# Test Drive the Analytics API

# Hands-on Lab

Presented by Sean Gubler
(sgubler@adobe.com)

# Lab Overview

The purpose of this lab session is to learn the value of use cases supported by the Adobe Analytics API. To accomplish this we will become familiar with API usage through hands on exercises. We will gain an appreciation for how simple the API is and how broadly across the Adobe Analytics platform it reaches.

# Exercise 1 – Make a simple API call using API Explorer

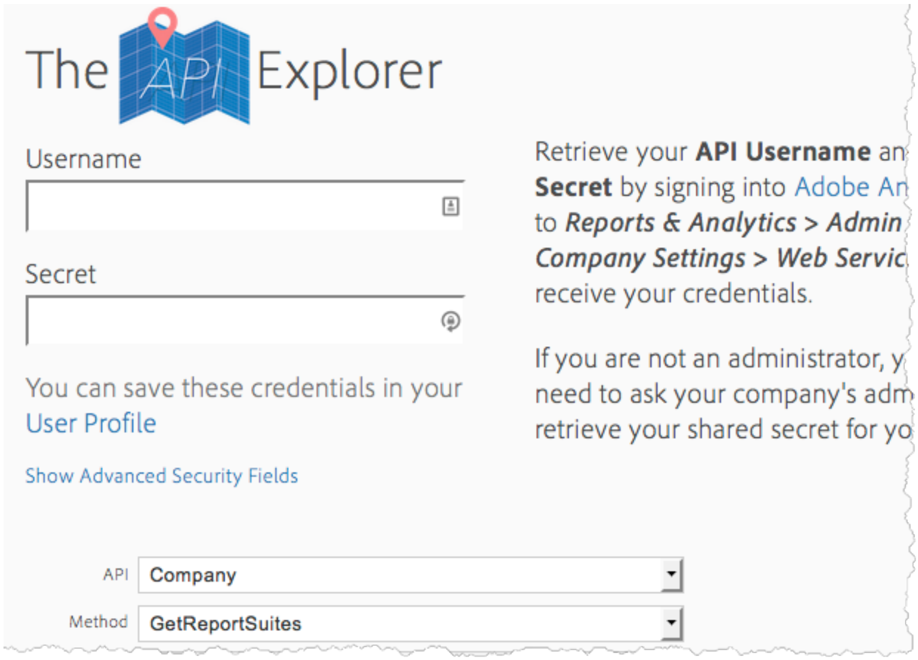
## Objectives

- Learn how the API explorer tool can be used to try API calls without writing any code

## Required Tasks

### Task 1 – Load the API explorer tool in your web browser

1. Open your browser and navigate to https://marketing.adobe.com/developer/api-explorer
2. Notice a page looking like the following.



Retrieve your **API Username** an
**Secret** by signing into Adobe An
to ***Reports & Analytics > Admin***
***Company Settings > Web Servic***
receive your credentials.

If you are not an administrator, y
need to ask your company's adm
retrieve your shared secret for yo

You can save these credentials in your User Profile

API Company

Method GetReportSuites

### Task 2 – Provide your API credentials

1. Supply the following values in the form
   a. Username: api_labs:Demo Inc
   b. Secret: 5b7b3a606e039b3de324aa8825db5713

### Task 3 – Select the API category and method

1. Set the following values in the form
   a. API: Company
   b. Method: GetReportSuites

### Task 4 – Modify the method parameters

1. In the **Request** text area paste the following empty JSON object

```
{}
```

To see the documentation for this method you can click the "Documentation" link in the API Explorer. Or for the full documentation page go here - https://marketing.adobe.com/developer/documentation/analytics-administration-1-4/r-getreportsuites-1

### Task 5 – Execute the call and analyze the response

1. Execute the API call be clicking the **Get Response** button
2. Look at the **Response** text area to see the result, which should be a JSON object containing a list of ALL standard report suites in the **Demo Inc** company.

### Task 6 – [Optional]  Filtering the list

1. How would you modify the parameters so that only the "Apple IPhone" report suite is returned?
   a. Modify the JSON parameters and run the query again.

