DENIVIP Analytics API

Version: 1.0

Date: August 1, 2012

Section: Client side developer's guide

Revision History

Date	Revision	Author	API version
01.08.2012	The first API version	VA Madison	1.0

Overview

The purpose of the API is to provide third-party automated systems with an interface to the video player analytics service. The API is built on the REST methodology.

Terms and Definitions

Project means a subscriber of the analytics service. Each service customer has one or more projects in the analytics system.

Token means a unique key assigned to the project. Based on the token, all operations involving data retrieval, sending commands from CMS, etc., are run.

Using the API

All requests to the service API are made using HTTP GET.

To use the service, you should first obtain a token. While registering a new project, the administrator generates a token and sends it to the customer.

The query to the API service is insensitive to the parameter order.

All requests to the API have the following format:

```
https://<service address>/<version API>/<API method>?token=<project token>&<parameters>
```

Example:

```
http://api.local/ get-user-views-history?token=4b5783300334000000000aa9&userId=1&page=2&limit=10
```

The server responds in the JSON format. Each server response carries the operation status in the errorCode property of the JSON object returned. The "ERROR_NONE" status means that the command has been run successfully.

Sample response:

```
{"errorCode": "ERROR_NONE"}
```

API Methods

For all the methods, the token parameter is used to identify the project

Data Retrieval

1. /get-stop-point

This method outputs a latest stop point for a given content viewed by a specific user.

Request Parameters:

- userId - user identifier, a mandatory parameter
- contentId - content identifier, a mandatory parameter
- contentType - content type, a mandatory parameter

```
Example:

http://api.local/get-stop-point?token=4b5783300334000000000aa9&userId=1&contentId=2
&contentType=broadcast


Service response:

{"errorCode": "ERROR_NONE", "stopPoint": 10}
```

2. /get-user-views-history

This method outputs the browsing history for the specified user.

Request Parameters:

- userId - user identifier, a mandatory parameter
- page - the number of the requested page. This parameter is optional. If empty, 1 is taken by default. The numbering starts with 1.
- limit - the number of elements on the page. The upper limit is configured at the project level. The default value is specified in the project settings.

```
Example:

http://api.local/get-user-views-history?token=4b5783300334000000000aa9&userId=1&page=2
&limit=10


Service response:
```

```
{"errorCode": "ERROR_NONE", "history": [
    {"contentId": 1,  "contentType": "broadcast", "stopPoint": 10},
    {"contentId": 22, "contentType": "broadcast", "stopPoint": 19990}
  ], "page": 2, "limit": 10}
```

3. /get-active-sessions-count-by-content

This method outputs active sessions for the specified content

Request Parameters:

- contentId - content identifier, a mandatory parameter
- contentType - content type, a mandatory parameter

Example:

```
http://api.local/get-active-sessions-count-by-content?token=4b5783300334000000000aa9
&contentId=1&contentType=broadcast
```

Service response:

```
{"errorCode": "ERROR_NONE", "contentId": 1, "contentType":"broadcast", "sessions": 17}
```

4. /get-user-sessions

This method outputs the number of active sessions of the specified user

Request Parameters:

- userId - the user identifier

Example:

```
http://api.local/get-user-sessions?token=4b5783300334000000000aa9&userId=1
```

Service response:

```
{"errorCode": "ERROR_NONE", "sessions": [
     {"contentId": 1,  "contentType": "broadcast", "stopPoint": 10},
     {"contentId": 13, "contentType": "broadcast", "stopPoint": 1440}
  ], "userId": 1}
```

Control Commands

1. /ban-user

This command prevents a user with a specified ID from viewing the content

Request Parameters:

- userId - the user identifier

Example:

```
http://api.local/ban-user?token=4b5783300334000000000aa9&userId=1
```

Service response:

```
{"errorCode": "ERROR_NONE"}
```

2. /unban-user

This command permits a user with a specified ID to view the content

Request Parameters:

- userId - the user identifier

Example:

```
http://api.local/unban-user?token=4b5783300334000000000aa9&userId=1
```

Service response:

```
{"errorCode": "ERROR_NONE"}
```

Error Codes

- ERROR_NONE - no error, the operation has been successful
- ERROR_UNKNOWN_REQUEST - a non-existing API method has been requested
- ERROR_TOKEN_IS_REQUIRED - the project token has not been specified
- ERROR_INVALID_TOKEN - an invalid token has been passed
- ERROR_INVALID_PARAMS - the full list of mandatory parameters has not been passed
- ERROR_MANDATORY_PARAMETER_WAS_MISSING - a mandatory parameter has not been specified
- ERROR_TOO_BIG_LIMIT - a too high value of a numeric parameter has been specified
- ERROR_DB – during an operation, a database error has occurred.