



EZMCOM User Behavior Analytics Web JavaScript Client Integration

Document Number
EZM_UBAWJS_00

Issue Date
25 Jul 2016

Version 1.0.6

Prepared for
EZMCOM Software Product Integration Developer

Prepared by

EZMCOM Inc.
4701 PATRICK HENRY DR, SANTA CLARA, CA, 950541863, US.
Tel: +1 510 396 3894
+60 (0)12 570 1114
Email: info@ezmcom.com

Copyright © 2016 by EZMCOM

This work is copyright. Other than as permitted by law, no part of this document may be reproduced, stored in a retrieval system or transmitted in any form or by any process without prior written permission.

User Behavior Analytics Client Integration

Document Scope And Intended Audience

The intended audience of this document is HTML/web developers who wish to integrate User Behavior Analytics script into web application page. The User Behavior Analytics REST API specification would also be outlined in this document.

The document will cover the steps of Javascript integration into the said HTML page, then the details of REST API integration.

It is assumed that the intended audience should has basic knowledge on HTML/Javascript coding and adequate skill to integrate REST API.

UBA Javascript Integration Steps

This section describes how to integrate User Behavior Analytics JavaScript into the login page.

1. Include the EzUBAnalytics script file in your login HTML. For an example:

```
<!-- Include the EZMCOM UBA script file -->  
<script src="./js/ezuba.min.js"></script>
```

NOTE: Please change the highlighted path accordingly based on your web application directory structure and the location of the script file.

2. Initialize the script in your Javascript tag. For an example:

```
<script type="text/javascript">  
  // @param offsetFlag   Optional. Only if provide true, it will offset the  
  //                    char code.  
  // @param sessionId   Optional. If not provided or provide null,  
  //                    it creates internal session Id.  
  // EzUBAnalytics.init();  
  EzUBAnalytics.init();  
</script>
```

It is a must to call this to initialize the EzUBAnalytics script engine.

3. Bind the input field DOM element. Call this to bind and register the EzUBAnalytics on textbox that you wish to have UBA enabled. For an example:

```
<script type="text/javascript">  
  // @param element     Mandatory. It is the DOM element of textbox  
  //                    for which to run UBA evaluation.  
  // EzUBAnalytics.bind(element);  
  EzUBAnalytics.bind(document.getElementById("inputPassword"));  
</script>
```

NOTE: Please change the highlighted value accordingly, as per your password input HTML field id.

4. Capture the analytics data. Call this when you ready to capture the UBA JSON data. For an example: upon login form submit or on click event of login form Submit button.

```
<script type="text/javascript">
// @param elEva      Mandatory. It is the DOM element of textbox
//                  for which you would like to extract it UBA JSON
//                  data.
// @param elDes      Optional. It is the DOM element of textbox or
//                  textarea for which you would like to output the
//                  UBA JSON data. If not provided or provide null.
//                  The result will be returned from this function only.
// EzUBAnalytics.ubaJson(elEva, elDes);
EzUBAnalytics.ubaJson(document.getElementById("inputPassword "),
document.getElementById("inputUBAData"));
</script>
```

NOTE: Please change the highlighted values accordingly, as per your password input HTML field id, and the hidden input field id, which to store the UBA JSON data.

5. You will need to add code to send the data in inputUBAData (from the js) and extract it in the server side (most likely as a hidden form field), this step requires customization and varies based on the technology stack being used.

UBA REST Integration

Previous section describes how to integrate the UBA javascript to collect UBA JSON data from HTML front-end. This section would then describe the details of UBA REST API specification, which serves for the purpose of UBA training and scoring services.

IMPORTANT NOTE: The UBA web service must NOT be called from the browser. If it is being applied to a password field it must be called only after the password is successfully validated.

UBA Scoring REST API

Name	ScoreAndUpdate	
HTTP Method	POST	
Hostname Context	https://<HOSTNAME>/uba-webservice	
URL	/uba/v1/apiKey/{apiKey}/accounts/{accountName}/profiles/{profileName}/scoreAndUpdate	
	apiKey	The api key provided by EZMCOM
	accountName	Unique username for the user
	profileName	Unique profile associated with the user, there must be a single unique profile name for each string that needs to be evaluated by UBA. Recommend to use the HTML input field ID for this value.
Query Parameter	operation	Optional. Specify the type of the UBA operation. Possible values: <ul style="list-style-type: none"> SCORE - Will treat the request as readonly, unless the score is greater than the threshold.
HTTP Headers	Content-Type	application/json
	Accept	application/json
Body	The data generated in the Javascript portion that is extracted on the server side	