# Exercise 2 – Create a new User

## Objectives

- Learn how create a new Analytics User by calling Permissions.AddLogin

## Required Tasks

### Task 1 – Navigate to API Explorer in your web browser

1. Open your browser and navigate to https://marketing.adobe.com/developer/api-explorer

### Task 2 – Provide your API credentials

1. Supply the following values in the form
   a. Username:   api_labs:Demo Inc
   b. Secret:    5b7b3a606e039b3de324aa8825db5713

### Task 3 – Select the correct API and Method

1. Set the following values in the form
    a. API: Permissions
    b. Method: AddLogin

Task 4 – Modify the parameters object and click **Get Response**

1. Modify the values in blue with your own information
    a. The “X” in “userX” (the login) should be replaced with your station number, example “user5”

```
{
      "create_dashboards":false,
      "email":"firstname_lastname@domain.com",
      "first_name":"firstname",
      "group_names":["All Report Access", "Current Data Users"],
      "is_admin":false,
      "is_temp":false,
      "last_name":"lastname",
      "login":"userX",
      "must_change_password":false,
      "password":"abc123",
      "phone_number":"",
      "rsid":"gubler.summit.lab1",
      "temp_end_date":"",
      "temp_start_date":"",
      "title":"some_title"
}
```

2. Click the Get Response button
3. Notice that “true” is returned in the response text area

Task 5 – Login to the UI with your new user.

1. Browse to https://my.omniture.com/login/
2. Login with the following
    a. Company: Demo Inc
    b. Username: userX
    c. Password: abc123
3. Notice that you are able to log in and look at reporting for various report suites. This is because you added this user to the group “All Report Access”

Question: Could this new user be used to make API calls?
Answer: NO! For that the “Web Service Access” group is needed in the “group_names” array.

# Exercise 3 – Get a list of Segments using a simple Java client (OAuth2)

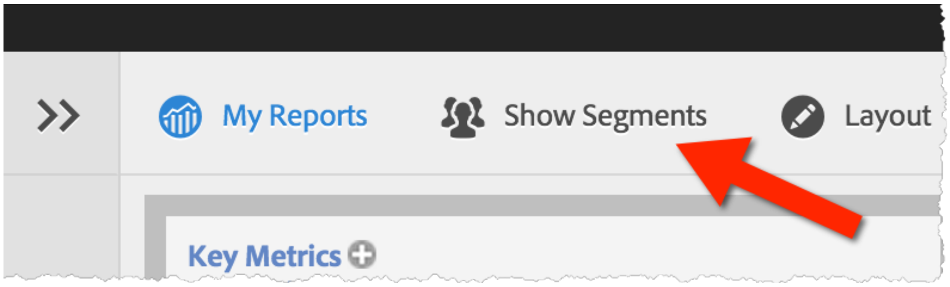
## Objectives

- Learn how pull out a list of Segments from Adobe Analytics using the Segments API

