LiveAction API is a way for users to access captured network data on the LiveAction server without direct use of the LiveAction Client. This allows the users to build up their own special reports or analysis through a programmatic way based upon the LiveAction data.

## How do I get started?

### 1) Enable the API Feature

The first step of using the API requires that the user has access to the LiveAction Management Console on the LiveAction Server machine. Enabling the API can then be done through the following steps:

1. Open the LiveAction Server Management Console
2. Click on the “Properties” tab
3. In the “Properties” pane, click on the “httpserver.api.enabled” property and select “True” from the dropdown box

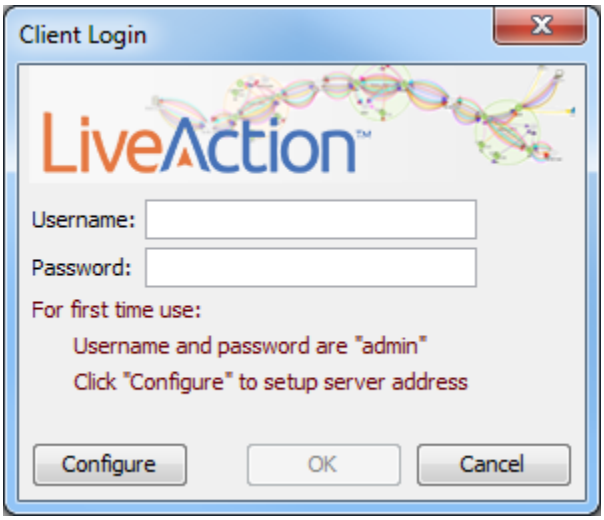



4. [OPTIONAL] Click on the “httpserver.api.port” property and adjust the port of the server that is desired for accessing the API.

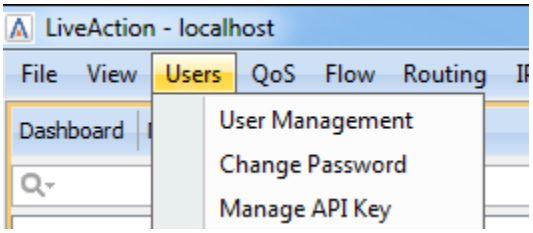
## 2) Acquire an API Key

The second step in accessing the API will be to acquire an API key from the client. This can be done through the following steps:

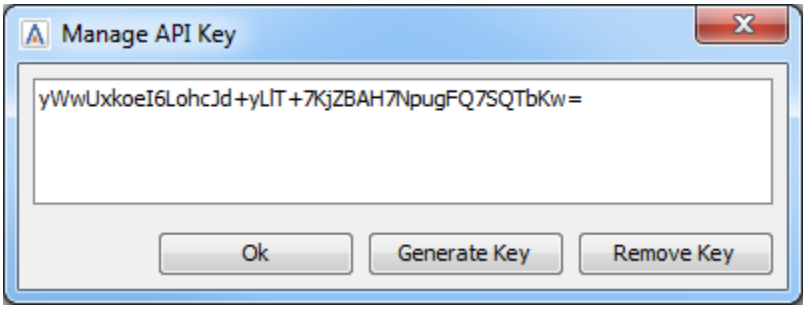
1. Open the LiveAction client and login in with the appropriate Username / Password credentials




2. Click on the “Users” button in the main screen header menu bar, and then click on the “Manage API Key” button. A new dialog will appear for managing the API key.




3. Click on the “Generate Key” button to create a new key. A key will appear in the dialog which can be copied to the clipboard.




This key can now be used for accessing the API.

## Using the API to Execute Requests

Once an API key has been generated, a user should be able to access the API resources by adding this key into the HTTP request headers. For simplicity's sake, we'll use curl for documenting examples. We'll also use the base URL "https://localhost:8093" as an example of the LiveAction server address/port. The API key used will be the value shown in the screenshot above.

A valid request can be performed by adding the key to the authorization header of the API request as a Bearer token as seen below.

```
$ curl -k -H "Authorization: Bearer
yWwUxkoeI6LohcJd+yLlT+7KjZBAH7NpugFQ7SQTbKw="
"https://localhost:8093/v1/status"
```

This should provide a response that looks like the following:

```
{
  "timestamp" : 1425428826845
}
```

The -k parameter is used because HTTPS is used by LiveAction, however it is a self-signed certificate so it will not be able to confirmed by a third party. The -H parameter is used to set the header information for the request.

