DATA IMPORT

This entry provides detailed information on the Data Import processes of the DMP

INTRODUCTION

One of the most important processes of the DMP is definitely importing data. In order to successfully import (custom) data into the Mapp DMP you will have to be aware of certain formats. All information and guidelines regarding data import.

USE CASES

Below a couple of use cases for the Data Import API feature:

- Profile Enrichment:
  Enrich the DMP pixel data by importing customer specific data into the DMP which may be stored in an external data source like a CRM system or an internal data warehouse.
- Data Layer:
  Benefit from an existing data layer implementation by sending specific data attributes from the external system to the DMP in real-time.

API SERVICE

The Mapp DMP includes an API endpoint to pass data to the DMP. By means of this service the DMP has the capability to accept any custom data that is sent in a JSON format through the external_data parameter (&data=). There are two ways to send the data; by using a GET request and also by using POST request. In cases of very large requests (i.e. more than 2k characters) - POST request is preferred. The correct response is 200, or 302. If you get 40x, it means that one of the parameters is not correct. Consider the request template and the parameter description below as a basis for building out a server to server (S2S) integration.

TEMPLATE

http://go.flx1.com/dp?m=customer_id&id=pixel_id&t=js&-data=url_encoded_JSON

Parameters	
m	This should contain the static customer_id relevant to your DMP account
id	This should contain the relevant pixel_id. You can get it by creating a new pixel in the DMP.
t	This should contain the endpoint (type) to return empty JavaScript or a 1x1 transparent GIF. Two values are possible: js or gif (default = js).
data	This should contain the URL encoded JSON string. Note that there is no limit to complexity of the JSON object since it can contain nested objects and arrays. If you are using a POST request, you can omit data parameter and send the JSON string (not URL encoded) as a body of the request.

The parameters as stated in the template can be considered as required parameters. In addition there are many other optional parameters that can be populated:

Optional Parameters	
&_check=1	This disables cookie syncing
&_nr=1	This disables a check for cookie stickyness
&	& A full list of available parameters can be found here

EXAMPLE

Please find below a specific example of custom data in JSON format and the corresponding GET request that will be send to the DMP.

JSON Object

```
{
 “Timestamp”: 1456213019,
 “UserId”: 4694200,
 “FirstName”: “John”,
 “LastName”: “Doe”,
 “Gender”: “1”,
 “Favorites”: [“Football”, “Tennis”, “Poker”],
 “TotalBets”: 56,
 “TotalDeposits”: 14,
 “CustomerLifetimeValue”: 7.0
}
```

URL Encoded JSON Object

```
%7B%0A%09%22Timestamp%22%3A%20
1456213019%2C%0A%09%22UserId%22%3A%204694200
%2C%0A%09%22FirstName%22%3A%20
%22John%22%2C%0A%09%22LastName%22%3A%20%22Do
e%22%2C%0A%09%22Gender%22%3A%20
%221%22%2C%0A%09%22Favorites%22%3A%20%5B%22
Football%22%2C%20%22Tennis%22%2C%20%22Pok-
er%22%5D%2C%0A%09%22TotalBets%22%
3A%2056%2C%0A%09%22TotalDeposits%22%3A%20
14%2C%0A%09%22CustomerLifetimeVal
ue%22%3A%207.0%0A%7D
```

Assembled URL

https://go.flx1.com/dp?m=100&id=10805&t=js&data=%7B%0A%09%22Timestamp%22%3A%201456213019%2C%0A%09%22UserId%22%3A%204694200%2C%0A%09%22FirstName%22%3A%20åå%22Doe%22%2C%0A%09%22Gender%22%3A%20%221%22%2C%0A%09%22Favorites%22%3A%20%5B%22Football%22%2C%20%22Tennis%22%2C%20%22Poker%22%5D%2C%0A%09%22TotalBets%22%3A%2056%2C%0A%09%22TotalDeposits%22%3A%2014%2C%0A%09%22CustomerLifetimeValue%22%3A%207.0%0A%7D

TIPS & TRICKS

Since the example only contains custom data for one specific customer, multiple requests needs to be send. In order to make this process as efficient as possible there are some tips and tricks applicable on both the data format and data upload.

Data Format

- Use integers and doubles rather than strings.
- Use unix timestamp for date and time elements.
- Ignore empty attributes.

Data Upload

- Upload in parallel, using http keep alive (re-using the open TCP connections).
- Make sure to not cache DNS so you can send data to all of our APIs.
- Do not use https for uploading.

UUID Parameter
The custom data request can also contain the &uuid parameter in case a customer wants to do the mapping on their end. This way you can send the relevant Mapp User ID along with the enriched data and the custom data attributes becomes immediately available for Audience Building purposes.

ACTIVATION

Once the implementation and the data upload are completed, the custom attributes can be enabled via the External Attributes API service for audience building purposes in the DMP interface.