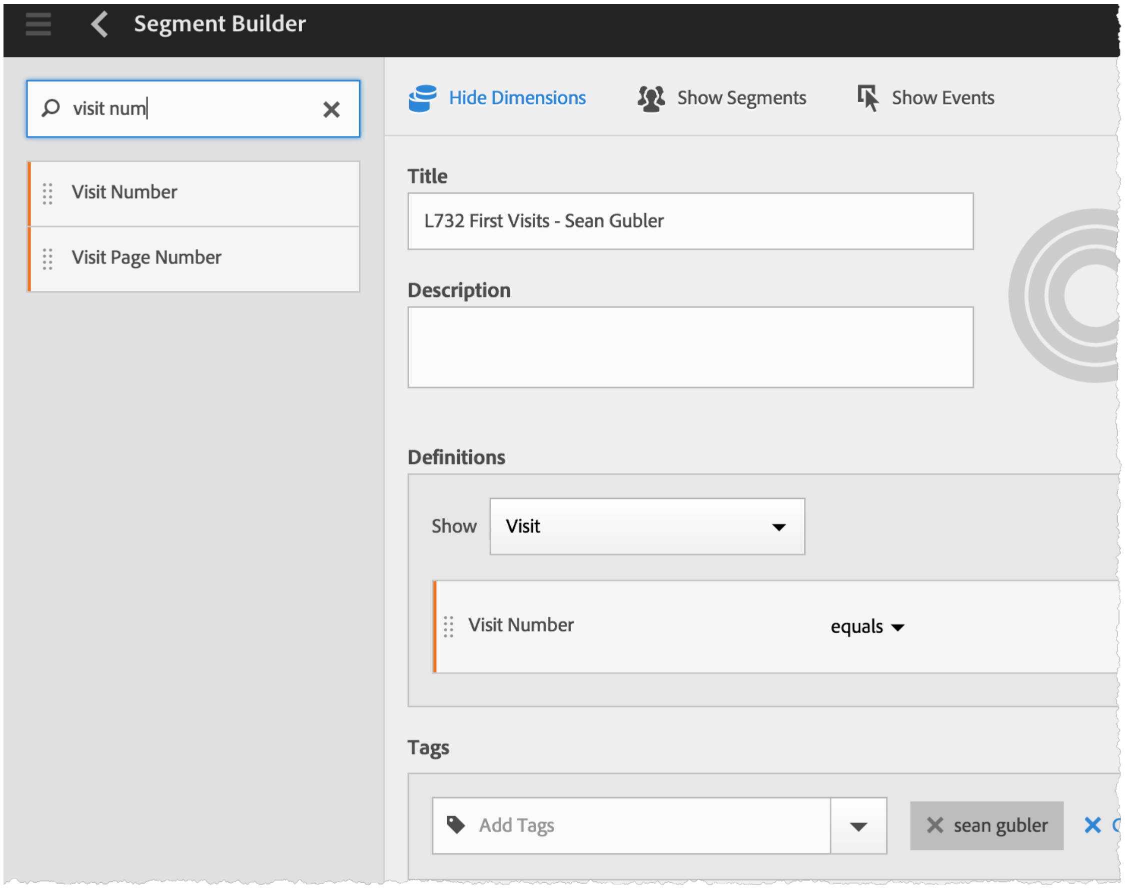
## Required Tasks

Task 1 – Create a new Segment in the Analytics UI

1. Login at https://my.omniture.com/login/ -- *you should already be logged in from the previous exercise*
2. Click on **Show Segments**



3. Click **Create new segment**
4. Create a new segment by filling in the form as follows
   a. Title: *L732 First Visits – FirstName LastName*
   b. Description: (leave blank)
   c. Definitions:
      i. Show: Visit
      ii. Rule: Visit Number equals 1 (hint: use the search to find Visit Number)
   d. Tags: FirstName LastName



Search: visit num

- Visit Number
- Visit Page Number

Title: L732 First Visits - Sean Gubler

Description:

Definitions

Show: Visit

Visit Number equals

Tags: sean gubler

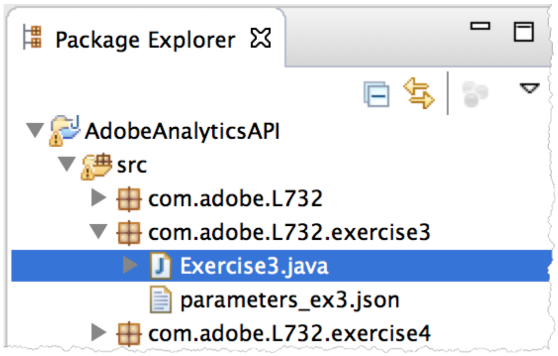
5. Click **Save**

Task 2 – Review with the instructor on how to create an “Application” at the Developer Connection. https://marketing.adobe.com/developer/applications

Task 3 – Open the Java Lab exercises

1. Launch Eclipse – there is an icon in the Dock at the bottom of your screen

2. In Package Explorer, expand the AdobeAnalyticsAPI project until you see Exercise3.java


Package Explorer

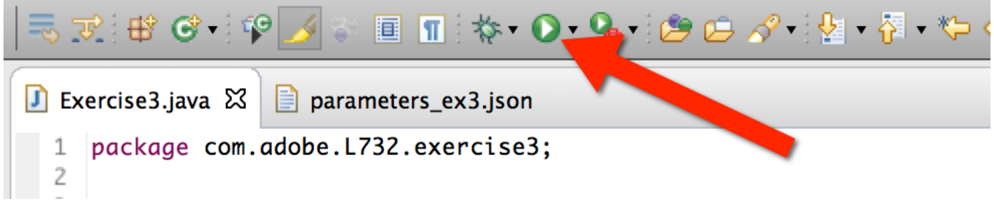
- AdobeAnalyticsAPI
  - src
    - com.adobe.L732
    - com.adobe.L732.exercise3
      - Exercise3.java
      - parameters_ex3.json
    - com.adobe.L732.exercise4

3. Open both Exercise3.java and parameters_ex3.json  (double click them)
4. Also, open the Charles application

To view the full details of the API request and response you can open up the **Charles** proxy application – the icon looks like your grandma’s lemoade pitcher in the Dock. The Java exercises are already set up to proxy the API traffic through Charles as long as DEBUG_PROXY variable is set to true.