Note:

- Key word "Bearer" needs to be exactly as shown, not "bearer" or "BEARER".
- For HTTPS access using browser or postman, make sure to use the IP address, not "localhost" and accept the security certificate for that IP address first by importing it into the browser.

## API Documentation and what is available ?

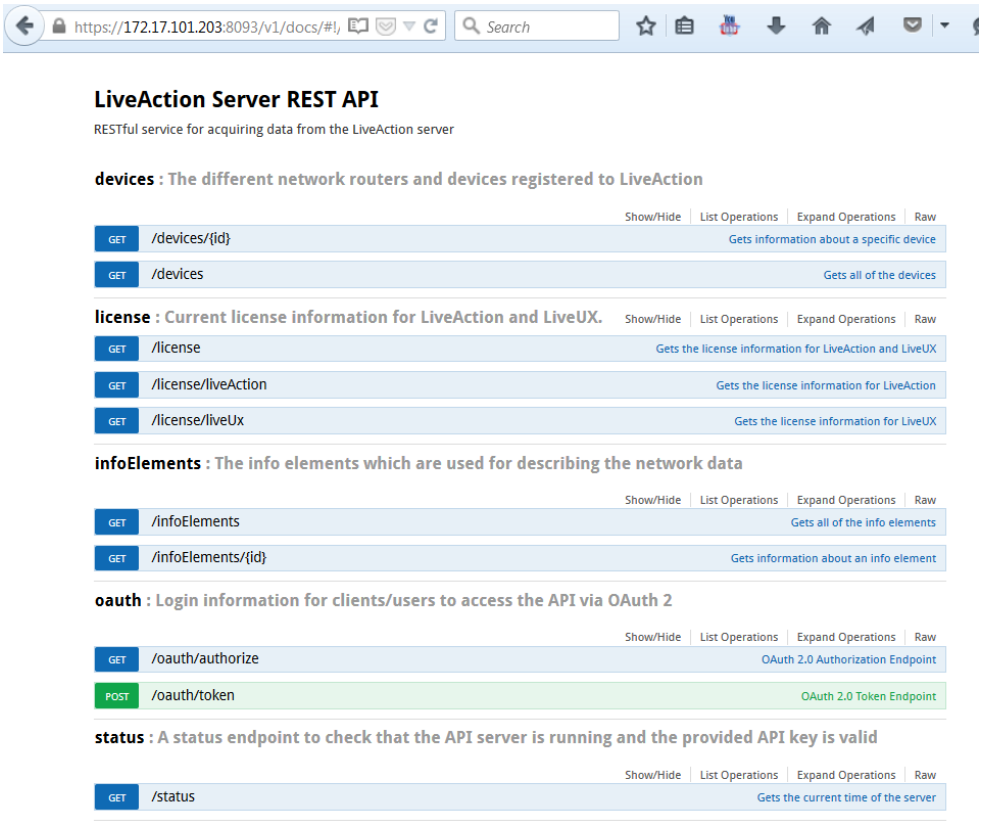
In this version (v1) of the API, the following features are available:

- Information about Devices
- Information about Flow Reports (Time Series and Summaries)
- Running Flow Reports (Time Series and Summaries)
- QoS Reports