NOTE:

The query parameter for "operation" is useful if the UBA data is deferred update. Introduce this query parameter in the URL, will instruct the UBA service to ONLY evaluating the data for producing the training/scoring info, but NOT updating the data. Take for an example, the application will challenge 2FA when the UBA is in training. Thus, on every login, the UBA data should be deferred update, which means, the UBA data should only updated into UBA service after the 2FA succeeded. The following illustrate the order of REST API URL to be called for this scenario:

1. Call UBA for score only but NOT update. URL example (with query parameter):
 /uba/v1/apiKey/{apiKey}/accounts/{accountName}/profiles/{profileName}/scoreAndUpdate?operation=SCORE
2. If UBA is in training, then challenge user for 2FA.
3. Only if 2FA succeeded, proceed to call UBA again for update, with the identical UBA JSON data in body. URL example (without query parameter):
 /uba/v1/apiKey/{apiKey}/accounts/{accountName}/profiles/{profileName}/scoreAndUpdate

Sample request body (this is the uba data generated in the js)

```
{
  "v": 1,
  "ts": 1442459911036,
  "fn": 0,
  "sid": "fo83qbty9tqpvicljbailfzplj714imn",
  "d": [[0, 1442459897696, 70], [1, 1442459897807, 70], [0, 1442459897880, 79], [1, 1442459897968, 79], [0, 1442459898072, 79], [1, 1442459898176, 79], [0, 1442459898203, 66], [1, 1442459898312,
```

```
66], [0, 1442459898376, 65], [0, 1442459898529, 82], [1,
1442459898568, 65], [1, 1442459898656, 82]],
  "h":
  "c3ab8ff13720e8ad9047dd39466b3c8974e592c2fa383d4a3960714caef0c4f4",
  "ua": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.85
Safari/537.36"
}
```

Sample response body (Training)

```
{
  "transactionId": "616c8725-ea0e-4d29-938e-0cd98fa0ec05",
  "timestamp": 1449651691582,
  "date": "2015-12-09T09:01Z",
  "additional_elements": {
    "score": "0.0",
    "training": "0",
    "attempt": "1"
  }
}
```

Sample response body (Training Completed)

```
{
  "transactionId": "15d96f87-0bd4-4aec-b0cf-1049bac3a91f",
  "timestamp": 1449651691582,
  "date": "2015-12-09T09:01Z",
  "additional_elements": {
    "score": "0.9972375690607737",
    "consecutiveFailureCount": "0",
    "training": "1",
    "threshold": "0.9962597105000737",
  }
}
```

Response fields	
transactionId	Unique transaction identifier
timestamp	Epoch time in milliseconds
date	Human readable date (yyyy-MM-dd'T'HH:mmX)
score	Calculated confidence score ranging from 0 (low confidence) to 1 (high confidence)
consecutiveFailureCount	The consecutive count of score falling below the threshold. This counter will reset to 0 once the user gets a score higher than the threshold.
training	0 indicates the user profile is in training 1 indicates the user profile has been trained
threshold	This value is calculated based on the user's training. Attempts scoring below this value should be considered as unlikely originating from the same user who performed the training.

Additional REST APIs

1. Reset UBA profile

To reset a particular user's UBA profile. Resetting of the UBA profile would get the user back to the training mode. This is useful when the integrated application would like to force an enrolled user back to the training mode again.

Http Request:

Name	reset	
HTTP Method	POST	
Hostname Context	https://<HOSTNAME>/uba-webservice	
URL	/uba/v1/apiKey/{apiKey}/accounts/{accountName}/profiles/{profileName}/reset	
	apiKey	The api key provided by EZMCOM
	accountName	Unique username for the user
	profileName	Unique profile associated with the user, there must be a single unique profile name for each string that needs to be evaluated by UBA. Recommend to use the HTML input field ID for this value.
HTTP Headers	Content-Type	application/json
	Accept	application/json
Body	Not required.	

Http Response Example:

Response Header:

```
HTTP/1.1 200 OK
Content-Type:application/json
```

Response Body:

```
{
  "transactionId": "616c8725-ea0e-4d29-938e-0cd98fa0ec05",
  "timestamp": 1449651691582,
  "date": "2015-12-09T09:01Z"
}
```