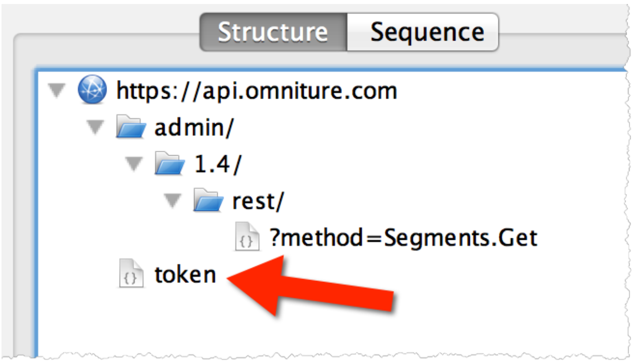
Task 4 – Run the script

1. View the parameters_ex3.json file. These are the parameters that will be sent to the API for this Segments.Get call
2. Run the script by clicking the Run icon


Exercise3.java  parameters_ex3.json

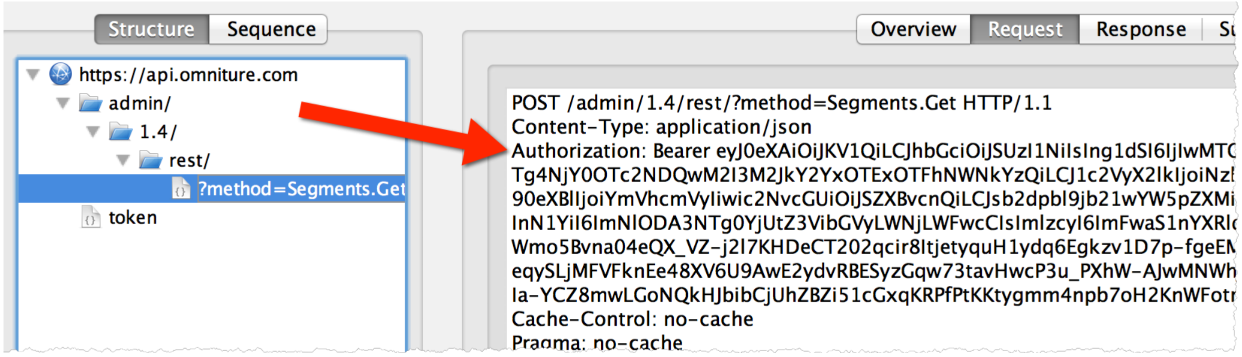
```
package com.adobe.L732.exercise3;
```

3. Notice the API response in the Console at the bottom. A list of Segments should be shown.
4. In Charles, because this was OAuth2 authentication, you will see that a separate “token” request was made.



- https://api.omniture.com
  - admin/
    - 1.4/
      - rest/
        - ?method=Segments.Get
  - token

5. In Charles you will also see the Authorization Header in the Segments.Get request



```
POST /admin/1.4/rest/?method=Segments.Get HTTP/1.1
Content-Type: application/json
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dSI6IjIwMT
Tg4NjY0OTc2NDQwM2I3M2JkY2YxOTExOTFhNWNkYzQiLCJ1c2VyX2lkIjoiNz
90eXBlIjoiYmVhcmVyIiwic2NvcGUiOiJSZXBvcnQiLCJsb2dpbl9jb21wYW5pZXMi
InN1YiI6ImNlODA3NTg0YjUtZ3VibGVyLWNjLWFwcCIsImlzcyI6ImFwaS1nYXRl
Wmo5Bvna04eQX_VZ-j2l7KHDeCT202qcir8ltjetyquH1ydq6Egkzv1D7p-fgeE
eqySLjMFVFknEe48XV6U9AwE2ydvRBESyzGqw73tavHwcP3u_PXhW-AJwMNWh
la-YCZ8mwLGoNQkHJbibCjUhZBZi51cGxqKRPfPtKKtygmm4npb7oH2KnWFot
Cache-Control: no-cache
Pragma: no-cache
```

Task 5 – [Optional] Get more details about the Segments

1. Alter the parameters_ex3.json file so that more than just the "id" and "name" fields are returned. Then run the application again. You may need to look at the documentation - https://marketing.adobe.com/developer/documentation/segments-1-4/r-get-1