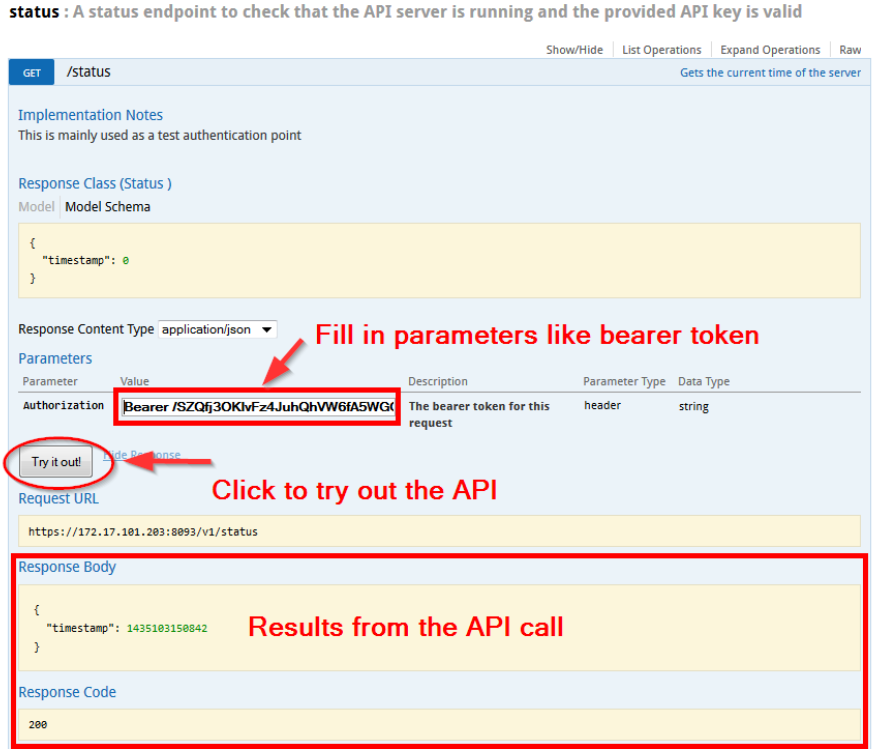
There is additional, detailed documentation about the different available resources here. The documentation may be found while the server is running by accessing the following URL (with the address/port changed to match the address/port of the server):

https://localhost:8093/v1/docs

The web page will have a listing of all the API calls that are available as shown below:




You can call the API directly from this web page to test out the calls. But make sure to accept the HTTPS SSL certificate using the IP address of the server, not localhost if doing it locally.



## How do I run a report and see the data that is behind the charts?

The equivalent of running a flow report can be done through the API using the runTimeSeries or runAggregation operation. The first step is for the user to determine which report he/she wants to run. The user is able to view all the possible flow reports that can be run by querying the "reports" resource for "flow" in the following way:

```
$ curl -k -H "Authorization: Bearer
yWwUxkoeI6LohcJd+yLlT+7KjZBAH7NpugFQ7SQTbKw="
"https://localhost:8093/v1/reports/flow/"
```

This will return a list of all the reports for flow data. Below is a short example of what the response may look like:

```
{
  "meta" : {
    "href" : "https://172.17.101.170:8093/v1/reports/flow",
    "http" : {
      "statusCode" : 200,
      "statusReason" : "OK"
    }
  },
  "aggregations" : [ {
    "href" : "https://172.17.101.170:8093/v1/reports/flow/0",
    "id" : "0",
    "name" : "Top Analysis",
    "defaultTechnologyType" : "basic",
    "technologyTypeOptions" : [ "basic", "medianet", "avc", "nsel", "pfr",
"wireless", "unknown" ]
  }, {
    "href" : "https://172.17.101.170:8093/v1/reports/flow/1",
    "id" : "1",
    "name" : "Top Conversations",
    "defaultTechnologyType" : "basic",
    "technologyTypeOptions" : [ "basic", "medianet", "avc", "nsel", "pfr",
"wireless", "unknown" ],
    "defaultSortBy" : "BIT_RATE",
    "sortByOptions" : [ "BYTE", "BIT_RATE", "IN_BYTE", "IN_BIT_RATE" ]
  },
  { … } ]
}
```

For example, running a particular time series aggregation for the "Application" report can be done with the following cURL command:

```
$ curl -k -H "Authorization: Bearer
yWwUxkoeI6LohcJd+yLlT+7KjZBAH7NpugFQ7SQTbKw="
"https://localhost:8093/v1/reports/flow/15/runTimeSeries"
```

The example response isn't shown here because of length, but more information can be found on the server API documentation page.

## Can I get a simpler version of that? How about a .csv file?

Absolutely. There is a "basic" view of the report which can be viewed by setting the optional parameter of the query parameter "view" to be the value of "basic." See the sample code below for an example of this feature.

```
$ curl -k -H "Authorization: Bearer
yWwUxkoeI6LohcJd+yLlT+7KjZBAH7NpugFQ7SQTbKw="
"https://localhost:8093/v1/reports/flow/15/runTimeSeries?view=basic"
```

There are also resources specific for generating .csv files. They can be accessed by adding the extension ".csv" to the existing resources.

```
$ curl -k -H "Authorization: Bearer
yWwUxkoeI6LohcJd+yLlT+7KjZBAH7NpugFQ7SQTbKw="
"https://localhost:8093/v1/reports/flow/15/runTimeSeries.csv"
```