Implementing Dental Insurance XML for Campaign {{CID}}

Overview

This is a Restful Web service created with the XML implementation, this web service allows to the get listings from the Vantage Media System and be consumed in XML format. This implementation restricts the pass of parameters in the URL query string. There is no limit on the number of parameters.

Overview

To use this service you need to do a HTTP REQUEST (**POST** type) to this URL:

http://marketplaces.vantagemedia.com/Search

The request Body should contain a structure of data as follows:

{"Campaign":"String","Zipcode":"String"}

Sample with Values:

{"Campaign":"{{CAMPAIGN}}","Zipcode":"{{ZIPCODE}}"}

Example of request:

http://marketplaces.vantagemedia.com/Search?state=CA&zipcode=90210&campaign=6604&engagementoption=4&md=

Query String Parameters

Parameters indicated with a "++" are mandatory. This information belongs to the client who is searching in the service consumed and MUST be passed in the query string parameters of the Request:

++ZipCode = Use this parameter to pass the location of the user making the query. This can only be the zip code in its 5 digit format.(**e.g., "zipcode": 90210**)

publisherId = This is your Publisher ID, and it will be assigned when you register on VantageMedia's Platform. Your Publisher ID is (**{{PUBLISHERID}}**)

++campaign = Campaigns are used to track separate implementations, placements or traffic sources. The Campaign ID for this implementation is ({{CAMPAIGN}})

maxresults = Indicates the max number of results intended for the visitor to see.

++ md = Marketplace definition for your private marketplace that identifies the vertical. The MD for this implementation is ({{MD}})

++state (required) = Specify state marketplace to use. Valid values are two character state codes. (e.g. state=CA, state=AK)

EngagementOption= this filter will allow you to obtain different types of engagements: clicks, calls or both. When not specified, it will default to clicks. See **Table 1** for Reference.

client_user_agent= the software agent for an HTTP request, for instance in a browser, will indicate version of the browser, platform (i.e., mobile), etc.

client_referrer= the HTTP header that identifies the address of the webpage.

client_ip= the IP address that accessed the website.

Sourcing and Tracking

To enable traffic sourcing the script has 2 parameters that you can use:

adSource= This receives strings of up to 24 alphanumeric characters (letters and numbers), you may also use underscore "_" and dash "-" characters. The purpose of the Adsource is to tag the source of the traffic, for example SEM, SEO, Display, Google, Yahoo, etc.

PublisherTrackingId = Similar to the previously explained parameter, this one will allow you up to 32 alphanumeric characters, underscore and dash. This tracking parameter can be used to track individual users with values like USER1234567, or any encoded ID you wish to pass.

To include it on your script, simply add the parameters into the parameter definition section of the code, as explained below.

No Index/ No follow Listings

We strongly recommend to <u>not index</u> the listings by any bot because it may be reflected in your overall traffic quality and penalizations on behalf of our advertisers may be applied, to avoid that add "no follow" to the rel attribute in every listing.

```
<a href="Advertiser_click_URL" rel="nofollow" target="_blank">text</a>
```

Example of query String with parameters and values

http://marketplaces.vantagemedia.com/Search?zipcode=90210&campaign=6604&maxresults=15&md=11&client_user_a

Multiproduct :
autoinsurance.net **healthplans.com** **CREDIT**and**LOAN**.com UNIVERSITY.COM

Results Output

Handling Errors

If there is an error in the request (missing parameters, headers, wrong data in parameters) the XML Structure of Listings will not be returned. Instead the following error message structure can be returned:

Error output format:

```
{
"Results": [],
"ResponseStatus": {
                "Status": "Error",
                "ErrorCode": "{###}",
                "Message": "{Error Name}"
            }
}
```

Error codes and messages

Error Code	Error Name	Exception Class
1	Internal Server Error	InternalServerErrorExcpetion : PerformanceMarketException
2	Invalid Input	InvalidInputException : PerformanceMarketException
3	Validation Error	ValidationErrorException : PerformanceMarketException