# Exercise 4 – Reporting API for a custom visualization

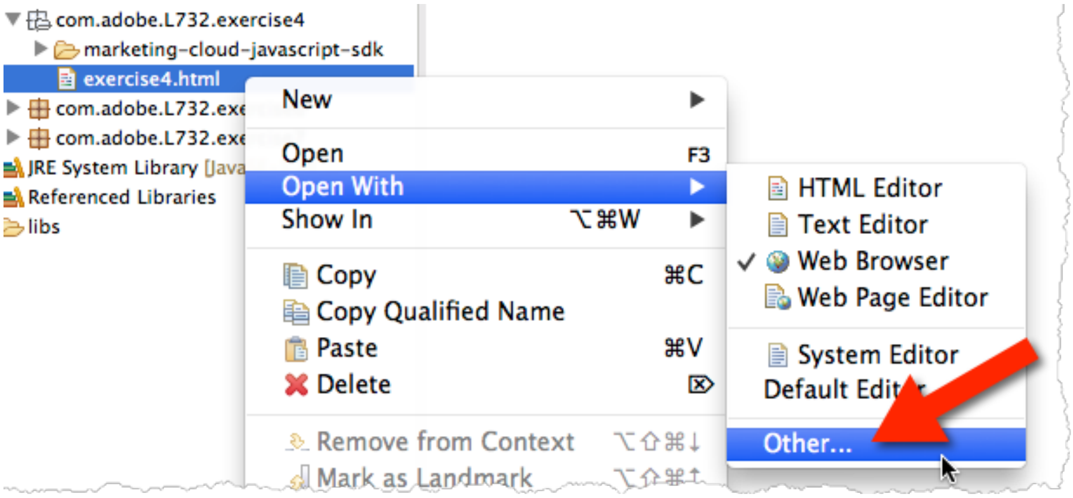
## Objectives

- Learn how pull reports from Adobe Analytics to power a custom reporting visualization

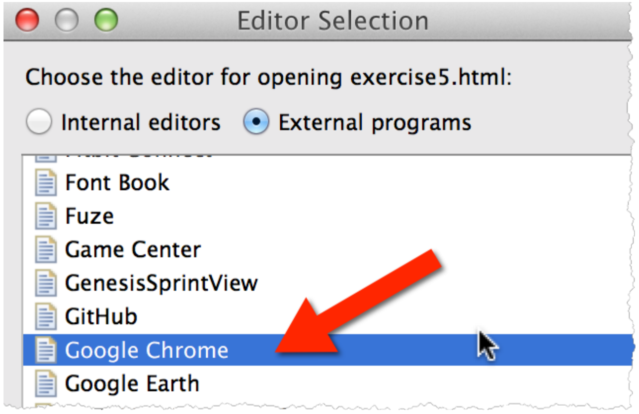
## Required Tasks

Task 1 – Create the reportDescription that will populate the chart with Visit data from the year 2014

1. In Eclipse, close the files in the editor pane from the previous exercises.
2. Expand the exercise4 package and open **exercise4.html** in the editor (double click it)
3. Fill out the properties of the reportDescription object (line 20) so that daily visits totals are retrieved from reportSuiteId = `gubler.jjesquiredev`
   a. ReportDescription documentation - https://marketing.adobe.com/developer/documentation/analytics-reporting-1-4/r-reportdescription-1
   b. If you give up the answer can be seen in the Answers section of this guide
4. Run the report by opening the html file in a web browser
   a. Right click on exercise4.html in the Package Explorer
   b. In the menu choose **Open With > Other...**



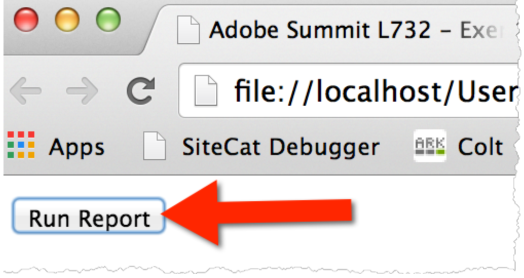
c. Then select **External programs** and **Google Chrome**



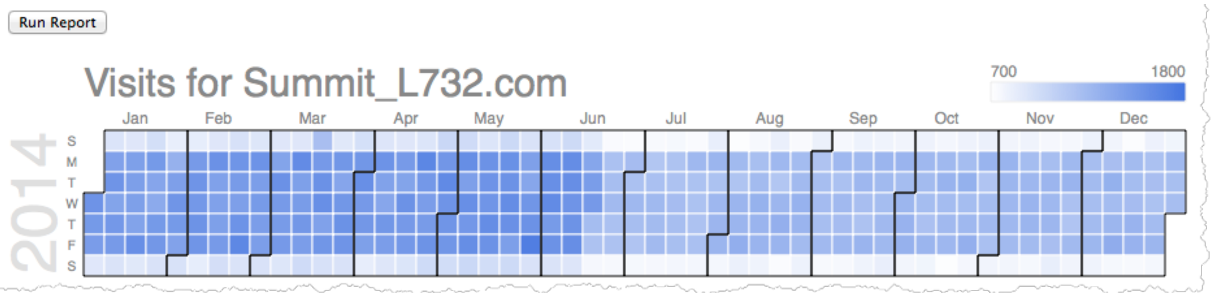
Choose the editor for opening exercise5.html:

○ Internal editors ● External programs

d. Once open in the browser click the **Run Report** button



e. The report should look like the following -



f. IMPORTANT - If you need to make changes to your reportDescription be sure to **Refresh** the browser before attempting to run the report again

5. [Optional] Adjust the reportDescription so that only the first time visits are reported. Hint: use the segment you created in Exercise 3.

> In Google Chrome you may want to use the Developer Tools to debug. On the menu bar go to **View > Developer > Developer Tools**. The Network and Console panels should be the most helpful.

# Exercise 5 – Sending live data to Adobe Analytics

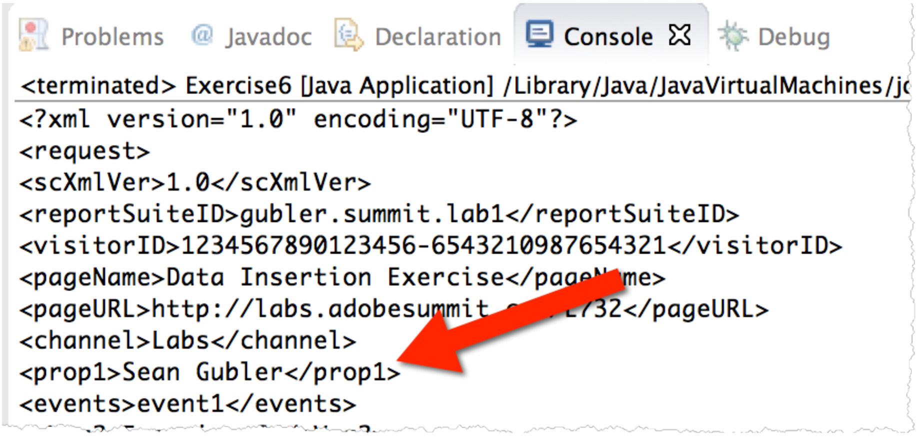
## Objectives

- Learn how to use the Data Insertion API to send live (hit level) data to a report suite

## Required Tasks

Task 1 – Edit the script to send in a PageView beacon

1. In Eclipse, close the files in the editor pane from the previous exercises
2. Expand the exercise5 package and open **Exercise5.java** in the editor (double click it)
3. Edit the value of "prop1" (line 28) so that it contains your full name
4. Run the script by clicking on the Run icon
5. Look in the Console to see the XML that was sent to the Analytics server. Your name should be there in the **prop1** element

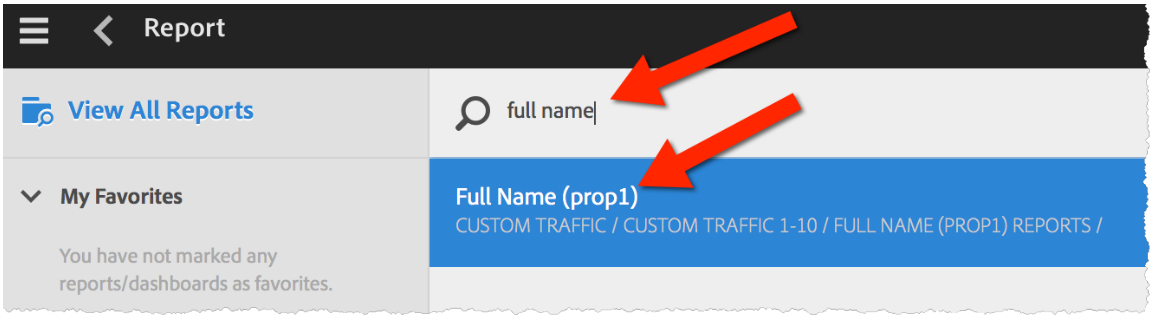


```
<terminated> Exercise6 [Java Application] /Library/Java/JavaVirtualMachines/jc
<?xml version="1.0" encoding="UTF-8"?>
<request>
<scXmlVer>1.0</scXmlVer>
<reportSuiteID>gubler.summit.lab1</reportSuiteID>
<visitorID>1234567890123456-6543210987654321</visitorID>
<pageName>Data Insertion Exercise</pageName>
<pageURL>http://labs.adobesummit.com/L732</pageURL>
<channel>Labs</channel>
<prop1>Sean Gubler</prop1>
<events>event1</events>
```

6. Also notice that a <status>SUCCESS</status> was returned by the server

Task 2 – Verify by looking in the reporting UI

1. Log in to the Adobe Analytics UI. Refer to exercise 2 for login instructions.
2. Select the **Summit – Lab 1** report suite from the Report Suite selector
3. Click **View All Reports** to expand the menu of reports
4. Type "full name" in the Search bar
5. Select the Full Name (prop1) report

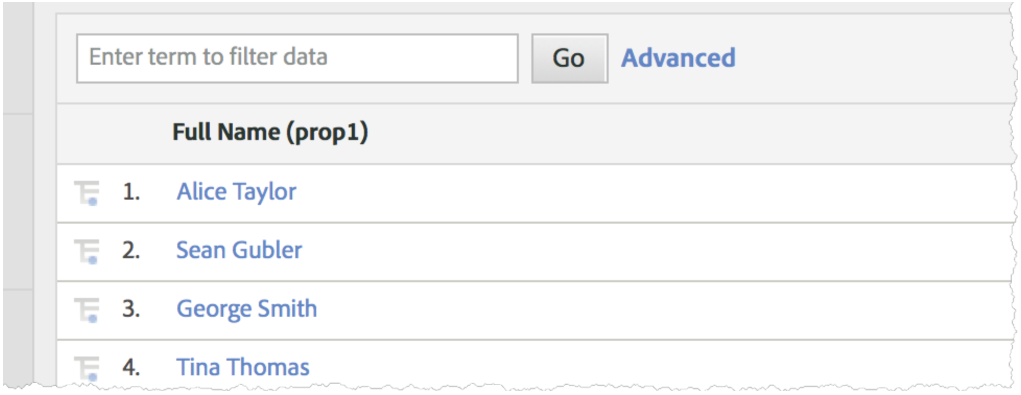


full name

Full Name (prop1)
CUSTOM TRAFFIC / CUSTOM TRAFFIC 1-10 / FULL NAME (PROP1) REPORTS /

You have not marked any reports/dashboards as favorites.

6. Select the current day in the Date range selector
7. Find your name in the report



| | Full Name (prop1) |
|---|---|
| 1. | Alice Taylor |
| 2. | Sean Gubler |
| 3. | George Smith |
| 4. | Tina Thomas |

When sending in hit level data using Data Insertion API it is important to get the Visitor Identification right. This is especially important if some hits of a visit are collected through the web browser and some are sent from a server using Data Insertaion API. Read more here - https://marketing.adobe.com/developer/documentation/data-insertion/c-visitor-id

# Exercise 6 – Processing an Analytics LiveStream

## Objectives

- Learn how to connect to a LiveStream and process individual hits

## Required Tasks

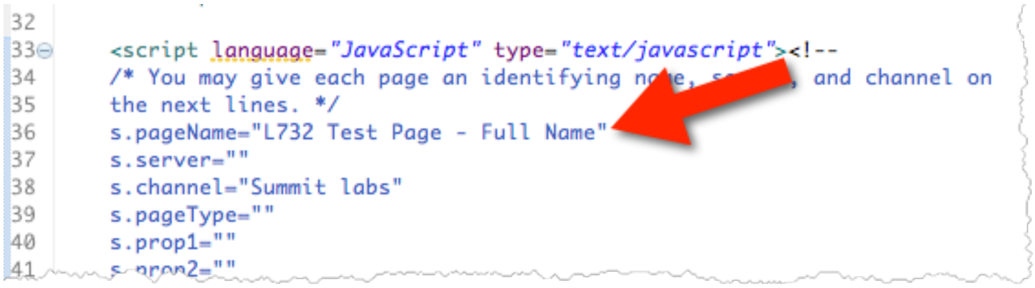
### Task 1 – Run the script to connect to the LiveStream

1. In Eclipse, close the files in the editor pane from the previous exercises
2. Expand the exercise6 package and open **Exercise6.java** in the editor (double click it)
3. Set the URL for the stream on line 25
    a. Use -- https://sjo.livestream.adobe.net/api/1/stream/summit732-X
    b. replace X with your station number
4. Set the API Token on line 28 – you can get this by running Exercise 3 again
5. Run the script by clicking on the Run icon
6. Notice that the Console shows "Connected to liveStream…"

Analytics LiveStreams must be configured by an Adobe representative before you can connect. See the details of requesting a LiveStream here… https://marketing.adobe.com/developer/documentation/analytics-live-stream/get-started-1

### Task 2 – Generate Analytics beacons to be observed through the Live Stream

1. In Eclipse open the page.html file in the html editor
2. Modify the pageName value (line 36) by appending your full name to it

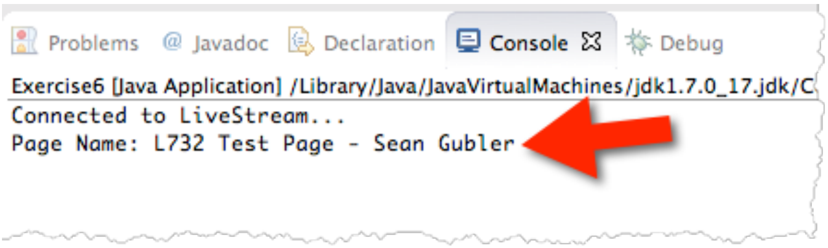


```
32
33    <script language="JavaScript" type="text/javascript"><!--
34    /* You may give each page an identifying name, server, and channel on
35    the next lines. */
36    s.pageName="L732 Test Page - Full Name"
37    s.server=""
38    s.channel="Summit labs"
39    s.pageType=""
40    s.prop1=""
41    s.prop2=""
```

3. Open **page.html** in an external browser such as Google Chrome
    a. Right click > Open With > Other > External Program > Google Chrome
4. Use the DigitalPulse debugger bookmark to be sure that an Analytics beacon was generated on the page

### Task 3 – Validate the active script connected to LiveStream has processed your beacon

1. In Eclipse look at the Console
2. Notice the lines where your name appended to the pageName. You may want to refresh the page a few times



```
Exercise6 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_17.jdk/C
Connected to LiveStream...
Page Name: L732 Test Page - Sean Gubler
```

Task 4 – [Optional] Modify the script so that it sends you a text message on a Purchase event

1. Stop the script by clicking the stop button (a red square)
2. In Exercise6.java modify the **notifyAddress** value (line 71) by providing your 10 digit phone number @ your mobile carrier. Example, 8015551234@txt.att.net
    a. Your mobile provider's email address format may be found here… http://www.emailtextmessages.com/
3. After saving the file, start the script again
4. The instructor will create a purchase event to trigger the notification

# Answers

## Exercise 1

[Optional] Company.GetReportSuites parameters to filter for only a specific report suite.

```
{
        "search":"Apple iPhone"
}
```

## Exercise 3

[Optional] Returning additional fields with the Segments.Get method.

```
{
        "accessLevel":"all",
        "fields":[
                "tags",
                "owner",
                "modified",
                "favorite"
        ]
}
```

[Optional] To filter for only your segment you'll need to use the filter attribute, either specifying by the "name" or "tags"

```
{
        "accessLevel":"all",
        "fields":[
                "tags",
                "owner",
                "modified",
                "favorite"
        ],
        "filter":{"name":"Full Name"}

}
```

## Exercise 4

The reportDescription for pulling a report the correct visits report.

```
"reportDescription":{
      "reportSuiteID":"gubler.jjesquiredev",
      "dateFrom":"2014-01-01",
      "dateTo":"2014-12-31",
      "dateGranularity":"day",
      "metrics":[{"id":"visits"}]
}
```

[Optional] With segment applied, note that the actual segment ID may vary.

```
"reportDescription":{
      "reportSuiteID":"gubler.jjesquiredev",
      "dateFrom":"2014-01-01",
      "dateTo":"2014-12-31",
      "dateGranularity":"day",
      "segments":[
            { "id":"54dd25d5e4b05c80944cd79f" }
      ],
      "metrics":[{"id":"visits"}]
}
```